



UNIVERSITAT JAUME I

ESCOLA SUPERIOR DE TECNOLOGIA I CIÈNCIES EXPERIMENTALS
GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

***Desarrollo de una herramienta
informática interactiva para el diseño de
sistemas de seguridad en máquinas***

TRABAJO FIN DE MÁSTER

AUTOR/A

Alberto Porras Vázquez

DIRECTOR/A

Julio Ariel Romero Pérez

Castellón, septiembre de 2016

“Quo usque tandem abutere [...] patientia nostra?”

Marco Tulio Cicerón

Resumen

Desarrollo de una Herramienta Informática Interactiva para el Diseño de Sistemas de Seguridad en Máquinas

Alberto Porras Vázquez

La seguridad es un requisito fundamental de los actuales sistemas de producción. Si bien los accidentes laborales provocados por el fallo imprevisible de la maquinaria o su uso indebido han estado presentes desde el primer día, no se tuvo constancia de su verdadera importancia hasta finales del siglo XX, por lo que su aplicación en el ámbito industrial resultó ser tardía.

Los aspectos relativos a la seguridad han de tenerse en cuenta en todas las etapas del ciclo de vida de una máquina: desde su diseño hasta su desmontaje, pasando por las fases de fabricación, instalación, funcionamiento y mantenimiento. De este modo, la integración de elementos de seguridad fiables en el lugar de trabajo se ha convertido en un requisito indispensable. Con ello, estos requerimientos han cambiado y evolucionado y han buscado adaptarse al creciente desarrollo de los sistemas automáticos, hasta materializarse en la legislación actual. Las normas que regulan el diseño de las partes de los sistemas de mando relacionados con la seguridad de las máquinas han experimentado así un importante cambio en los últimos años: las normas EN ISO 13849-1 y EN 62061 han sustituido a la EN 954-1, que dejó de ser válida en diciembre de 2011. En las normas actuales se plantea la metodología de diseño como un proceso iterativo que requiere la realización de una gran cantidad de cálculos. Además, el resultado final depende de numerosos parámetros relacionados entre sí. Esto hace imprescindible el uso de programas informáticos como ayuda en el diseño.

El propósito del proyecto es crear una herramienta informática interactiva que permita asistir al diseño de los sistemas de seguridad en máquinas teniendo en cuenta la directiva relativa a máquinas vigente a nivel europeo y con aplicación en el territorio nacional, en concreto la norma EN ISO 13849-1. Dicha herramienta debe permitir realizar simulaciones de sistemas de seguridad y evaluar así sus parámetros más importantes, guiando al usuario en todo momento durante la fase de diseño. En general,

dados unos requisitos de seguridad, se obtendrán las características de los elementos que conformarán el sistema (sensores, elementos de mando, relés de seguridad, etc.) y se propondrán, en última instancia, los dispositivos pertinentes suministrados por determinados fabricantes del sector. Por consiguiente, el desarrollo de un programa de tales características requiere de un estudio exhaustivo y de un conocimiento profundo de la normativa correspondiente, de manera que ésta pueda ser plasmada de forma precisa en la herramienta desarrollada.

Para la implementación de la aplicación informática se usará el software Easy Java Simulations (EjsS), que permite la creación y programación de simulaciones interactivas en Java o JavaScript. Esto permitirá tener una herramienta que podrá ejecutarse en distintos sistemas operativos o como un applet en un navegador web, lo cual puede ser un elemento que favorezca su explotación a nivel industrial. La aplicación desarrollada será evaluada mediante su aplicación a varios casos de estudio para corroborar su efectividad como herramienta de ayuda al diseño de las partes de los sistemas de mando relacionados con la seguridad en máquinas.

Palabras clave: *seguridad en máquinas, fallo peligroso, diseño, simulación, programa interactivo, ISO 13849-1, parte del sistema de mando relativa a la seguridad.*

Índice

Resumen.....	4
1. Introducción.....	12
1.1. Objetivos.....	12
1.2. Alcance y límites.....	13
1.3. Antecedentes.....	15
1.4. Estructura del documento.....	17
2. Conceptos clave.....	20
2.1. Introducción.....	20
2.2. Parte relativa a la seguridad de un sistema de control (SRP/CS).....	21
2.3. Nivel de fiabilidad (PL).....	22
2.4. Categoría (Cat.).....	23
2.5. Tiempo medio hasta fallo peligroso (MTTF _d).....	25
2.6. Cobertura de diagnóstico (DC).....	26
2.7. Fallo de causa común (CCF).....	27
2.8. Probabilidad de un fallo peligroso por hora (PFH _d).....	28
3. Normativa EN ISO 13849-1.....	30
3.1. Introducción.....	30
3.2. Diseño de la seguridad.....	32
3.2.1. Definición de las funciones de seguridad y especificación de sus características.....	34
3.2.2. Determinación del nivel de fiabilidad requerido.....	36
3.2.3. Identificación de los elementos relativos a la seguridad.....	37
3.2.4. Evaluación del nivel de fiabilidad resultante.....	38

3.2.5. Verificación del nivel de fiabilidad del sistema.....	40
3.2.6. Validación de las funciones de seguridad.....	40
3.3. Información adicional.....	41
4. Metodología propuesta.....	42
4.1. Introducción.....	42
4.2. Diseño del sistema de seguridad.....	43
4.2.1. Identificación de las funciones de seguridad y especificación de sus características.....	45
4.2.2. Determinación del nivel de fiabilidad requerido de la función de seguridad.....	45
4.2.3. Asignación del nivel de fiabilidad PL de la función de seguridad.....	46
4.2.4. Determinación del nivel de fiabilidad PL.....	46
Consideración 1: categoría.....	46
Consideración 2: cobertura de diagnóstico.....	50
Consideración 3: tiempo medio hasta fallo peligroso.....	51
4.2.5. Determinación del $MTTF_d$ del SRP/CS.....	54
4.2.6. Evaluación de las características de los SRP/CS.....	56
4.2.7. Selección de los componentes de seguridad.....	57
5. Diseño y programación.....	60
5.1. Introducción.....	60
5.2. Entorno gráfico del programa.....	62
5.2.1. Barra de herramientas.....	63
Definición del sistema.....	64
Cálculo del nivel de fiabilidad.....	66
Introducción de nuevos dispositivos.....	68
Ayuda.....	69
5.2.2. Árbol de la estructura.....	70
5.2.3. Área de trabajo.....	71
Especificaciones del caso de estudio.....	71
Pestaña 1: gráfico de las características del sistema.....	73
Pestañas 2, 3 y 4: estudio particular de las SRP/CS.....	74
5.2.4. Área de mensajes.....	78

5.2.5. Área de dispositivos.....	78
5.3. Procedimiento de diseño.....	81
5.3.1. Definición del sistema.....	82
5.3.2. Manipulación del sistema.....	83
5.3.3. Visualización de los resultados.....	85
6. Aplicación de la herramienta.....	86
6.1. Introducción.....	86
6.2. Ejemplo 1: célula de trabajo.....	87
6.2.1. Presentación del caso.....	87
6.2.2. Resolución teórica.....	87
6.2.3. Resolución experimental.....	89
6.3. Ejemplo 2: sistema redundante.....	92
6.3.1. Presentación de caso.....	92
6.3.2. Resolución teórica.....	93
6.3.3. Resolución experimental.....	95
6.4. Conclusiones.....	96
7. Conclusiones.....	98
7.1. Resumen y consideraciones.....	98
7.2. Oportunidades de mejora y desarrollo.....	99
Bibliografía.....	102
Programas de cálculo.....	103
Lista de símbolos, abreviaciones y siglas.....	104
Anexo A. Tablas.....	106
A1. Cobertura de diagnóstico (DC).....	106
A2. Probabilidad de fallo peligroso por hora (PFH _d).....	109
Anexo B. Código del programa.....	110
B1. Botón “Máquina”.....	110

B2. Botón “Riesgo”	110
B3. Botón “Función de seguridad”	112
B4. Botón “Nivel de fiabilidad”	118
B5. Botón “Lista de dispositivos”	121
B6. Botón “Optimización”	122
B7. Botón “Dispositivos de seguridad”	131
B8. Botón “Optimización económica”	139
B9. Activación de botones y ventanas	144
B10. Gráfico general	146
B11. Gráfico $B10_d = f(n_{op})$	146
B11.1. Dispositivo de entrada	146
B11.2. Unidad lógica	147
B11.3. Dispositivo de salida	147
B12. Gráfico $MTTF_d = f(PFH_d)$	148
B12.1. Dispositivo de entrada	148
B12.2. Unidad lógica	166
B12.3. Dispositivo de salida	185
B13. Ventana “Dispositivos de seguridad”	204
B13.1. Dispositivo de entrada	204
B13.1.1. Ordenación por defecto	205
B13.1.2. Ordenación según PL	205
B13.1.3. Ordenación según PFH_d	208
B13.1.4. Ordenación según $MTTF_d$	209
B13.1.5. Ordenación según tecnología	211
B13.1.6. Ordenación según precio	214
B13.2. Unidad lógica	215
B13.2.1. Ordenación por defecto	216
B13.2.2. Ordenación según PL	217
B13.2.3. Ordenación según PFH_d	219
B13.2.4. Ordenación según $MTTF_d$	221
B13.2.5. Ordenación según tecnología	222
B13.2.6. Ordenación según precio	225
B13.3. Dispositivo de salida	226
B13.3.1. Ordenación por defecto	227

B13.3.2. Ordenación según PL.....	228
B13.3.3. Ordenación según PFH_d	230
B13.3.4. Ordenación según $MTTF_d$	232
B13.3.5. Ordenación según tecnología.....	233
B13.3.6. Ordenación según precio.....	236
Anexo C. Estudio de los programas de diseño.....	238
C1. SISTEMA.....	238
C2. Safety Evaluation Tool.....	239
C3. PAScal.....	239
C4. ProSET®.....	240
C5. Consideraciones.....	241
Pliego de condiciones.....	242
I. Especificaciones de los elementos constitutivos del proyecto.....	242
II. Reglamentación y normativas aplicables.....	242
Presupuesto.....	244
I. Estado de mediciones.....	244
II. Presupuesto.....	245
III. Estudio económico.....	245

1

Introducción

La Seguridad en Máquinas se define como la disciplina que estudia aquellos aspectos relacionados con el diseño de sistemas y el control de los parámetros de sus componentes para la consecución de conjuntos seguros desde los puntos de vista de la salud y el estado de los usuarios de los mismos, por un lado, y de los daños generados en la máquina y en su entorno circundante, por el otro. La preocupación creciente por seleccionar e implementar correctamente dichos diseños seguros ha supuesto una revolución en los últimos años que se ha plasmado en una normativa cambiante con la propia maquinaria, que ha perseguido su adaptación a la arquitectura y tecnología de ésta mediante la verificación de unos requisitos de seguridad, fiabilidad, rendimiento y coste impuestos.

En este capítulo se ahondará en la estrategia propuesta a tal efecto y se expondrá una primera aproximación a la metodología, recursos y estructura seguidos en este informe.

1.1. Objetivos.

El objetivo de este proyecto es el desarrollo de una herramienta informática interactiva que asista al diseño de los sistemas de seguridad en maquinaria industrial de acuerdo con las directrices fijadas por la Directiva de Máquinas vigente y, en particular, por la norma internacional derivada de la misma, EN ISO 13849-1, con aplicación en el territorio nacional. Dicho programa permitirá igualmente la simulación de los sistemas estudiados y evaluar, por consiguiente, el conjunto de parámetros más importantes que lo definen. Con esto, el usuario de la herramienta será guiado por la misma a lo largo del proceso de diseño, la cual le proporcionará ideas y recomendaciones que se verán reflejadas en reglas y restricciones que permitan tomar decisiones acerca del diseño y obtener, en última instancia, las características de los dispositivos a integrar en el sistema, junto con un listado de los productos sugeridos según los fabricantes del sector, que recojan las propiedades halladas, dados unos requisitos de seguridad iniciales.

Para la consecución de tal fin, se propone el uso de la herramienta de programación Easy Java Simulations (EjsS), que permite la creación, de manera sencilla, de un interfaz siguiendo un lenguaje de Java o JavaScript, el cual asegurará con el formato del producto final un fácil acceso, y por ende un amplio alcance, del mismo por parte de las empresas afectadas por la normativa tratada.

1.2. Alcance y límites.

El marco de aplicación del presente proyecto viene fijado por la Directiva de Máquinas 2006/42/EC y la norma EN ISO 13849-1. La Directiva se aplicará, en general, a máquinas; equipos intercambiables; componentes de seguridad; accesorios de elevación; cadenas, cables y cinchas; dispositivos amovibles de transmisión mecánica; y cuasi máquinas y se excluirán de su campo de aplicación los componentes de seguridad utilizados como recambio en máquinas, los equipos o máquinas diseñados para fines específicos (uso nuclear, militar, en parques de atracciones, etc.), determinados medios de transporte, armas, y productos y equipos eléctricos o electrónicos. Además, se excluirá toda máquina existente en la Unión Europea con anterioridad a la aprobación de la norma presente, a no ser que haya sido modificada de manera significativa. Será de aplicación exclusiva a los estados miembro de la Unión Europea, los cuales no deberán prohibir, restringir o prevenir su implantación mediante la aplicación de leyes nacionales, ordenanzas o estándares de carácter más restrictivo en lo concerniente a la calidad de la máquina. Por otro lado, la norma EN ISO 13849-1 se encuentra armonizada con la Directiva anterior. Una norma armonizada es aquella que describe los objetivos y requerimientos de una Directiva Europea, a la que hacen referencia de manera detallada, y que cuyo cumplimiento constituye una prueba de presunción de su conformidad con ésta, sustituyendo con ello todo estándar nacional abarcando el mismo campo de aplicación. A diferencia de la Directiva 2006/42/EC, el estándar EN ISO 13849-1 es de alcance internacional, si bien de aplicación voluntaria.

Según el nivel de detalle alcanzado con el estándar o la clase de elemento afectado por éste, se distinguen tres tipos de estándar:

- Estándar de tipo A. Describe normas de seguridad elementales y expone los principales conceptos, aspectos y principios afectando a la máquina y su diseño. Se incluyen la evaluación de riesgos, los elementos de protección y las paradas de emergencia, entre otros. En el ámbito estudiado, se refleja en la norma EN ISO 12100.
- Estándar de tipo B. Describe normas de seguridad genéricas, centrándose en aspectos o dispositivos específicos de seguridad aplicables a un amplio rango de maquinaria. Se diferencian entre estándares de clase B-1, referentes a aspectos de seguridad particulares (distancias de seguridad, protección eléctrica, etc.), y

de clase B-2, referentes a dispositivos de seguridad (resguardos, guardas, dispositivos de paro de emergencia, etc.). En el ámbito estudiado, se refleja por las normas ISO 13855 (EN 999) o EN ISO 13851 (EN 574), entre otras.

- Estándar de tipo C. Describe normas de seguridad para maquinaria específica, tales como prensas hidráulicas o mecánicas o equipos de parada. Tiene prioridad por encima de los estándares de tipo A y B, pudiendo hacer referencia a ambos tipos indistintamente. En el ámbito estudiado, se refleja por las normas EN 415-4 o EN ISO 10218-1, entre otras.

La Figura 1.1. ilustra la tipología de la normativa de seguridad relativa a la Directiva de Máquinas, realizando una clasificación de los tres tipos de estándar mencionados en la forma de bolas de color rojo, verde o azul en función de si representan estándares de tipo A, B o C, respectivamente.

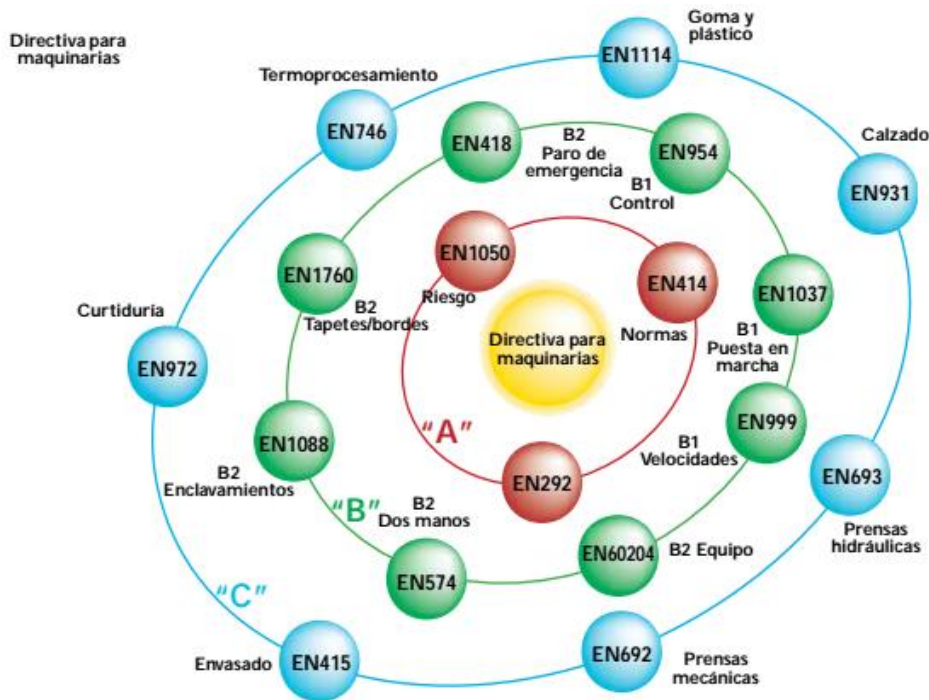


Figura 1.1. Tipología de la normativa de seguridad relativa a la Directiva de Máquinas.

La norma EN ISO 13849-1 se encuadra dentro de los estándares de tipo B-1 y, como tal, abarcará los aspectos mencionados previamente, excluyendo todo aquel campo de aplicación no incluido en su categoría.

Las personas afectadas directamente por esta normativa y que por tanto asumirán responsabilidades de su correcta aplicación son definidas por la Directiva 2009/104/EC, relativa a la utilización de equipos de trabajo, y son dos: el proveedor de la máquina y

su usuario, cada uno con obligaciones claramente delimitadas. Se entiende por proveedor aquella figura que construye, modifica y/o integra máquinas, así como aquélla que las distribuye, sin ser necesariamente fabricante. El usuario es aquella persona que utiliza la máquina. Los aspectos a cubrir por el proveedor son los de realizar un diseño seguro del equipo, de acuerdo con los requerimientos de seguridad y de salud de los trabajadores dictados por la Directiva de Máquinas; elaborar instrucciones de funcionamiento de la máquina en el idioma oficial del país donde será suministrada; preparar la documentación técnica de la máquina, conteniendo diagramas, cálculos y ensayos que den apoyo a la conformidad del producto de acuerdo con la Directiva de Máquinas; y emitir una declaración de conformidad con respecto a la normativa anterior, reflejándose en un marcado CE del producto. Por otro lado, los usuarios deben velar, en la medida de sus posibilidades y en función del escenario, por su propia seguridad y salud en el trabajo y por la de aquellas personas en su entorno que puedan verse afectadas directamente por su actividad profesional. En el caso de un empresario, éste adoptará las medidas necesarias con tal de asegurar que las condiciones de seguridad y de salud mínimas de los usuarios son alcanzadas, cerciorándose que se comprueba el estado de los equipos de manera periódica, informando a los trabajadores acerca de los equipos de trabajo utilizados y formando a los mismos sobre los riesgos en que pueden derivar su uso.

1.3. Antecedentes.

Las directivas europeas constituyen estándares fijados por la Comisión Europea y por el Consejo de la Unión Europea que son aplicados en el conjunto de los estados miembros de la Unión Europea a través de sus leyes nacionales y que atienden al creciente interés de tener un mercado abierto de libre movimiento en la totalidad de su territorio, junto con un anhelo de proteger y preservar el estado y la salud de sus habitantes tanto en la esfera privada como en la profesional. Los principales entes que se ven afectados por las directivas son por tanto los fabricantes que elaboran los productos concernientes a las mismas y las organizaciones introduciendo o comercializándolos.

El afán por regular la seguridad en el entorno laboral había estado en suspenso desde que comenzara la actividad productiva o industrial, en general, tanto en pequeña como a gran escala. Los accidentes debido a fallos no previstos de la maquinaria o su utilización inadecuada siempre han estado presentes, si bien las primeras acciones legales no se tramitaron hasta finales del siglo XX. Los estándares concernientes a la seguridad en máquinas han evolucionado desde su primera emisión en 1989, con la Directiva 89/392/EEC, hasta la actualidad. En efecto, este hecho responde al cambio constante que ha experimentado, y sigue teniendo, la tecnología de control de máquinas integrada en las mismas, la cual ha progresado de circuitos de arquitectura sencilla a sistemas hard y soft cada vez más complejos, tales como los circuitos integrados o los

microprocesadores. Aunque los beneficios de disponer una tecnología más sofisticada son claros, añade un nivel de dificultad adicional al diseño de los sistemas de control, puesto que su fallo conlleva un comportamiento impredecible de la máquina y puede comportar riesgos adicionales al personal o a los elementos de su entorno, maquinaria o productos. Esto supone que, a mayor complejidad del sistema de control, mayor es la dificultad de identificar y prevenir las consecuencias de los fallos.

La consideración de las distintas arquitecturas aplicadas en los sistemas de seguridad y la aplicación de nuevos conceptos abarcando aspectos no atendidos anteriormente consolidaron la aparición de nuevas directivas que con los años serían modificadas o completadas mediante la emisión de enmiendas. Con tal propósito, surge la norma EN 954-1 en 1996, la cual desembocaría en 1999 en una primera versión de la norma ISO 13849-1, más completa que la anterior al tener en cuenta aspectos cuantitativos de la fiabilidad de los componentes, además de los aspectos estructurales o de selección recogidos en la normativa de la que derivaba. No obstante, no sería hasta la revisión de la norma ISO 13849-1 en 2006 cuando aparece por primera vez el término de nivel de fiabilidad o PL para referirse a aquella habilidad de las partes del sistema de control relativas a la seguridad de ejecutar una función de seguridad bajo condiciones de operación predecibles, referido en términos de probabilidades de fallo peligroso por hora. La implantación de la Directiva 2006/42/EC en el mismo año y la sustitución definitiva de la norma EN 954-1 en 2011, tras ser decretada su retirada a partir del 2006, supone un cambio de paradigma, en tanto que el enfoque determinista de las partes relativas a la seguridad en un sistema de control pierde importancia en pro de una visión enfocada más hacia los aspectos probabilísticos cuantificables de los eventos con una gestión de los fallos de causa común en los sistemas.

En la actualidad, son dos las normas que compiten por suceder a la normativa retirada EN 954-1. Por un lado, se halla la versión actualizada del estándar EN ISO 13849-1:2015, elaborada específicamente para representar una continuación lógica de dicha norma. Por otra parte, la normativa EN IEC 62061, desarrollada a partir de la EN IEC 61508 por la Comisión Electrotécnica Internacional (IEC), disputa su posición con un punto de vista que dota de una mayor importancia a los aspectos probabilísticos de la fiabilidad, requiriendo de un mayor grado de comprensión por parte de su usuario; a pesar de ello, y contrariamente a la anterior norma ISO, los estándares IEC son específicos del sector de los componentes eléctricos, electrónicos y de la electrónica programable, por lo que la extrapolación de sus procedimientos a elementos de naturaleza dispar no es posible. La Figura 1.2. ilustra el posicionamiento de los estándares EN ISO 13849-1 y EN IEC 62061 con respecto de la norma EN 954-1.

Como se expondrá en el Capítulo 3, si bien la norma IEC constituye un enfoque que comúnmente es aplicado en el ámbito de la Seguridad en Máquinas, no es de interés para el desarrollo del proyecto presentado en este documento. Su estudio se llevará a cabo en base a los preceptos y directrices marcados por la norma EN ISO 13849-1.

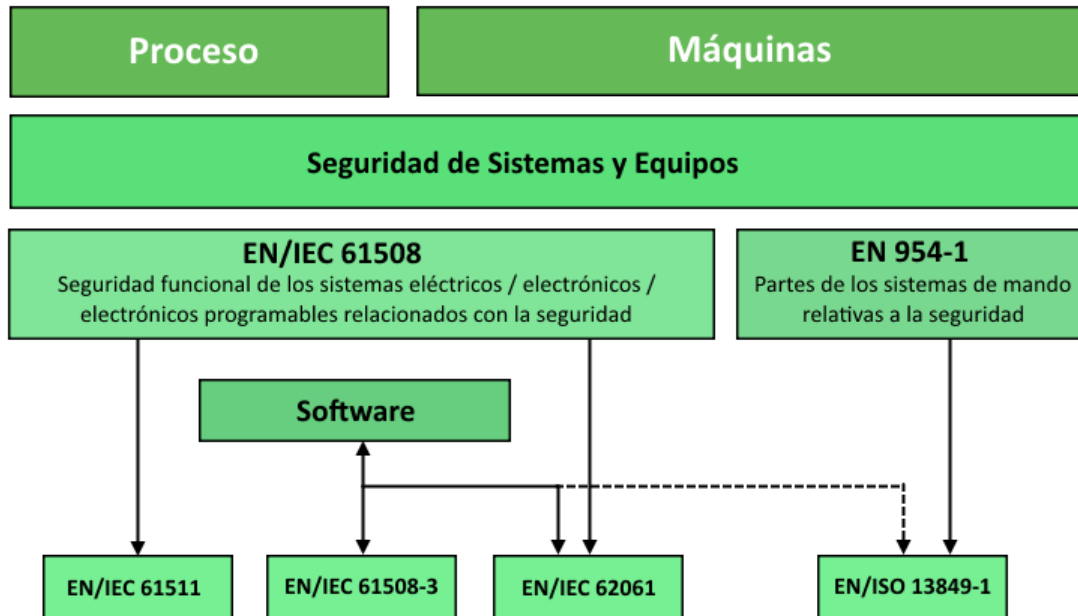


Figura 1.2. Posicionamiento de los estándares EN ISO 13849-1 y EN IEC 62061 con respecto de la norma EN 954-1.

1.4. Estructura del documento.

El documento contiene siete capítulos, el primero de los cuales viene representado por esta introducción. El contenido detallado de los capítulos principales se resume en los puntos listados a continuación.

- El **Capítulo 2** recoge el conjunto de conceptos básicos relativos a la Seguridad en Máquinas requeridos para el seguimiento del desarrollo del proyecto y de las conclusiones extraídas.
- El **Capítulo 3** introduce las nociones elementales de la normativa EN ISO 13849-1, que constituye la base y el marco de aplicación del programa diseñado. En las secciones que contiene se ahondará en el proceso seguido para la evaluación del nivel de seguridad de un sistema.
- El **Capítulo 4** propone la metodología a programar para el diseño seguro de sistemas industriales, basándose en los conceptos y principios descritos en los capítulos precedentes.
- El **Capítulo 5** constituye una introducción a la herramienta de programación EjsS y sus principales características, exponiendo a continuación de manera

detallada su aplicación en el diseño de una interfaz para la implantación de la metodología elaborada anteriormente.

- El **Capítulo 6** presenta varios ejemplos prácticos resultantes de la aplicación de la herramienta de diseño elaborada y la posterior comparación de los resultados obtenidos con aquéllos hallados en las situaciones reales.
- El **Capítulo 7** recoge las conclusiones finales extraídas del proyecto a partir del análisis del conjunto de los capítulos del mismo.

2

Conceptos clave

Antes de abordar el problema del diseño seguro de sistemas, es preciso remarcar que el campo de la Seguridad de Máquinas, como todo ámbito teórico o práctico, dispone de su propia terminología específica con la que es preciso estar familiarizado para el mejor entendimiento de los procedimientos básicos descritos por el mismo. Una vez asentada una base de conceptos elementales, podrán abarcarse situaciones de problemática dispar y realizar los diseños oportunos que compete a esta disciplina.

2.1. Introducción.

En este capítulo se van a presentar las principales nociones que se necesitarán a lo largo de este proyecto. El marco de estudio de la Seguridad de Máquinas tratado en el documento se llevará a nivel de las partes relativas a la seguridad que constituyen un sistema de control cualquiera, esto es, todos los parámetros e ideas enunciados irán referenciados a una estructura integrada en la máquina en sí. Dichas estructuras son incorporadas con el fin de prevenir o controlar los posibles fallos que puedan surgir durante el ciclo de vida del aparato y, por tanto, para asegurar su efectividad, requieren de niveles de fiabilidad dados. Estos niveles de fiabilidad se estiman en función de las características de los dispositivos de la estructura que los incluye; la disposición de tales dispositivos, el tiempo transcurrido hasta un fallo peligroso, la capacidad de detectarlo o el fallo debido a causas comunes permiten singularizar el comportamiento de los dispositivos asociados y diferenciar su grado de seguridad con respecto del de los demás.

El capítulo se organiza de la siguiente manera. En la Sección 2.2. se proporciona una breve definición acerca de las partes relativas a la seguridad de un sistema de control, las cuales constituyen el marco de aplicación de los elementos que siguen. La Sección 2.3. expone el concepto de nivel de fiabilidad, si bien se ha dejado expresamente incompleto para detallar su cálculo y sus particularidades en capítulos

futuros. Las secciones comprendidas entre la 2.4. hasta la 2.8. introducen los conceptos, valores permitidos y clasificaciones de categoría, tiempo medio hasta el fallo peligroso, cobertura del análisis, fallo de causa común y probabilidad de fallo peligroso por hora, nociones que describen el comportamiento de los subsistemas y que la combinación de las cuales permitirá, en última instancia, hallar el nivel de fiabilidad del mismo.

2.2. Parte relativa a la seguridad de un sistema de control (SRP/CS).

Las partes relativas a la seguridad de un sistema de control (del inglés, *Safety-Related Part of a Control System*) constituyen estructuras ordenadas que implementan una función de seguridad mediante el uso de dispositivos que conforman en sí un subsistema o unidad y que realizan las funciones de captar las señales de entrada, procesar la información recibida y generar en consecuencia salidas relacionadas con la seguridad en la forma de respuestas. Esto se traduce en la utilización de dispositivos de entrada (sensores, botones o interruptores), unidades lógicas (autómatas, módulos o relés de seguridad) y dispositivos de salida (actuadores o contactores). La configuración que los elementos de un sistema de control pueden adoptar es diversa. Algunos sistemas simples pueden no poseer una solución lógica y concentrar su función en la implementación de sendos subsistemas de entrada y de salida con un único canal. En otras ocasiones, los subsistemas pueden contener más de un dispositivo para efectuar su función, con comunicaciones de doble canal o bloques. Del mismo modo, puede darse el caso de sistemas combinando ambas clases de configuración, con bloques de doble canal o de canal individual, incrementando igualmente la complejidad del conjunto. El esquema de la Figura 2.1. ilustra la estructura de una parte relativa a la seguridad de un sistema de control con un canal individual integrando un dispositivo de entrada, un dispositivo lógico y un dispositivo de salida.

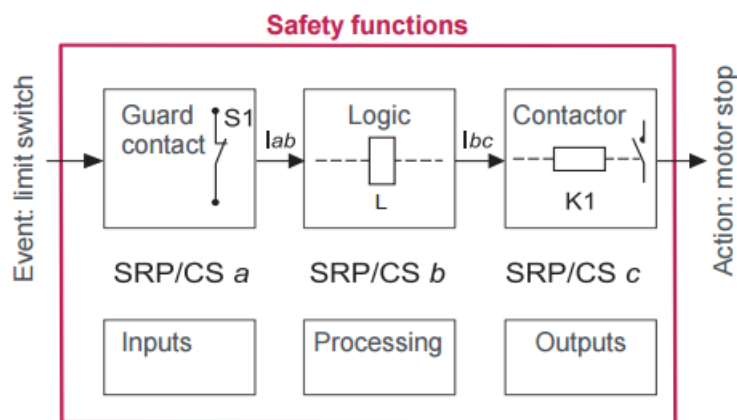


Figura 2.1. Estructura de una parte relativa a la seguridad de un sistema de control compuesta por tres dispositivos o subsistemas: entrada, lógico y salida.

2.3. Nivel de fiabilidad (PL).

El nivel de fiabilidad o nivel de prestación (del inglés, *Performance Level*) es un parámetro que indica la capacidad de un componente de seguridad de realizar una función de seguridad bajo condiciones predecibles, con el fin de obtener una reducción esperada en el riesgo de un determinado sistema. Dado que los componentes de seguridad no se encuentran de manera aislada en los sistemas de control, sino que se integran entre ellos para constituir una misma función de seguridad, se emplea el término de nivel de fiabilidad requerido (PL_r) para referirse al nivel de prestación que debe alcanzar ésta para asegurar su cumplimiento y la consiguiente reducción del riesgo que se deriva de ella.

El nivel de fiabilidad requerido se evalúa en base a las características del riesgo sobre el que se pretende actuar: su impacto, relativo a las personas, el entorno o la propia máquina; el grado de exposición al mismo; y la probabilidad que existe de reducirlo o evitarlo. En un grado de detalle más discreto, el nivel de fiabilidad de cada subsistema que integra una función de seguridad depende de cuatro requerimientos cuantitativos cubriendo aspectos básicos distintos: la categoría (descripción de la estructura), el tiempo medio hasta fallo peligroso (medida de la fiabilidad), la cobertura de diagnóstico (medida de cuán diagnosticable es) y el fallo de causa común (medida de la resistencia). A ello se le suman requerimientos de carácter cualitativo, tales como: el comportamiento del sistema de seguridad en condiciones de fallo, la monitorización de los fallos sistemáticos, el uso de software relativo a la seguridad o el grado de implementación de la función de seguridad en condiciones ambientales específicas. Independientemente del punto de referencia marcado, el nivel de fiabilidad se expresa en una escala alfabética, donde un nivel “a” indica un grado de protección ante el riesgo bajo y un nivel “e” implica un grado de protección ante el mismo elevado.

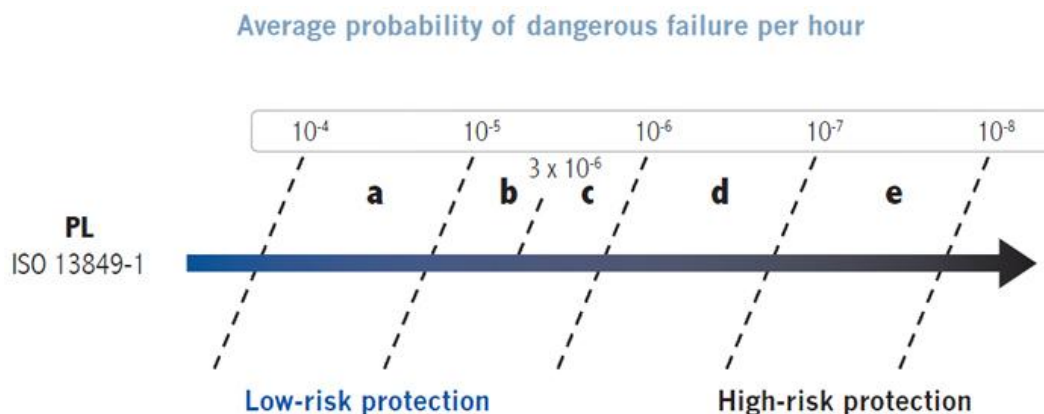


Figura 2.2. Escala alfabética del nivel de fiabilidad según la norma EN ISO 13849-1.

En el Capítulo 3 se detalla el procedimiento seguido en la determinación del nivel de fiabilidad de los sistemas de control y sus componentes, de acuerdo con la norma EN ISO 13849-1.

2.4. Categoría (Cat.).

La categoría de un sistema de control es un parámetro que representa la clasificación de sus partes relativas a la seguridad en base a su resistencia frente a defectos y a su comportamiento cuando expuesto a las condiciones de los mismos, y que atiende a la arquitectura o disposición de dichos elementos, su fiabilidad y su capacidad para detectar los defectos. Se entiende por defecto el estado de incapacidad del dispositivo para desempeñar su función requerida, con la excepción de situaciones de mantenimiento preventivo, acciones programadas o falta de recursos externos. Pueden distinguirse hasta un total de cinco categorías que, de menor a mayor complejidad, se identifican como B, 1, 2, 3 y 4. Las arquitecturas que se describen a continuación, representadas en forma de esquema lógico, no son de uso obligatorio, si bien la omisión de su utilización exige la evaluación de la categoría mediante métodos alternativos.

Categoría B y 1. Ambas categorías comparten la misma estructura designada y el diseño, selección y montaje de sus dispositivos deben respetar los principios de seguridad descritos en la norma EN ISO 13849-2. No existen medios para la detección de fallos. Una situación de fallo puede resultar en la pérdida de la función de seguridad. No obstante, la probabilidad de que esto ocurra en los sistemas de categoría 1 es menor, al implementar, de manera adicional, principios de seguridad probados.

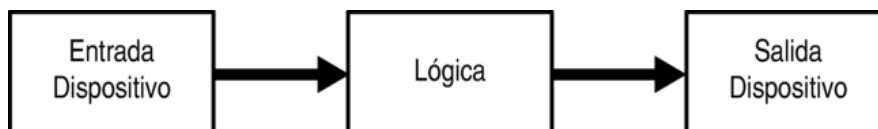


Figura 2.3. Esquema lógico de un sistema de categoría B ó 1.

Categoría 2. El diseño, selección y montaje de los dispositivos integrados deben respetar los principios de seguridad descritos en la norma EN ISO 13849-2. Los fallos son detectados a partir de un monitoreo mediante una prueba funcional del sistema o subsistema ejecutada con una frecuencia mínima de cien pruebas por cada demanda de la función de seguridad, y perduran desde su ocurrencia hasta su detección.

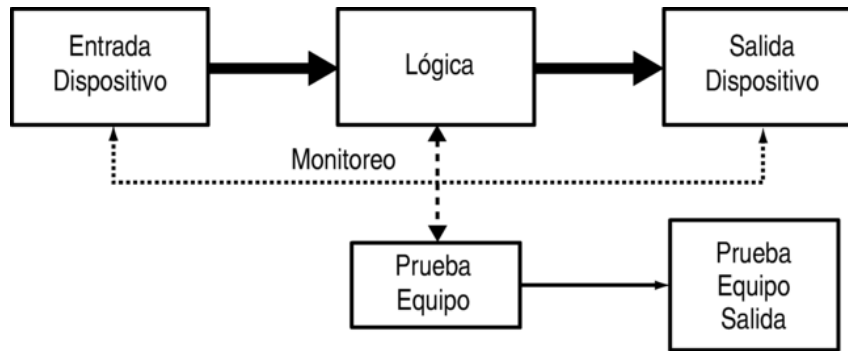


Figura 2.4. Esquema lógico de un sistema de categoría 2.

Categoría 3. El diseño, selección y montaje de los dispositivos integrados deben respetar los principios de seguridad descritos en la norma EN ISO 13849-2. En caso de fallo único, la función de seguridad siempre está asegurada, lo cual suele conseguirse con una estructura de doble canal. La mayoría de los fallos son detectados, bien durante la ejecución de la función de seguridad o cuando el monitoreo por pruebas funcionales del sistema o subsistemas es llevado a cabo. Una acumulación de fallos puede desembocar en la pérdida de la función de seguridad.

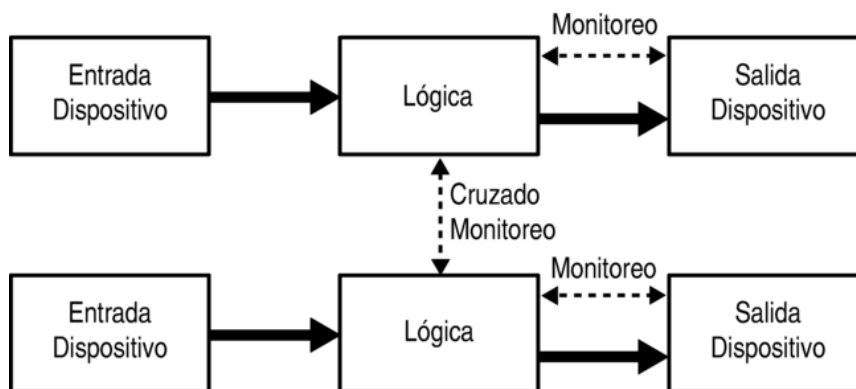


Figura 2.5. Esquema lógico de un sistema de categoría 3 y 4.

Categoría 4. El diseño, selección y montaje de los dispositivos integrados deben respetar los principios de seguridad descritos en la norma EN ISO 13849-2. Presenta un diagrama lógico idéntico al de un sistema de categoría 3, si bien exige un mayor grado de monitoreo entre las unidades lógicas y los dispositivos de salida. Para todo tipo de fallos la función de seguridad siempre está asegurada y su detección se realiza de manera puntual. Contrariamente a su estructura homóloga de categoría inferior, una acumulación de fallos no desemboca en una pérdida de la función de seguridad.

2.5. Tiempo medio hasta fallo peligroso (MTTF_d).

El tiempo medio hasta fallo peligroso (del inglés, *Mean Time To dangerous Failure*) es una indicación teórica del tiempo medio transcurrido, expresado en años, que tarda cada uno de los canales de la arquitectura de un sistema en padecer un fallo peligroso dada una función de seguridad. La estimación del tiempo medio hasta fallo peligroso se lleva a cabo jerárquicamente haciendo uso de tres tipos de recursos: a) la información suministrada por el fabricante del producto, b) los métodos descritos en los anexos C (Cálculo o valoración del MTTF_d para componentes individuales) y D (Método simplificado para estimar el MTTF_d para cada canal) de la norma EN ISO 13849-1, o c) la designación de un tiempo por defecto de diez años. En general, se establecen tres niveles que permiten catalogar el tiempo medio hasta fallo peligroso de un canal en función de su duración, para la cual se impone un valor máximo de 100 años con el fin de limitar la dependencia exclusiva de sus partes relativas a la seguridad en la fiabilidad y permitir que pueda así abarcar otros aspectos de su arquitectura. No obstante, este límite puede ser sobrepasado en componentes individuales. La clasificación expuesta se muestra en la Tabla 2.1.

Designación	Rango
Bajo	3 años ≤ MTTF _d < 10 años
Medio	10 años ≤ MTTF _d < 30 años
Alto	30 años ≤ MTTF _d < 100 años

Tabla 2.1. Clasificación del tiempo medio hasta fallo peligroso de un canal.

Dado un canal individual de un sistema de control, el tiempo medio hasta fallo peligroso del propio canal es determinado a partir de los MTTF_{di} singulares de los N elementos independientes que lo constituyen y contribuyen a su función de seguridad, siguiendo la expresión (2.1).

$$\frac{1}{MTTF_d} = \sum_{i=1}^N \frac{1}{MTTF_{di}} \quad (2.1)$$

En sistemas de control de doble canal con tiempos medio hasta fallo peligroso por canal diferente, puede hallarse el MTTF_d del conjunto bien asumiendo el caso más desfavorable, esto es, el valor más bajo, o bien analíticamente con la expresión (2.2).

$$MTTF_d = \frac{2}{3} \cdot \left(MTTF_{dc1} + MTTF_{dc2} - \frac{1}{\frac{1}{MTTF_{dc1}} + \frac{1}{MTTF_{dc2}}} \right) \quad (2.2)$$

donde $MTTF_{dC1}$ y $MTTF_{dC2}$ son los tiempos medio hasta fallo de dos canales distintos de un mismo sistema de control.

En un nivel más discreto, el tiempo medio hasta fallo peligroso de componentes sujetos a desgaste por su actividad mecánica suele expresarse en términos de su ciclo de operación, incluyéndose en éstos dispositivos de naturaleza mecánica, electromecánica o neumática y excluyéndose aquéllos hidráulicos, eléctricos o electrónicos. El tiempo medio hasta fallo peligroso se exprime como una función del número medio de ciclos hasta que el 10% de los componentes falla de manera peligrosa B_{10d} transcurrido un tiempo de operaciones anuales n_{op} .

$$MTTF_d = \frac{B_{10d}}{0,1 \cdot n_{op}} \quad (2.3)$$

El tiempo medio de operaciones anuales se define a su vez en base al número medio de días de utilización por año d_{op} , el número de horas de utilización por día h_{op} y el tiempo medio entre el comienzo de dos días sucesivos de utilización del componente t_{ciclo} , expresado en segundos por ciclo.

$$n_{op} = \frac{d_{op} \cdot h_{op} \cdot 3600 s / h}{t_{ciclo}} \quad (2.4)$$

Otra manera de representar estos datos es a través del tiempo medio hasta el fallo peligroso del 10% de los componentes T_{10d} , el cual limita el uso de los dispositivos y se deriva directamente del número de ciclos B_{10d} detallado anteriormente.

$$T_{10d} = \frac{B_{10d}}{n_{op}} \quad (2.5)$$

En general, independientemente de la naturaleza de los elementos constituyendo las partes relativas a la seguridad de un sistema de control, los valores de su tiempo medio hasta fallo peligroso y de su número medio de ciclos hasta el fallo peligroso del 10% de los mismos pueden extraerse de tablas proporcionadas por las normas internacionales pertinentes o ser proveídos por los propios fabricantes de los productos.

2.6. Cobertura de diagnóstico (DC).

La cobertura del diagnóstico (del inglés, *Diagnostic Coverage*) es una medida de la capacidad, expresada en forma de porcentaje, de detectar un fallo peligroso por parte de un sistema o de uno de los elementos que lo constituyen. La cobertura de diagnóstico caracteriza por tanto una eficiencia que, en términos analíticos, se expresa como la relación entre la tasa de fallos peligrosos detectados y la tasa total de fallos peligrosos.

La tasa de fallos peligrosos detectados refleja la cantidad de eventos hallados que podrían derivar en una pérdida de la función de seguridad del sistema considerado, para los cuales se ha logrado devolverlo a su estado de seguridad original. La tasa total de fallos peligrosos representa el conjunto de eventos que suponen una pérdida de la función de seguridad del sistema considerado e incluye tanto los fallos que son detectados como los que no lo son.

Existen varias formas de evaluar la cobertura de diagnóstico. Uno de estos métodos corresponde a la evaluación de los riesgos mediante el análisis de los modos de fallo y sus efectos (FMEA) o a través de técnicas similares. Sin embargo, la norma EN ISO 13849-1 propone un enfoque simplificado basado en datos históricos y presenta en sus anexos valores estimados en función del tipo de dispositivo, de acuerdo a las arquitecturas designadas. La cobertura de diagnóstico se desglosa en cuatro rangos con unos valores base de 60%, 90% y 99%. La Tabla 2.2. muestra dicha clasificación.

Denotación	Rango
Nula	$DC < 60\%$
Baja	$60\% \leq DC < 90\%$
Media	$90\% \leq DC < 99\%$
Alta	$99\% \leq DC$

Tabla 2.2. Clasificación del diagnóstico de cobertura.

Para una parte relativa a la seguridad de un sistema de control integrada por distintos componentes desempeñando una misma función de seguridad, la cobertura de diagnóstico media del conjunto DC_{avg} es dada por la relación entre los tiempos medios hasta fallo peligroso y las coberturas de diagnóstico individuales de los N elementos que realizan la función de seguridad, coherente con la definición de cobertura de diagnóstico presentada al principio de esta sección.

$$DC_{avg} = \frac{\frac{DC_1}{MTTF_{d1}} + \frac{DC_2}{MTTF_{d2}} + \dots + \frac{DC_n}{MTTF_{dN}}}{\frac{1}{MTTF_{d1}} + \frac{1}{MTTF_{d2}} + \dots + \frac{1}{MTTF_{dN}}} \quad (2.6)$$

2.7. Fallo de causa común (CCF).

El fallo de causa común es aquel fallo producido en varios componentes de un mismo sistema como consecuencia de un evento individual, sin que sea resultado de la relación existente entre dichos elementos. La importancia de los fallos de causa común radica en su aparición en sistemas con estructura redundante de doble canal, donde es

posible controlar fallos aleatorios individualmente en cada uno de los canales, pero se es incapaz de actuar cuando el fallo afecta a ambos canales de comunicación a la vez. Los factores que pueden inducir tales eventos son principalmente condiciones ambientales desfavorables, interferencias electromagnéticas o sobrecargas que no son tratadas durante la fase de diseño del sistema de control. Por ello, es importante determinar si un sistema dispone o no de las medidas necesarias para reducir sus efectos. La norma EN ISO 13849-1 propone un método simplificado y cuantitativo para la estimación del grado de robustez frente a fallos de causa común del conjunto de un sistema, para el cual se ponderan distintos aspectos en base a las medidas que se han aplicado. A continuación, se presentan los aspectos abarcados; en paréntesis se indica la puntuación que corresponde a cada uno de estos ítems.

- a) Separación física del cableado y asilamiento de los caminos de las señales (15).
- b) Diversidad de la tecnología, diseños o principios físicos empleados, así como el uso de componentes de diferentes fabricantes (20).
- c) Diseño y aplicación de componentes de seguridad y experiencia en el uso de los mismos (20).
- d) Evaluación y análisis de los modos de fallo del sistema y sus efectos (5).
- e) Competencia y entrenamiento de los diseñadores y del personal de mantenimiento responsables (5).
- f) Efectos del entorno sobre el sistema, tales como filtraciones, perturbaciones electromagnéticas o eventos provocados por la acción de la temperatura, choques, vibraciones o humedad (35).

Se estima que el sistema cumple con los requisitos mínimos de seguridad y se encuentra protegido frente a fallos de causa común cuando la puntuación total del listado anterior alcanza un valor de 65, siendo 100 la puntuación máxima alcanzable en una situación ideal. En caso de no cumplir esta condición, se considera el proceso fallido y se seleccionarán medidas adicionales hasta su cumplimiento.

2.8. Probabilidad de fallo peligroso por hora (PFH_d).

La probabilidad de fallo peligroso por hora (del inglés, *Probability of Dangerous Failure per Hour*) es un indicador cuantitativo de la probabilidad media de un sistema o subsistema de padecer un fallo peligroso en el transcurso de una hora. La determinación de la probabilidad de fallo peligroso por hora de un sistema o subsistema se realiza a partir del análisis de los dispositivos que lo integran y de sus probabilidades de fallo

individuales. En general, este sistema es dividido en bloques funcionales siguiendo una arquitectura con sendos bloques de entrada, lógico y de salida, cada uno conteniendo una serie de elementos determinados cuyas probabilidades de fallo suelen estar normalmente provistas por sus mismos fabricantes. La probabilidad de fallo peligroso por hora total del sistema puede entonces evaluarse como la suma aritmética de las probabilidades de sus componentes.

$$PFH_d(\text{total}) = PFH_d(\text{entrada}) + PFH_d(\text{lógico}) + PFH_d(\text{salida}) \quad (2.7)$$

A modo de ejemplo, se presenta el sistema de seguridad de la Figura 2.6. de una máquina empaquetadora, conformado por dos botones de paro de emergencia, dos interruptores de posición, un relé de seguridad y dos contactores controlados redundantes. Para cada función de seguridad mostrada del mismo se ha calculado su correspondiente PFH_d de acuerdo con el método expuesto.

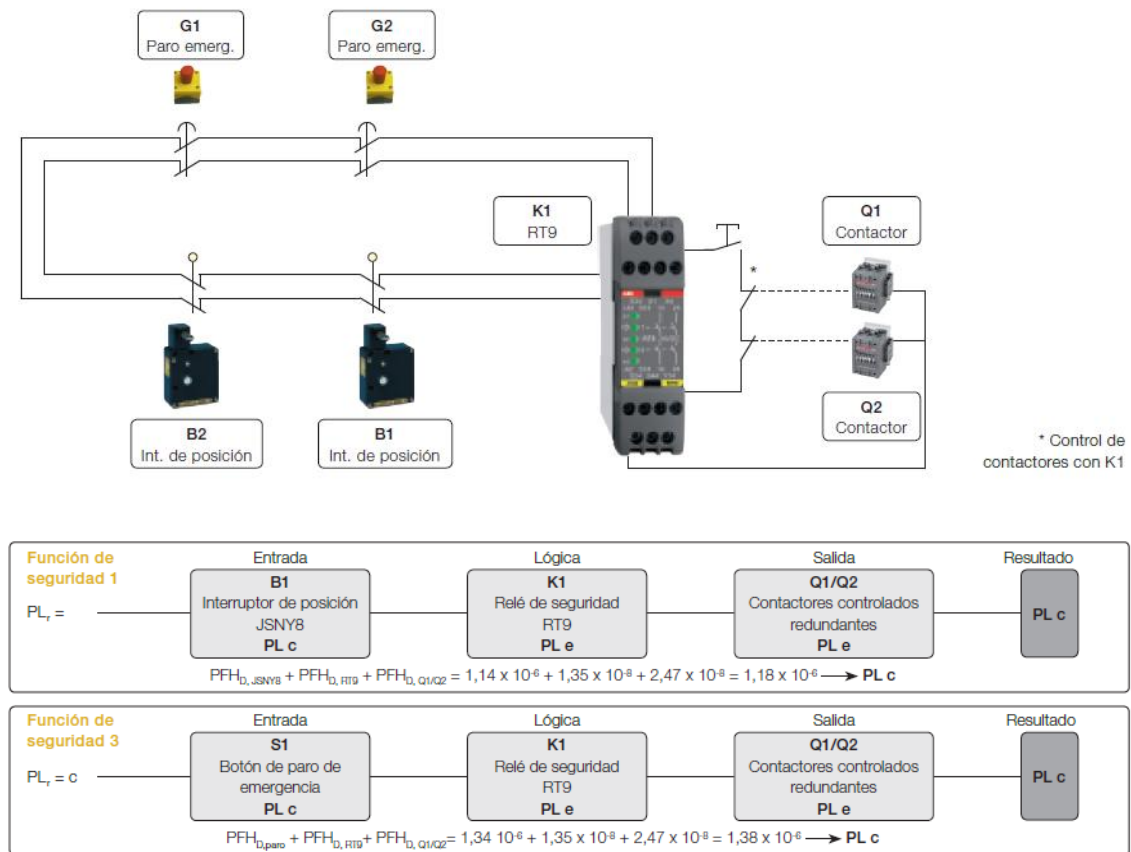


Figura 2.6. Sistema de seguridad de una máquina empaquetadora y cálculo del PFH_d total de sus funciones de seguridad.

3

Normativa EN ISO 13849-1

En el capítulo anterior se han descrito conceptos clave relacionados con la seguridad en máquinas, tales como el nivel de fiabilidad (PL) o la cobertura de diagnóstico (DC). Dada la importancia y el uso habitual de dichos términos en sistemas de control de la seguridad, durante las últimas décadas diferentes instituciones, tanto europeas como de carácter internacional, han pretendido enmarcar y delimitar su ámbito de aplicación, resultando en una serie de normativas, leyes y directivas con distinto rango de autoridad y de alcance. La norma EN ISO 13849-1 constituye un estándar aplicable a todo tipo de tecnologías reconocido en 157 países que establece las pautas necesarias para el diseño de sistemas seguros.

3.1. Introducción.

En este capítulo se exponen las principales características de la norma EN ISO 13849-1 y, en particular, el procedimiento definido por la misma seguido para el diseño seguro de sistemas, para el que se detallarán sus diversas etapas. La norma EN ISO 13849-1 fue redactada por primera vez en el año 1999 entre el Comité Europeo de Normalización (CEN) y la Organización Internacional de Normalización (ISO) a través de sendos comités y tomando como base la antigua norma EN 954-1, relativa a la seguridad de las máquinas. Este nuevo estándar surge como consecuencia de un cambio generacional en el enfoque que hasta entonces se tenía de las normas de seguridad en sistemas de control, para el cual se requerían de nuevos conceptos, cálculos y procedimientos. Junto con el estándar EN ISO 13849-1, aparece la norma EN IEC 62061, desarrollada por la Comisión Electrotécnica Internacional (IEC) y derivada de la normativa EN IEC 61508, relativa a la seguridad funcional de los sistemas eléctricos, electrónicos o electrónicos programables relacionados con la seguridad.

Tras la retirada definitiva de la norma EN 954-1 en el año 2011, los estándares EN ISO 13849-1 y EN IEC 62061 compiten por suceder a la norma obsoleta y

convertirse en el nuevo marco de referencia de la Seguridad en Máquinas. Si bien ambos se fundamentan en principios probabilísticos similares, el enfoque que implementan difiere en la práctica. El método descrito en la norma EN IEC 62061 busca aplicar funciones de seguridad complejas en sistemas con arquitectura que anteriormente no era convencional. Por el contrario, la norma EN ISO 13849-1 aplica un procedimiento más sencillo en el que se asocian funciones de seguridad más simples a sistemas con arquitectura convencional, modelizada por la propia normativa.

La capacidad de un componente de cumplir con una función de seguridad dada es conocida como nivel de fiabilidad o PL según la norma EN ISO 13849-1; sus características ya han sido descritas en la Sección 2.3. En la norma homóloga de la IEC, esta aptitud es evaluada a partir del nivel de integridad o SIL (*Safety Integrity Level*), que define el grado de reducción de riesgo que proporciona una función de seguridad. El nivel de integridad se desglosa en tres rangos: SIL1, SIL2 y SIL3 en base a un proceso numérico iterativo teniendo en cuenta la frecuencia de un evento peligroso, su probabilidad de ocurrencia y la probabilidad de evitarlo. Ambos criterios son equivalentes y comparables con la probabilidad de fallo peligroso por hora PFH_d de un sistema o subsistema. No obstante, cabe remarcar que el valor máximo de probabilidad de fallo peligroso por hora alcanzado está condicionado por el procedimiento empleado, siendo notablemente inferior al utilizar niveles de integridad como base de cálculo.

PL (EN ISO 13849-1)	SIL (EN IEC 62061)	PFH_d
a	Sin correspondencia	$\geq 10^{-5}$ a $< 10^{-4}$
b	SIL1	$\geq 3 \times 10^{-6}$ a $< 10^{-5}$
c	SIL1	$\geq 10^{-6}$ a $< 3 \times 10^{-6}$
d	SIL2	$\geq 10^{-7}$ a $< 10^{-6}$
e	SIL3	$\geq 10^{-8}$ a $< 10^{-7}$

Tabla 3.1. Equivalencia entre el nivel de fiabilidad (PL), el nivel de integridad (SIL) y la probabilidad de fallo peligroso por hora (PFH_d).

Esta equivalencia facilita, salvo particularidades de la norma, utilizar indistintamente cualquiera de los dos métodos. La excepción a esto se halla en el campo de aplicación de los estándares. En efecto, la normativa EN IEC 62061 está limitada a sistemas de mando eléctricos, electrónicos o electrónicos programables. Sin embargo, la normativa EN ISO 13849-1 abarca un espectro tecnológico más amplio, extendiéndose a sistemas de naturaleza hidráulica, neumática y mecánica. Este hecho conlleva que en la actualidad la norma EN ISO 13849-1 sea de aplicación más extendida, reservándose la norma EN IEC 62061 para aquellos casos en que el uso de la primera es insuficiente.

Tecnología	EN ISO 13849-1	EN IEC 62061
Hidráulica	Aplicable	No aplicable
Neumática	Aplicable	No aplicable
Mecánica	Aplicable	No aplicable
Eléctrica	Aplicable	Aplicable
Electrónica	Aplicable	Aplicable
Electrónica programable	Aplicable	Aplicable

Tabla 3.2. Ámbito de aplicación de las normativas EN ISO 13849-1 y EN IEC 62061.

El capítulo se organiza de la siguiente manera. La Sección 3.2. presenta el esquema general propuesto por la norma EN ISO 13849-1 para el diseño de los medios de seguridad requeridos en un sistema y la validación del nivel de seguridad resultante del mismo. Los apartados que se derivan a continuación pretenden caracterizar cada uno de los pasos mostrados y detallar, en la medida de lo posible, las consideraciones y procesos efectuados en cada caso.

3.2. Diseño de la seguridad.

El proceso de diseño de la seguridad de un sistema según la norma EN ISO 13849-1 pasa inevitablemente por la evaluación del riesgo de acuerdo con los procedimientos descritos en los estándares ISO 12100 e ISO 14121, relativos a los términos, metodología y principios técnicos de diseño y a los principios de evaluación del riesgo, respectivamente. Según dicho proceso, previo a la etapa de reducción de riesgos, es preciso determinar los límites de la máquina, así como hallar los peligros, estimar los riesgos inherentes a éstos y evaluarlos. Los riesgos extraídos de la evaluación pueden entonces eliminarse o paliarse de tres maneras diferentes, siguiendo un orden de implantación que atiende al riesgo residual que permanece tras su aplicación o a su éxito fallido tras el mismo: a) mediante la modificación del diseño intrínseco del sistema durante su fase de diseño, b) mediante la aplicación de dispositivos de protección y medidas preventivas suplementarias en el sistema, o c) mediante la información de los usuarios y su protección a través de equipos de protección individuales (EPI).

La dependencia de las medidas protectivas seleccionadas a un sistema de control conlleva la ejecución de un proceso iterativo de seis etapas, detalladas en la norma EN ISO 13849-1, que constituyen el proceso de diseño de sus partes relativas a la seguridad. El esquema del proceso se ilustra en la Figura 3.1. Tal y como se indica en éste, los pasos seguidos, por orden de aplicación, son:

1. Identificación de las funciones de seguridad y especificación de sus características.

2. Determinación del nivel de fiabilidad requerido de la función de seguridad.
3. Diseño y ejecución técnica de la función de seguridad.
4. Estimación del nivel de fiabilidad PL obtenido.
5. Verificación del nivel de fiabilidad de la función de seguridad.
6. Validación de los requisitos de la función de seguridad.

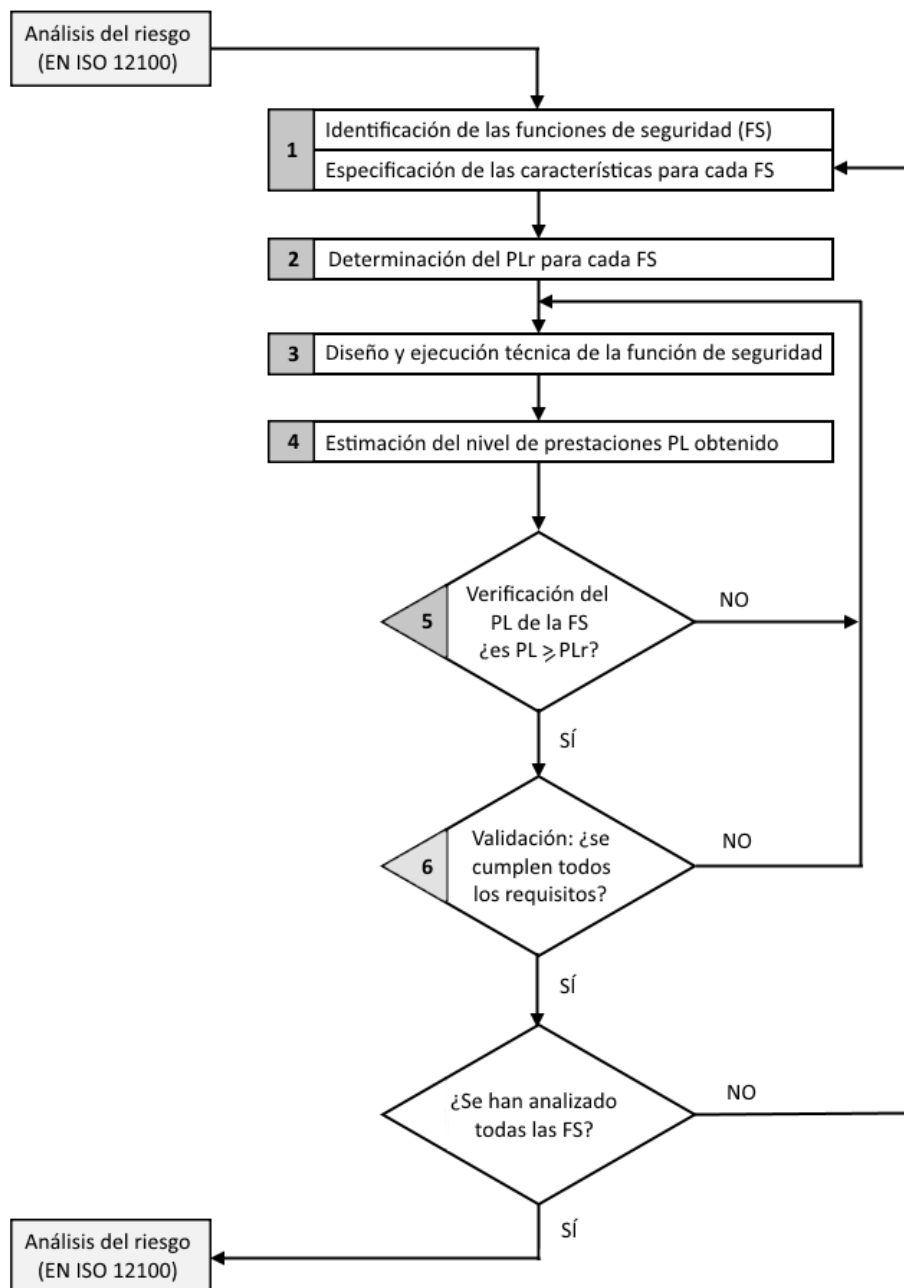


Figura 3.1. Esquema general del proceso de diseño de la seguridad de un sistema, según la norma EN ISO 13849-1.

Para cada una de las funciones de seguridad identificadas en el primer punto listado, las etapas que siguen se ejecutarán cíclicamente hasta satisfacer los requerimientos de seguridad mínimos establecidos por la normativa en los pasos 5 y 6. A continuación, se profundiza en cada uno de los seis puntos enumerados del proceso de diseño.

3.2.1. Definición de las funciones de seguridad y especificación de sus características.

Una vez delimitado el alcance del sistema para el que se desea diseñar la seguridad e identificados y evaluados sus riesgos, debe determinarse el conjunto de funciones de seguridad que restringirán su acontecimiento. Una función de seguridad se define como aquella función de la máquina cuyo objetivo es mantener el equipo bajo control y en un estado de seguridad ante una situación de peligro; un fallo en su ejecución podría suponer el incremento inminente del riesgo en la propia máquina. Todo riesgo que no ha sido eliminado durante la fase de diseño del sistema tiene asociada una función de seguridad que debe ser suplida por una o varias partes relativas a la seguridad de su sistema de control. Del mismo modo, varias funciones de seguridad pueden compartir una o más partes relativas a la seguridad que las ejecuten. La imagen de la Figura 3.2. constituye un ejemplo de aplicación de varias funciones de seguridad en un mismo sistema, cada una de ellas implementada por un dispositivo diferente.

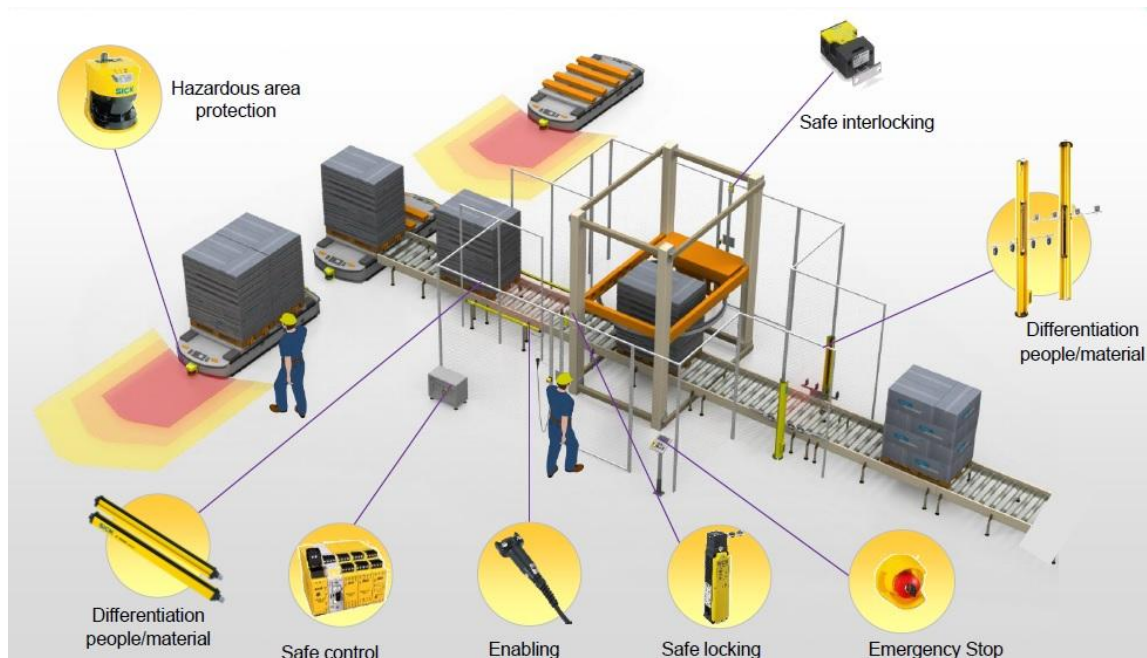


Figura 3.2. Aplicación de varias funciones de seguridad en un mismo sistema.

Las funciones de seguridad que se considerarán en el marco del presente proyecto son las siguientes:

- a) Impedir el acceso de manera permanente.
- b) Impedir el acceso de manera temporal.
- c) Detener piezas, sustancias y radiaciones.
- d) Provocar la parada.
- e) Evitar un arranque intempestivo.
- f) Impedir el arranque.
- g) Combinación: provocar la parada e impedir el arranque.
- h) Permitir el paso de materiales.
- i) Monitorizar los parámetros de la máquina.
- j) Desactivar funciones de seguridad manual y temporalmente de forma limitada.
- k) Parada en caso de emergencia.
- l) Indicadores y alarmas relevantes para la seguridad.

La mayoría de las máquinas pueden adoptar distintos estados o trabajar según diversos modos de operación. Por este motivo, es posible combinar las funciones de seguridad listadas entre ellas con el fin de adaptarse a la nueva realidad funcional y acondicionarse a los nuevos requerimientos de seguridad del conjunto.

Las características de cada función de seguridad deben estar claramente especificadas para el diseño apropiado de las partes relativas a la seguridad que las integran. En general, debe ofrecerse una descripción de la función de seguridad, precisando su tiempo de respuesta, así como las condiciones bajo las que ésta se dispara y permanece activa. Por otro lado, desde el punto de vista de la máquina, debe disponerse igualmente de una descripción de su funcionamiento, de sus parámetros funcionales y del comportamiento de la misma cuando la función de seguridad se encuentra activa, al igual que la frecuencia con la que dicha función de seguridad es demandada. La descripción de la función de seguridad quedará completada una vez se especifique su nivel de fiabilidad requerido; el método empleado para la evaluación de este parámetro se expone en el siguiente apartado.

3.2.2. Determinación del nivel de fiabilidad requerido.

Toda función de seguridad se caracteriza por un determinado nivel de fiabilidad o de prestación indicando su capacidad para ejecutarse satisfactoriamente ante unas condiciones de operación predecibles. El cumplimiento de una función de seguridad se alcanza a través de la integración de diversos dispositivos en el sistema de control que, de manera conjunta, permiten alcanzar los requisitos mínimos de seguridad de la propia función de seguridad. Esta condición mínima es dada por el nivel de prestación requerido PL_r . La norma EN ISO 13849-1 proporciona una metodología sencilla para la evaluación de dicho parámetro fundamentada en un árbol de decisiones como el mostrado en la Figura 3.3.

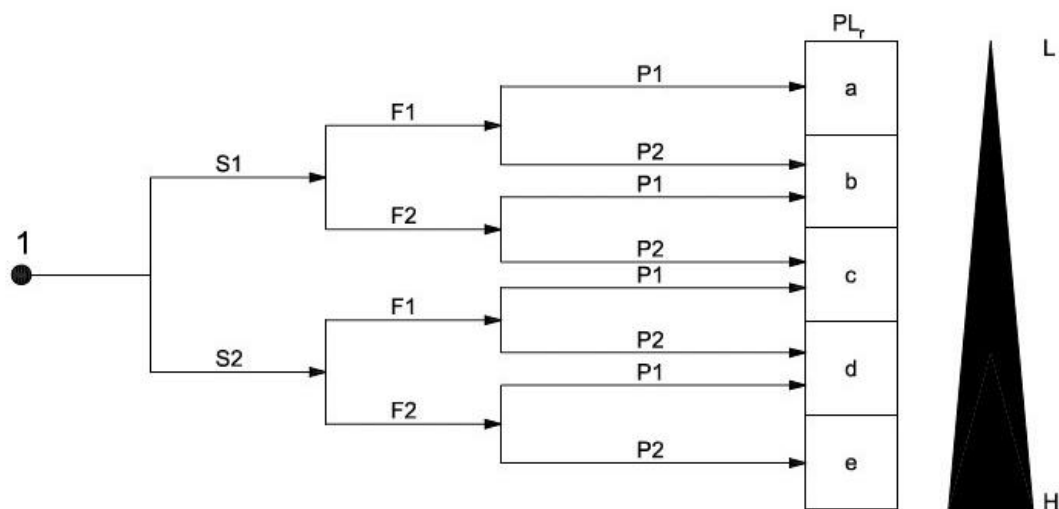


Figura 3.3. Árbol de decisiones establecido por la norma EN ISO 13849-1.

El nivel de fiabilidad requerido depende directamente del riesgo para el cual se implementa. Este riesgo se define en base a tres índices, cada uno de los cuales pudiendo reflejar dos estados o situaciones diferentes u opuestos: la severidad del daño (S), estimada en función de las consecuencias del accidente y del tiempo de recuperación que le sigue; la frecuencia y/o tiempo de exposición (F), estimada en función de la recurrencia y del tiempo de sometimiento a una situación de peligro con respecto del tiempo total de utilización de la máquina o sistema; y la probabilidad de evitar o de reducir el peligro (P), estimada en función de la posibilidad de detectar un peligro a través de sus características o por el intermedio de dispositivos (sensores). En la Tabla 3.3. se recogen las distintas opciones que abarcan estos índices.

S	Severidad.	S1	Daño leve (normalmente reversible o reparable).
		S2	Daño grave (normalmente irreversible o irreparable).

F	Frecuencia y/o duración de la exposición al peligro.	F1	Raro a frecuente y/o corta duración de exposición.
		F2	Frecuente a continuo y/o larga duración de exposición.
P	Probabilidad de evitar el peligro o de limitar el daño.	P1	Posible en determinadas condiciones.
		P2	Raramente posible.

Tabla 3.3. Parámetros definitorios de un riesgo.

Partiendo del punto de inicio con el número 1 y siguiendo el conjunto de itinerarios posibles (ver Figura 3.3.), la combinación de los tres índices anteriores con sus dos casos cada uno supone la delimitación de cinco escenarios diferentes adoptables por la función de seguridad. Cada una de estas cinco situaciones tendrá por tanto asignada una letra de acuerdo con la escala alfabética del nivel de fiabilidad, descrita en la Sección 2.3. Un nivel de prestación “a” conlleva una baja contribución a la reducción del riesgo por parte de la función de seguridad. Por el contrario, un nivel de prestación “e” supone una contribución elevada a la reducción del mismo. Este hecho puede corroborarse a partir del gradiente que se observa a la derecha del árbol de decisiones presentado, donde en su base se sitúa una participación alta (H o *high*) y en su extremo superior una participación baja (L o *low*).

3.2.3. Diseño y ejecución técnica de la función de seguridad.

La verificación de las funciones de seguridad debe llevarse a cabo mediante un proceso iterativo de selección y comprobación de la tecnología a instalar que asegure la funcionalidad de los conceptos de seguridad y su integración adecuada en el sistema, al mismo tiempo que evite el surgimiento de nuevos riesgos con su inclusión. Los factores que se tienen en cuenta durante esta etapa son, entre otros: a) las características físicas y funcionales de la máquina y su diseño; b) las características del entorno tales como vibraciones, rangos de temperatura, choques o perturbaciones electromagnéticas; c) los aspectos humanos, como el número de personas que se estima en el área de operación o su formación; y d) las características de los elementos a implementar. En última instancia, se dispondrá de un sistema integrado por una serie de elementos cubriendo los requerimientos de seguridad establecidos en la fase anterior y organizados según una estructura de canal individual o doble para hacer frente a los defectos de su propia arquitectura y garantizar el comportamiento funcional que se requiere de ellos. La Tabla 3.4. recoge algunos ejemplos de dispositivos en los que pueden materializarse las funciones de seguridad presentadas en el Apartado 3.2.1.

Función de seguridad		Dispositivo asociado
a	Impedir el acceso de manera permanente.	Panel, barrera.
		Resguardo físico o mecánico.
b	Impedir el acceso de manera temporal.	Panel, barrera.

		Resguardo físico o mecánico.
c	Detener piezas, sustancias y radiaciones.	Cubierta de protección con ventana, valla.
d	Provocar la parada.	Puerta protectora con dispositivo de bloqueo sin fijación.
		Barrera fotoeléctrica de seguridad multihaz.
e	Evitar un arranque intempestivo.	Pulsador de validación.
f	Impedir el arranque.	Sistema de consignación mediante llave.
		Cortina de seguridad.
g	Combinación: provocar la parada e impedir el arranque.	Cortina fotoeléctrica.
		Escáner de seguridad.
h	Permitir el paso de materiales.	Cortina fotoeléctricas horizontal.
		Escáner láser de seguridad.
i	Monitorizar los parámetros de la máquina.	Sensor (de posición, velocidad, aceleración, temperatura, presión, etc.).
j	Desactivar funciones de seguridad manual y temporalmente de forma limitada.	Pulsador de validación.
k	Parada en caso de emergencia.	Interruptor de presión de parada de emergencia.
l	Indicadores y alarmas relevantes para la seguridad.	Indicador de bloqueo.
		Dispositivo de advertencia de arranque.
		Luz indicadora.

Tabla 3.4. Ejemplos de funciones de seguridad y dispositivos capaces de ejecutarlas.

3.2.4. Estimación del nivel de fiabilidad obtenido.

Después de seleccionar los dispositivos pertinentes responsables de asegurar una función de seguridad dada, es preciso evaluar el nivel de fiabilidad alcanzado con su implementación con el fin de poder comprobar posteriormente que se garantizan los requerimientos de seguridad planteados al inicio del proceso de diseño. Existen principalmente tres enfoques a la hora de estimar el nivel de fiabilidad obtenido en la fase de diseño anterior. El primero de ellos consiste en una estimación de carácter tabular del nivel de fiabilidad de la función de seguridad en base a los niveles de prestación individuales de los subsistemas que la conforman. Un segundo método se fundamenta en la adición de las probabilidades de fallo peligroso por hora particulares de los mismos subsistemas, conformando la probabilidad de fallo peligroso por hora total de la función de seguridad que representa su conjunto. A pesar de la versatilidad de ambos procesos, no se contempla su aplicación en aquellos casos en que se carece de la información adecuada de los elementos considerados. Por este motivo, la norma EN ISO 13849-1 propone un tercer procedimiento alternativo y complementario a los dos anteriores que permite evaluar el nivel de fiabilidad de un sistema o subsistema a partir de la combinación de cuatro parámetros: la categoría del sistema o subsistema (Cat.), el tiempo medio hasta fallo peligroso ($MTTF_d$), la cobertura de diagnóstico (DC) y, en el

caso de las categorías 2, 3 y 4, el grado de robustez del diseño frente a los fallos de causa común (CCF).

La Figura 3.4. presenta el gráfico en que se asienta esta tercera metodología. El esquema muestra las relaciones existentes entre la categoría, el tiempo medio hasta fallo peligroso y la cobertura de diagnóstico; la fortaleza del diseño frente a los fallos de causa común requiere de un análisis separado. La interacción de estas tres variables impone nuevos intervalos de valores admisibles de las mismas, caracterizando igualmente un rango de niveles de fiabilidad alcanzables por el sistema o subsistema en cada caso. En el eje de ordenadas se representan los valores discretos de nivel de prestación alcanzable, por un lado, y los valores equivalentes de probabilidad de fallo peligroso por hora, por el otro. En el eje de coordenadas se representa la combinación entre la categoría y la cobertura de diagnóstico. Mientras que categorías de clase B ó 1 no tienen un valor de cobertura de diagnóstico asignado, arquitecturas de orden superior permiten hasta dos tipos de cobertura posibles (baja o media), con excepción de la categoría 4, para la cual, dada su naturaleza, sólo una capacidad de detección de fallo alta es permitida. Los diferentes valores de tiempo medio hasta fallo peligroso se reflejan en sendas barras verticales para cada combinación de categoría y cobertura de diagnóstico representada.

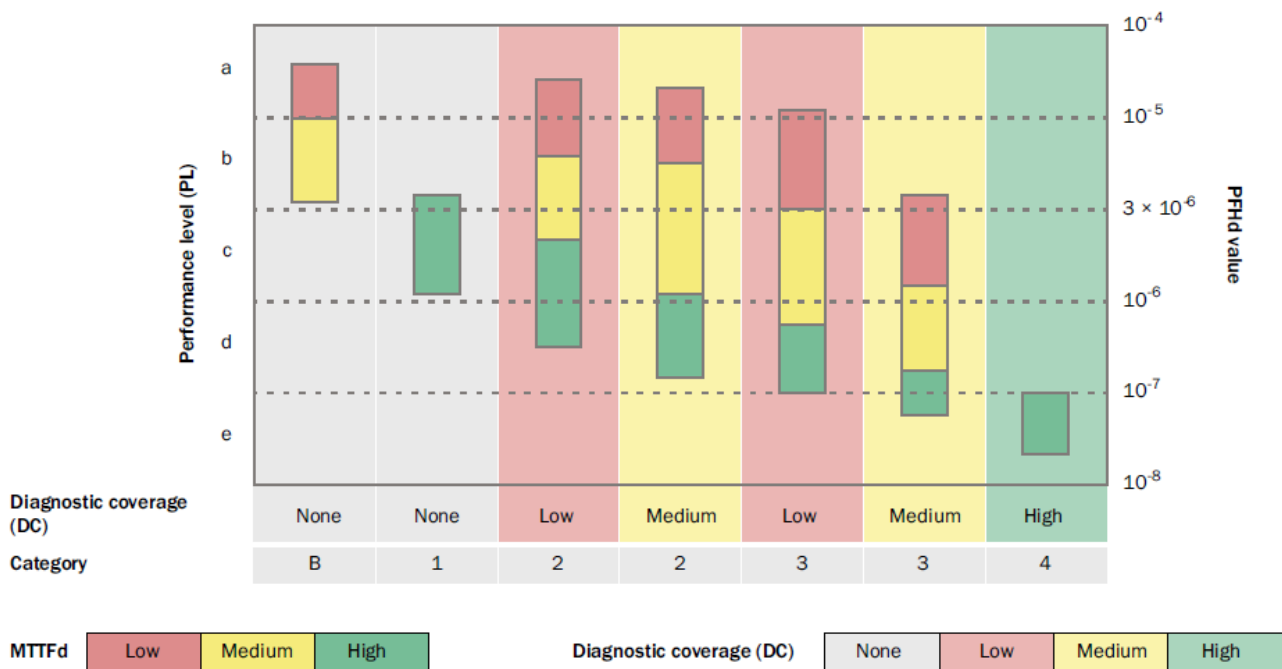


Figura 3.4. Relación entre categoría, tiempo medio hasta fallo peligroso y cobertura de diagnóstico para cada nivel de fiabilidad.

En la Tabla 3.5. se resume el conjunto de relaciones posibles entre las distintas variables mencionadas e ilustradas en el gráfico anterior.

Categoría		<i>B</i>	<i>1</i>	<i>2</i>	<i>2</i>	<i>3</i>	<i>3</i>	<i>4</i>
DC _{avg}		<i>ninguna</i>	<i>ninguna</i>	<i>baja</i>	<i>media</i>	<i>baja</i>	<i>media</i>	<i>alta</i>
MTTF _d (por canal)	<i>bajo</i>	a	-	a	b	b	c	-
	<i>medio</i>	b	-	b	c	c	d	-
	<i>alto</i>	-	c	c	d	d	d	e

Tabla 3.5. Procedimiento simplificado para la evaluación del nivel de fiabilidad.

Para valores más precisos del nivel de fiabilidad en función del tiempo medio hasta fallo peligroso de cada canal, remitirse a la Tabla A2.1. en el Anexo A.

3.2.5. Verificación del nivel de fiabilidad del sistema.

Como ya se ha indicado en el apartado anterior, la verificación del cumplimiento de la seguridad funcional alcanzada con los elementos seleccionados es llevada a cabo en términos del nivel de fiabilidad global de la función de seguridad diseñada y del sistema sobre la cual actúa. La condición de obligado cumplimiento exige que, para toda parte relativa a la seguridad de un sistema de control o combinación de las mismas definiendo una única función de seguridad, el nivel de fiabilidad global de ésta (PL) debe ser igual o superior al nivel de fiabilidad requerido por el sistema o subsistema (PL_r) que la implementa. En caso de no cumplirse este requisito, debe procederse a un rediseño de la función de seguridad y a la subsiguiente reevaluación de su nivel de prestación.

Junto con el nivel de fiabilidad, también deben verificarse aspectos adicionales de la función de seguridad tales como la efectividad o la ergonomía de su diseño (forma, color, estética o manipulación, entre otros). Estas tareas suelen ser realizadas por el fabricante o el instalador de las medidas protectoras de forma paralela al proceso de ejecución, generalmente a través de la cumplimentación de formularios siguiendo un formato de checklist.

3.2.6. Validación de los requisitos de la función de seguridad.

El procedimiento de validación tiene como objetivo comprobar la conformidad del conjunto de partes relativas a la seguridad integradas en una función de seguridad con respecto de los requisitos de seguridad establecidos en la norma EN ISO 13849-1 y las especificaciones impuestas durante su fase de diseño. La validación de la función de seguridad se realiza a través de diversos procesos complementarios. Destacan: la verificación técnica del emplazamiento y la efectividad de los elementos instalados, las comprobaciones a nivel práctico del comportamiento de dichos dispositivos ante situaciones de peligro y resultados teóricos obtenidos de simulaciones, y la validación

de los requerimientos ambientales (temperatura, humedad, vibraciones, interferencias electromagnéticas, etc.) mediante ensayos funcionales. Estas tareas suelen ser efectuadas de forma paralela al proceso de ejecución por personas ajenas a la construcción de los componentes de seguridad. En caso de no cumplir con los requisitos, debe procederse a un rediseño de la función de seguridad y a la subsiguiente reevaluación de su nivel de prestación.

3.3. Información adicional.

La exposición al tiempo de las partes relativas a la seguridad de un sistema de control supone un deterioro continuo de la seguridad de la máquina, sus usuarios y su entorno, así como la inducción de situaciones peligrosas adicionales. Este hecho conlleva que se provea de manera indispensable, junto con los planos de diseño del sistema resultante, la documentación necesaria para mantener y asegurar su funcionamiento específico. Esta información consta de tres documentos básicos: a) instrucciones para la consecución de operaciones de mantenimiento preventivo y correctivo, en las que se incluyen inspecciones periódicas de los elementos; b) documentación técnica de las características de los dispositivos, donde se especifica su nivel de fiabilidad y tiempo medio hasta fallo peligroso, entre otros, así como las medidas de seguridad a tomar en caso de un uso incorrecto de los mismos; y c) información para el uso seguro de los dispositivos por parte de los usuarios, donde se detallan especificaciones tales como los límites de operación, indicadores, alarmas u otros modos de control.

4

Metodología propuesta

La norma EN ISO 13849-1 constituye un estándar de carácter disruptivo con respecto de sus predecesores. Dadas las implicaciones prácticas de la norma en los procesos industriales, es precisa la existencia de herramientas que permitan la aplicación de sus disposiciones de una manera sencilla e inmediata. Si bien en la actualidad se hallan diversos programas que satisfacen dichas funciones, existen varios enfoques que aún no han sido abordados. En el presente capítulo se pretende dar respuesta a estas necesidades no cubiertas con la propuesta de una metodología en vista a ser implementada posteriormente en la forma de un programa informático para su fácil distribución y explotación por las partes interesadas.

4.1. Introducción.

La implantación de la norma EN ISO 13849-1 ha conllevado desde su aparición en 2011 la puesta simultánea en el mercado de nuevas herramientas informáticas capaces de asistir a los diseñadores en las tareas de modelización y simulación de los sistemas industriales, tomando como base los preceptos descritos en el nuevo estándar. Éste es el caso de programas tales como SISTEMA (del inglés, *Safety Integrity Software Tool for the Evaluation of Machine Applications*), del Instituto alemán de salud ocupacional (IFA), o PAScal, de la empresa alemana Pilz. SISTEMA es una herramienta gratuita para el modelado de la estructura de los componentes de control de seguridad a partir de su arquitectura designada. Los distintos parámetros relativos a los riesgos o a la fiabilidad son introducidos en el interfaz de usuario, donde los cálculos se encuentran automatizados con diferentes niveles de detalle y el modelo es actualizado de manera instantánea con cada modificación realizada. La información puede ser ingresada manualmente o automáticamente importando una biblioteca de datos proveída por el fabricante de componentes. De igual modo, PAScal es un programa gratuito que permite la determinación del nivel de fiabilidad (según la norma EN ISO 13849-1) o del nivel de integridad (según la norma IEC 62061) de funciones de seguridad y de instalaciones en base a los componentes empleados. Para ello, utiliza un interfaz gráfico

permitiendo visualizar en qué grado es alcanzado dicho nivel de prestación y con qué componentes, indicando qué mejoras son necesarias implementar.

A pesar de la utilidad de los programas mencionados en lo que respecta a la modelización y simulación de los sistemas de seguridad, se carece de una herramienta específica para la selección de los dispositivos de seguridad integrados en dichos sistemas a partir de los requisitos que la naturaleza de sus funciones de seguridad les impone. En efecto, los softwares actuales siguen el procedimiento detallado por la norma EN ISO 13849-1 (ver Figura 3.1.) y parten de modelos que incorporan desde sus etapas iniciales los componentes de seguridad, de los que se extraen los correspondientes parámetros de cálculo utilizados en la evaluación del nivel de prestación. Sin embargo, lo que se propone en el presente documento es la elaboración de una herramienta que siga un procedimiento opuesto al descrito, esto es, partir de un modelo caracterizado por un determinado nivel de fiabilidad obtenido de la evaluación de las características de la función de seguridad, del cual se derivan los componentes que se precisan instalar para garantizar los requerimientos de seguridad iniciales.

El capítulo se organiza de la siguiente manera. La Sección 4.2. presenta el esquema general propuesto para el diseño de los sistemas de seguridad a partir de los requerimientos de seguridad impuestos por el entorno. Los apartados que se derivan a continuación pretenden caracterizar cada uno de los pasos mostrados y detallar, en la medida de lo posible, las consideraciones y procesos efectuados en cada caso.

4.2. Diseño del sistema de seguridad.

La metodología propuesta para la selección de los dispositivos de seguridad de un sistema de control cualquiera atiende a un procedimiento iterativo de siete etapas que toma como punto de partida la evaluación de los riesgos según las indicaciones de los estándares ISO 12100 e ISO 14121, junto con las subsiguientes acciones correctivas que se derivan del análisis. Como ocurre en la norma EN ISO 13849-1, el procedimiento de diseño expuesto abarca solamente la selección de aquellos elementos que permitan eliminar o reducir los riesgos residuales que permanecen tras la aplicación de dichas medidas de seguridad, de cuya información se delimitan y extraen las funciones de seguridad pertinentes. El esquema del proceso se ilustra en la Figura 4.1. Tal y como se indica en éste, los pasos seguidos, por orden de aplicación, son:

1. Identificación de las funciones de seguridad y especificación de sus características.
2. Determinación del nivel de fiabilidad requerido de la función de seguridad.
3. Asignación del nivel de fiabilidad PL de la función de seguridad.

4. Determinación del nivel de fiabilidad PL del SRP/CS.
5. Determinación del $MTTF_d$ del SRP/CS.
6. Evaluación de las características de los SRP/CS.
7. Selección del componente de seguridad.

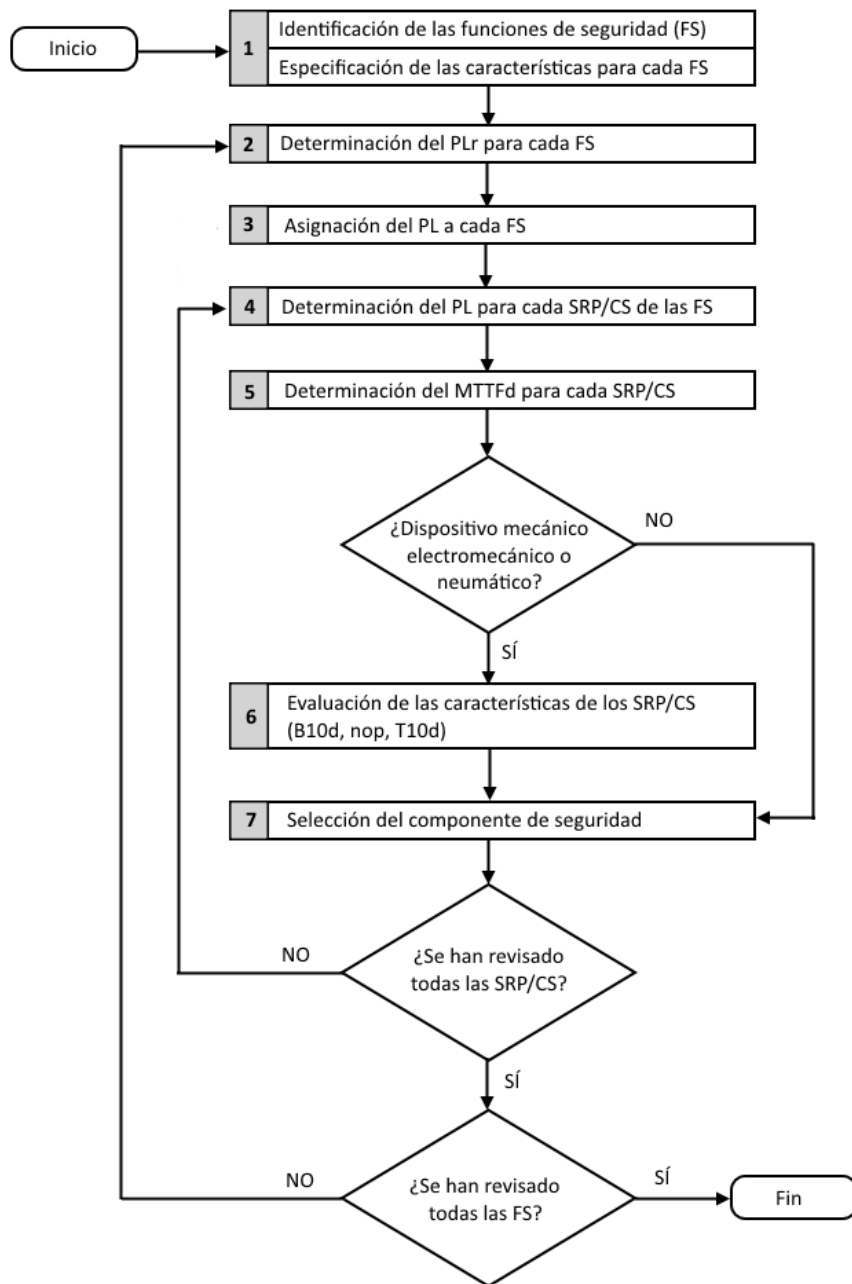


Figura 4.1. Esquema general de la metodología propuesta, basada en la norma EN ISO 13849-1.

Para cada una de las funciones de seguridad identificadas en el primer punto listado, las etapas que siguen se ejecutarán cíclicamente hasta definir los componentes de seguridad de todas las partes relativas a la seguridad del sistema de control que las ejecutan. A continuación, se profundiza en cada uno de los siete puntos enumerados del proceso de diseño.

4.2.1. Identificación de las funciones de seguridad y especificación de sus características.

El apartado “Identificación de las funciones de seguridad y especificación de sus características” ya ha sido descrito en el Apartado 3.2.1. del capítulo anterior. En él, las funciones de seguridad requeridas por un sistema cualquiera son determinadas en base a los riesgos que no han sido eliminados durante la fase de diseño del mismo. Una función de seguridad debe ser llevada a cabo por al menos una parte relativa a la seguridad del sistema de control de la estructura, pudiendo haber solapamientos entre dispositivos de seguridad ejecutando una misma función de seguridad o combinaciones entre éstos para abarcar distintos estados o modos de funcionamiento del conjunto. Se han distinguido hasta doce funciones de seguridad diferentes que son aplicables en el marco del presente proyecto. Las características de cada una de estas funciones de seguridad deben estar delimitadas con respecto de la parte relativa a la seguridad que la ejecuta (descripción de la función, tiempo de respuesta y condiciones de disparo y activación) y de la máquina donde se integra (descripción del funcionamiento de la máquina, parámetros funcionales, frecuencia de demanda de la función de seguridad y comportamiento cuando ésta se encuentra activa). La definición del nivel de fiabilidad requerido que completa la descripción de la función de seguridad se expone en el apartado que se presenta a continuación.

4.2.2. Determinación del nivel de fiabilidad requerido de la función de seguridad.

El apartado “Determinación del nivel de fiabilidad requerido de la función de seguridad” ya ha sido descrito en el Apartado 3.2.2. del capítulo anterior. En él, el nivel de fiabilidad requerido de un sistema es evaluado a partir de la aplicación de un árbol de decisiones donde interactúan tres índices: la severidad del daño (S), la frecuencia y/o tiempo de exposición (F) y la probabilidad de evitar o de reducir el peligro (P). Estos tres índices pueden representar dos situaciones distintas cada uno, de tal forma que su combinación supone la definición de hasta cinco escenarios diferentes caracterizados por un nivel de prestación específico y un rango de contribución a la reducción del riesgo por parte de la función de seguridad. El nivel de fiabilidad obtenido tiene asignada una letra de una escala alfabética, que lo representa.

4.2.3. Asignación del nivel de fiabilidad PL de la función de seguridad.

Como se indicaba en el apartado 3.2.5. “Verificación del nivel de fiabilidad del sistema”, el requisito que asegura el cumplimiento de la seguridad de un sistema cualquiera es aquél que establece que su nivel de fiabilidad (PL), obtenido tras la implementación de los dispositivos de seguridad pertinentes, sea igual o superior al nivel de fiabilidad requerido (PL_r), determinado en el apartado anterior a través del procedimiento detallado. Por este motivo, el requerimiento mínimo de obligado cumplimiento corresponde con la igualdad entre ambos niveles de fiabilidad, teórico y práctico. Esta condición es por tanto la que se tendrá en consideración en los pasos subsiguientes del proceso de diseño, al representar la situación más restrictiva de todas.

4.2.4. Determinación del nivel de fiabilidad PL del SRP/CS.

Una vez hallado el nivel de fiabilidad de un sistema de control, es preciso determinar la contribución en éste de cada una de las partes relativas a la seguridad que lo conforman. Dado que el nivel de fiabilidad atiende a una escala alfabética, resulta imposible operar con ella y, por tanto, se requiere de una transformación de la escala alfabética mencionada a una nueva de carácter numérico, con la que sí podrá alcanzarse el objetivo perseguido. En efecto, como ya se ha comentado en la Sección 3.1., existe una equivalencia entre el nivel de prestación de un sistema o subsistema y su probabilidad de fallo peligroso por hora (PFH_d); esta relación se hace notable en el gráfico de la Figura 3.4., del mismo capítulo. No obstante, puede también observarse que para un mismo nivel de fiabilidad existe un amplio rango de valores admisibles de probabilidad de fallo peligroso por hora. La amplitud y las magnitudes máxima y mínima alcanzadas por este espectro, además, se ven limitadas por la interacción de otros tres parámetros del sistema: la categoría, la cobertura de diagnóstico y el tiempo medio hasta fallo peligroso.

Con el fin de asignar un valor de probabilidad de fallo peligroso por hora a un sistema de control cualquiera, dado un nivel de fiabilidad específico, y poder fraccionar dicha probabilidad entre sus componentes de acuerdo a su aportación a la seguridad global, es necesario partir de unas consideraciones iniciales para cada uno los tres parámetros indicados. El establecimiento de un criterio acertado de selección para éstos permitirá en última instancia abordar el objetivo de partida de una manera más precisa.

Consideración 1: categoría.

Como comentado en la Sección 2.4., existen dos clases básicas de arquitectura en función del número de canales que integran: de canal individual y de doble canal. Si bien todo elemento de un sistema o subsistema tiene asociada una probabilidad de fallo

peligroso por hora, la manipulación de varias de éstas solamente es posible cuando dichos componentes se encuentran dispuestos en serie en un único canal; sistemas o subsistemas de doble canal no se encuentran contemplados. En este segundo caso, puede optarse por el empleo de una herramienta alternativa que sea capaz de operar independientemente de la estructura seleccionada. El empleo del tiempo medio hasta fallo peligroso ($MTTF_d$) del sistema o subsistema permite abordar adecuadamente este problema. Sin embargo, su manejo, aunque no complicado, puede resultar confuso a largo plazo. La opción más sencilla, propuesta para la metodología que se expone en este capítulo, se halla en desglosar las arquitecturas de doble canal en sendas estructuras de canal individual y trabajar de manera separada cada una de ellas por el intermedio de sus respectivas probabilidades de fallo peligroso por hora.

Se distinguen hasta cuatro construcciones diferentes en que pueden encontrarse los componentes de un sistema o subsistema. A continuación, se detalla la manera en que cada una de estas estructuras debe ser tratada.

Situación 1. Corresponde a la arquitectura más simple, constituida por un dispositivo de entrada, una unidad lógica y un dispositivo de salida dispuestos en serie en un único canal. Cada uno de los elementos del sistema es caracterizado por una probabilidad de fallo peligroso por hora específica. En última instancia, se pretende que todas las situaciones presentadas a continuación se adapten en mayor o menor medida a un esquema similar al de la Situación 1, ilustrado en la Figura 4.2, lo que permitirá simplificar el estudio de los sistemas de control.

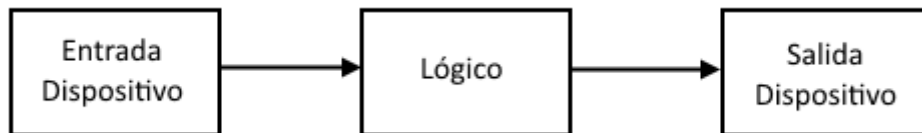


Figura 4.2. Esquema lógico de un sistema correspondiente a la Situación 1.

Situación 2. Corresponde a una arquitectura en la que los componentes se disponen en un único canal, como mostrado en la situación anterior, pero a diferencia de ésta, dos o más dispositivos ejecutan una misma función de entrada, lógica o de salida. El tratamiento aplicado es diferente según la colocación de los elementos afectados.

Cuando los componentes realizando una misma función se distribuyen en serie, dichos elementos se agrupan en una sola parte relativa a la seguridad y la forma final del sistema deviene similar a la indicada en la Figura 4.2. La probabilidad de fallo peligroso por hora de la parte compuesta resultante corresponde a la suma de las probabilidades individuales de los elementos que la integran. Esto comporta que, posterior al análisis efectuado, puede realizarse un desglose de la nueva función en sus componentes

constituyentes con el objetivo de alcanzar un mayor nivel de detalle y de variedad en la selección de los dispositivos, distribuyendo el requerimiento de probabilidad de fallo peligroso por hora entre varios instrumentos en lugar de uno solo. El esquema lógico de la Figura 4.3. muestra la transformación de un sistema con dos dispositivos de entrada en serie en un sistema de control equivalente con un solo elemento de entrada, formado por la combinación de los dos anteriores.

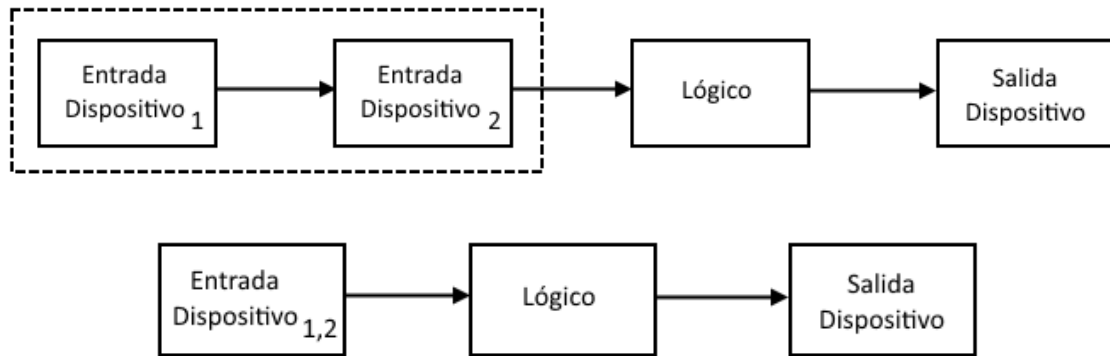
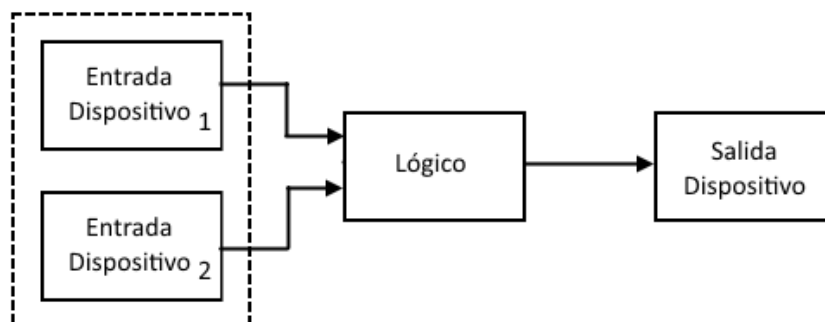


Figura 4.3. Esquema lógico de un sistema correspondiente a la Situación 2, con dos dispositivos de entrada en serie.

Cuando los componentes realizando una misma función se distribuyen en paralelo, el sistema de control se divide en tantos sistemas como elementos de dicha función contenga, obteniendo finalmente una serie de estructuras similares a la indicada en la Figura 4.2. Contrariamente a lo que ocurría antes, la probabilidad de fallo peligroso de los elementos afectados no se ve alterada y se mantiene constante a lo largo de la transformación. El esquema lógico de la Figura 4.4. muestra la partición de un sistema con dos dispositivos de entrada en paralelo en dos sistemas de control equivalentes, cada uno de las cuales integrando uno de los elementos de entrada anteriores.



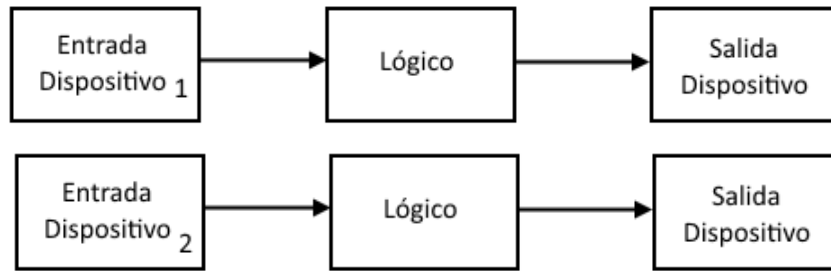


Figura 4.4. Esquema lógico de un sistema correspondiente a la Situación 2, con dos dispositivos de entrada en paralelo.

Situación 3. Corresponde a una arquitectura en la que los componentes se disponen en dos canales conteniendo, cada uno, un dispositivo de entrada, una unidad lógica y un dispositivo de salida. La simplificación del sistema de doble canal se lleva a cabo a través de la división del mismo en dos canales individuales siguiendo una estructura como la ilustrada en la Figura 4.2. Como ocurre en el segundo caso expuesto en la Situación 2, la probabilidad de fallo peligroso de cada uno de los elementos del sistema se mantiene constante y no sufre modificación alguna durante el fraccionamiento. El esquema lógico de la Figura 4.5. muestra la partición de un sistema de doble canal en sendos sistemas de canal individual.

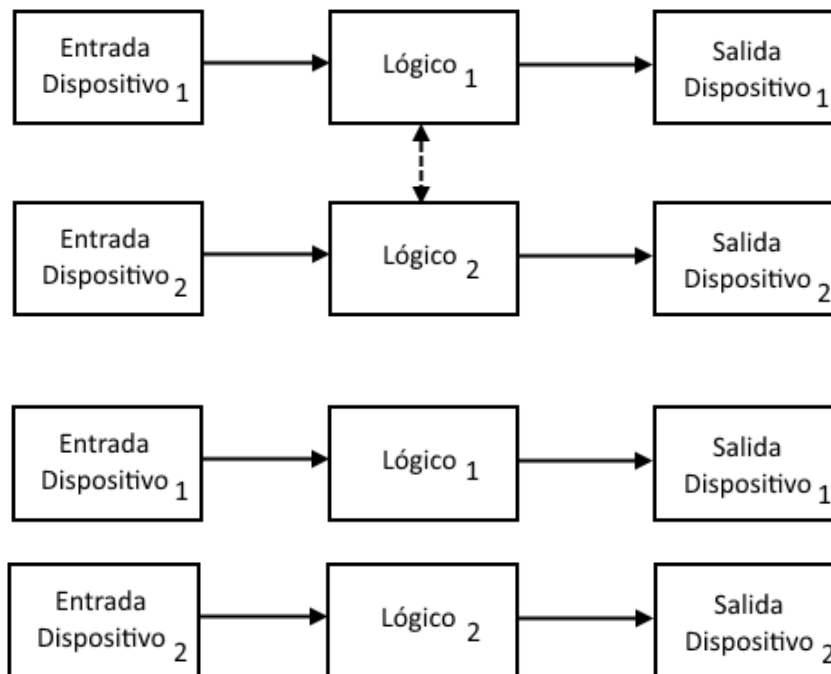


Figura 4.5. Esquema lógico de un sistema correspondiente a la Situación 3.

Debe tenerse en cuenta que, aunque en la práctica se tienen ahora dos sistemas de canal individual, en la teoría sigue siendo un único sistema de doble canal y, como tal, deberá tenerse en consideración durante la selección final de los componentes del mismo, debiendo reunir éstos las características adecuadas para la comunicación y monitoreo entre elementos.

Situación 4. Corresponde a una arquitectura que resulta de la combinación de las Situaciones 2 y 3, esto es, en la que los componentes se disponen en dos canales conteniendo, cada uno, al menos un dispositivo de entrada, una unidad lógica y un dispositivo de salida, y donde además existen dos o más componentes ejecutando una misma función de entrada, lógica o de salida en un canal o en los dos, por separado o de manera conjunta. La simplificación del sistema se lleva a cabo de acuerdo a las indicaciones presentadas en los casos de las Situaciones 2 y 3. El esquema lógico de la Figura 4.6. muestra la simplificación en dos canales individuales de un sistema de doble canal donde en el primero de ellos existen dos dispositivos en serie ejecutando una misma función de entrada.

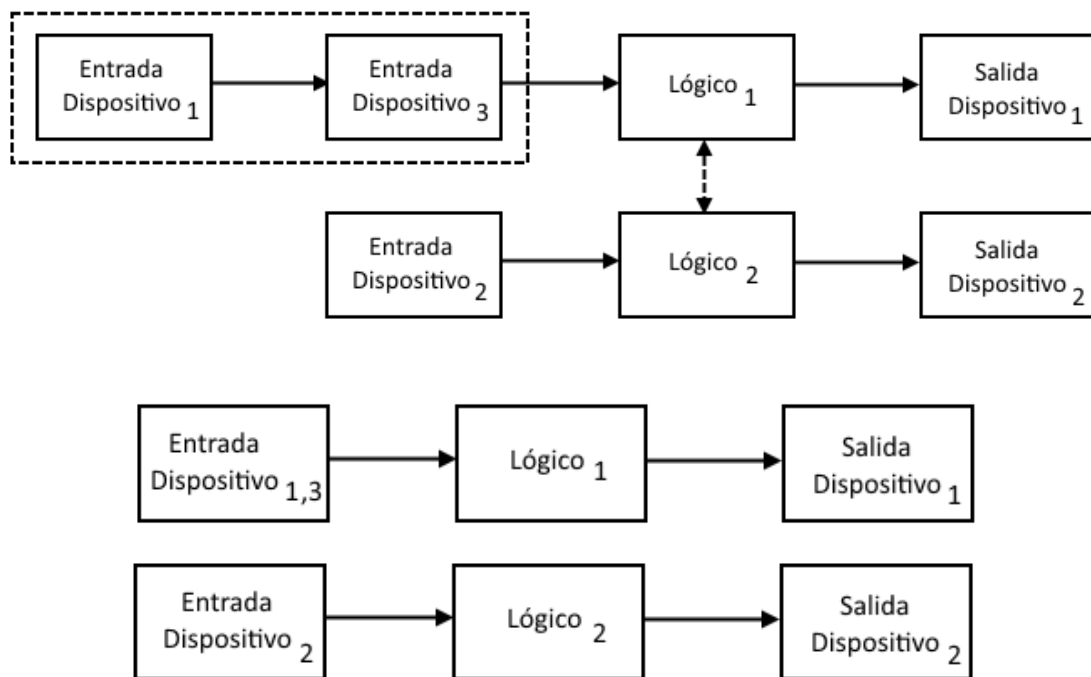


Figura 4.6. Esquema lógico de un sistema correspondiente a la Situación 4, con dos dispositivos de entrada en serie en uno de sus canales.

Consideración 2: cobertura de diagnóstico.

Como comentado en la Sección 2.6., se distinguen cuatro clases de cobertura de diagnóstico en función de la capacidad de un sistema o subsistema de detectar un fallo peligroso: nula, baja, media y alta. Existe una relación notable entre la arquitectura de

un sistema o subsistema y su cobertura de diagnóstico. En efecto, sistemas con un solo canal no suelen estar capacitados para la localización de fallos en el conjunto, este hecho comporta un riesgo a la correcta ejecución de las funciones de seguridad que implementan. Por el contrario, sistemas de doble canal permiten detectar los fallos que puedan ocurrir durante el funcionamiento del circuito de componentes. Sin embargo, presentan un límite en cuanto a la acumulación de fallos que son capaces de admitir; cuanto mayor es la categoría del sistema, mayor es el límite que condiciona el funcionamiento normal de la función de seguridad y, por ende, mayor es su cobertura de diagnóstico, es decir, mayor es la detección de fallos que pueden derivar en una pérdida de la función de seguridad con respecto del total localizado.

La relación entre cobertura de diagnóstico y arquitectura se pone de manifiesto en el gráfico de la Figura 3.4. tomado como elemento base en este apartado. En determinados casos, una categoría tiene asociada una sola cobertura de diagnóstico; esto afecta en particular a las categorías B, 1 y 4. No obstante, para las restantes categorías, 2 y 3, se tienen dos coberturas de diagnóstico asociadas: baja y media. Cuando se trabaje con una de estas dos últimas categorías y no se disponga de la información necesaria que permita asignar claramente una cobertura de diagnóstico concreta, se optará por una cobertura de clase baja. Ésta es una solución arbitraria que, sin embargo, representa una opción intermedia que permite la detección de un amplio rango de fallos (entre el 60% y el 90% del total medido) y que explota al máximo las capacidades de la estructura afectada.

Consideración 3: tiempo medio hasta fallo peligroso.

Como comentado en la Sección 2.5., se definen tres rangos de tiempo medio hasta fallo peligroso de acuerdo a la duración inscrita en ellos: bajo, medio y alto. Para cada combinación de categoría, diagnóstico de cobertura y nivel de fiabilidad existe un intervalo de tiempo medio hasta fallo peligroso admisible, que puede llegar a incluir hasta dos de los tres niveles citados.

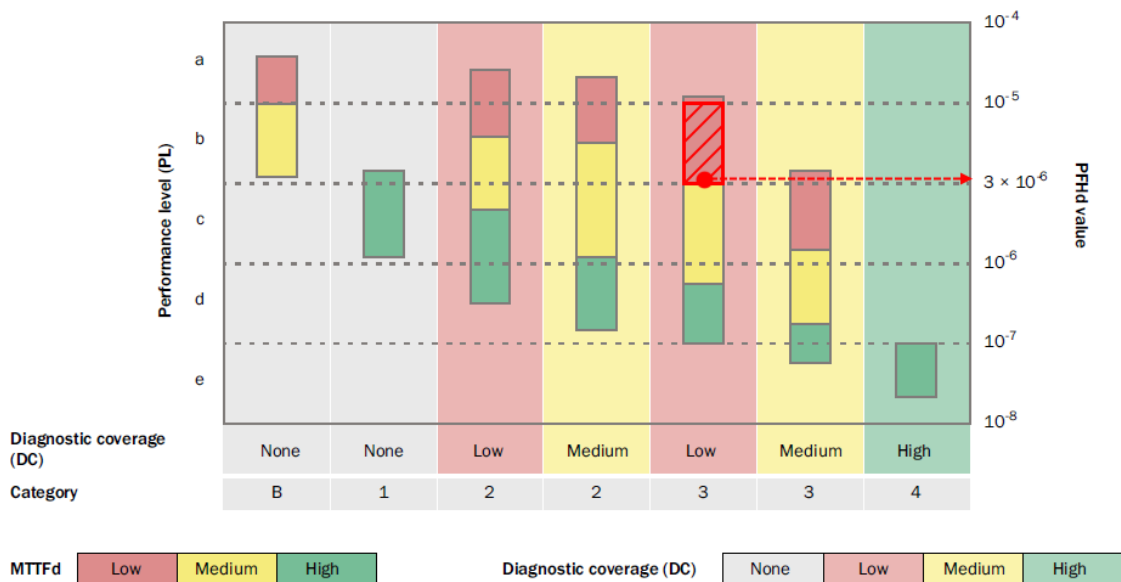
Son dos los puntos a comentar y delimitar a este respecto. En primer lugar, y como se ha explicado en la consideración anterior, cuando expuesto a la elección entre dos niveles de tiempo medio hasta fallo peligroso, esto es, bien entre un rango bajo y uno medio o bien entre un rango medio y uno alto, y se carezca de la información necesaria que permita tomar una decisión concreta, se optará por un nivel medio de tiempo medio hasta fallo peligroso. Nuevamente, esta opción constituye una posición intermedia y equilibrada entre los dos extremos posibles, evitando posicionarse en el caso más desfavorable, así como en el más óptimo o ideal. Por otro lado, cabe recordar que un nivel de fiabilidad tiene asociado un rango de tiempo medio hasta fallo peligroso y no un único valor discreto. Por este motivo, una vez seleccionado el intervalo de tiempo medio hasta fallo peligroso pertinente, se escogerá aquel valor mínimo de tiempo contenido en el mismo y perteneciente al nivel de prestaciones estudiado.

Partiendo del nivel de fiabilidad y del tipo arquitectura de un sistema o subsistema cualquiera, y atendiendo a las consideraciones anteriores, se desemboca en la determinación de su respectiva probabilidad de fallo peligroso por hora.

Ejemplo 4.1. (PFH_a global de un sistema)

Una función de seguridad es ejecutada a partir de una serie de dispositivos dispuestos según una arquitectura de doble canal de categoría 3. El nivel de fiabilidad requerido por el sistema es un PL c.

Con el fin de evaluar la probabilidad de fallo peligroso por hora del sistema se emplea el diagrama de la Figura 3.4. que se ilustra nuevamente a continuación. Para su uso, se precisa de la especificación de dos parámetros adicionales: la cobertura de diagnóstico y el tiempo medio hasta fallo peligroso.



Para un sistema de categoría 3 dos coberturas de diagnóstico son posibles: baja y media. Siguiendo las indicaciones expuestas anteriormente, dicha cobertura de diagnóstico se acota a una clase baja. Esto permite delimitar el tiempo medio hasta fallo peligroso hasta el rango representado por el área rallada de la figura. Como se observa, dicho intervalo está compuesto por sendos niveles de tiempo bajo y medio. Por consiguiente, el tiempo medio hasta fallo peligroso se limita a una clase media y el valor puntual del mismo que se escoge será el mínimo circunscrito en él, marcado por el punto rojo. Leyendo la escala en el lado derecho del diagrama se extrae finalmente la probabilidad de fallo peligroso resultante del sistema. Ésta corresponde a un PFH_a de $3,21 \cdot 10^{-6}$.

La Tabla 4.1. recoge el conjunto de probabilidades de fallo peligroso por hora consideradas para cada combinación de nivel de fiabilidad, categoría y diagnóstico de cobertura. Para ello se han tenido en cuenta las limitaciones impuestas a los rangos de tiempo medio hasta fallo peligroso y a la cobertura de diagnóstico del sistema o subsistema, expuestas a lo largo del presente apartado.

Cat. DC		PFH _d				
		B / ninguna	1 / ninguna	2 / baja	3 / baja	4 / alto
PL	a	3,80E-05	-	2,58E-05	1,26E-05	-
	b	1,14E-05	3,80E-06	7,18E-06	3,21E-06	-
	c	-	3,00E-6	3,00E-06	3,00E-06	-
	d	-	-	1,00E-06	1,00E-06	-
	e	-	-	-	-	9,54E-08

Tabla 4.1. Probabilidades de fallo peligroso por hora consideradas.

El resultado del estudio hecho hasta ahora permite la obtención de un sistema de canal individual con un solo bloque de entrada, uno lógico y uno de salida caracterizado por una determinada probabilidad de fallo peligroso por hora. Las herramientas de las que se disponen posibilitan el cálculo de las probabilidades de fallo peligroso por hora particulares de cada uno de estos bloques o dispositivos. La expresión (2.7) presentada en el Capítulo 2 exprime la probabilidad de fallo peligroso por hora de un sistema o subsistema como la suma de las probabilidades de sus partes relativas a la seguridad.

La repartición de la probabilidad de fallo peligroso por hora de un sistema de control entre sus partes relativas a la seguridad puede realizarse en base al peso o a la influencia sobre la seguridad de cada una de éstas. Esta relación puede expresarse según la expresión (4.1), donde x , y y z representan las contribuciones del dispositivo de entrada, de la unidad lógica y del dispositivo de salida, respectivamente, tal que su suma equivale a la unidad.

$$PFH_d(total) = (x + y + z) \cdot PFH_d(total) \quad \forall x, y, z \setminus x + y + z = 1 \quad (4.1)$$

Donde:

$$\begin{aligned} PFH_d(entrada) &= x \cdot PFH_d(total) \\ PFH_d(lógico) &= y \cdot PFH_d(total) \\ PFH_d(salida) &= z \cdot PFH_d(total) \end{aligned} \quad (4.2)$$

Atendiendo a la equivalencia existente entre el nivel de fiabilidad y la probabilidad de fallo peligroso por hora, estos nuevos valores de probabilidad de cada

parte relativa a la seguridad del sistema de control pueden volverse a traducir en sendos niveles de prestación y expresarse con su respectiva escala alfabética.

Ejemplo 4.2. (PFH_a de un SRP/CS)

Se parte del sistema presentado en el Ejemplo 4.1. y del valor de probabilidad hasta fallo peligroso extraído de su análisis. Éste último corresponde a $3,21 \cdot 10^{-6}$. Dadas las características del sistema (no especificadas en el enunciado), se desea que el dispositivo de entrada represente el 30% de la probabilidad hasta fallo peligroso del sistema, la unidad lógica el 45% y el dispositivo de salida el 25%.

La distribución de la contribución a la probabilidad hasta fallo peligroso del sistema se traduce en unos parámetros x , y , z equivalentes a 0,30; 0,45 y 0,25 respectivamente. Por consiguiente, se calculan las siguientes probabilidades:

$$PFH_a(\text{entrada}) = 0,30 \cdot 3,21 \cdot 10^{-6} = 9,63 \cdot 10^{-7} \quad PL\ c$$

$$PFH_a(\text{lógico}) = 0,45 \cdot 3,21 \cdot 10^{-6} = 1,44 \cdot 10^{-6} \quad PL\ c$$

$$PFH_a(\text{salida}) = 0,25 \cdot 3,21 \cdot 10^{-6} = 8,03 \cdot 10^{-7} \quad PL\ d$$

Evidentemente, puede comprobarse que la suma de las tres probabilidades evaluadas se identifica con la probabilidad del conjunto del sistema.

$$PFH_a(\text{total}) = 9,63 \cdot 10^{-7} + 1,44 \cdot 10^{-6} + 8,03 \cdot 10^{-7} = 3,21 \cdot 10^{-6}$$

4.2.5. Determinación del MTTF_a del SRP/CS.

El tiempo medio hasta fallo peligroso de un sistema de control o de cualquiera de sus partes relativas a la seguridad puede determinarse a partir de sus respectivas probabilidades de fallo peligroso por hora. En el gráfico de la Figura 3.4. se observa que para cada combinación de categoría y cobertura de diagnóstico se establecen intervalos de tiempo medio hasta fallo peligroso acotados para determinados niveles de fiabilidad. Dado que existe una equivalencia demostrada entre el nivel de prestación de un sistema y su probabilidad de fallo peligroso por hora, puede deducirse entonces una relación entre este último parámetro y el tiempo medio hasta fallo peligroso por hora. Dicha relación es reflejada en la forma de una curva de valores de un parámetro expresados en función del otro.

En la Figura 4.7. se representa el conjunto de curvas mencionadas para cada combinación de categoría y cobertura de diagnóstico. En la Sección A2 del Anexo A se presenta la tabla de valores de tiempo medio hasta fallo peligroso y de probabilidad de

fallo peligroso por hora que se ha empleado como base para el trazo de las gráficas. Si bien las relaciones ilustradas pueden expresarse usando construcciones diferentes (exponencial, logarítmica, polinómica, etc.), la forma que asegura un coeficiente de determinación R^2 más próximo a la unidad corresponde a una expresión potencial de la forma $MTTF_d = a \cdot (PFH_d)^b$.

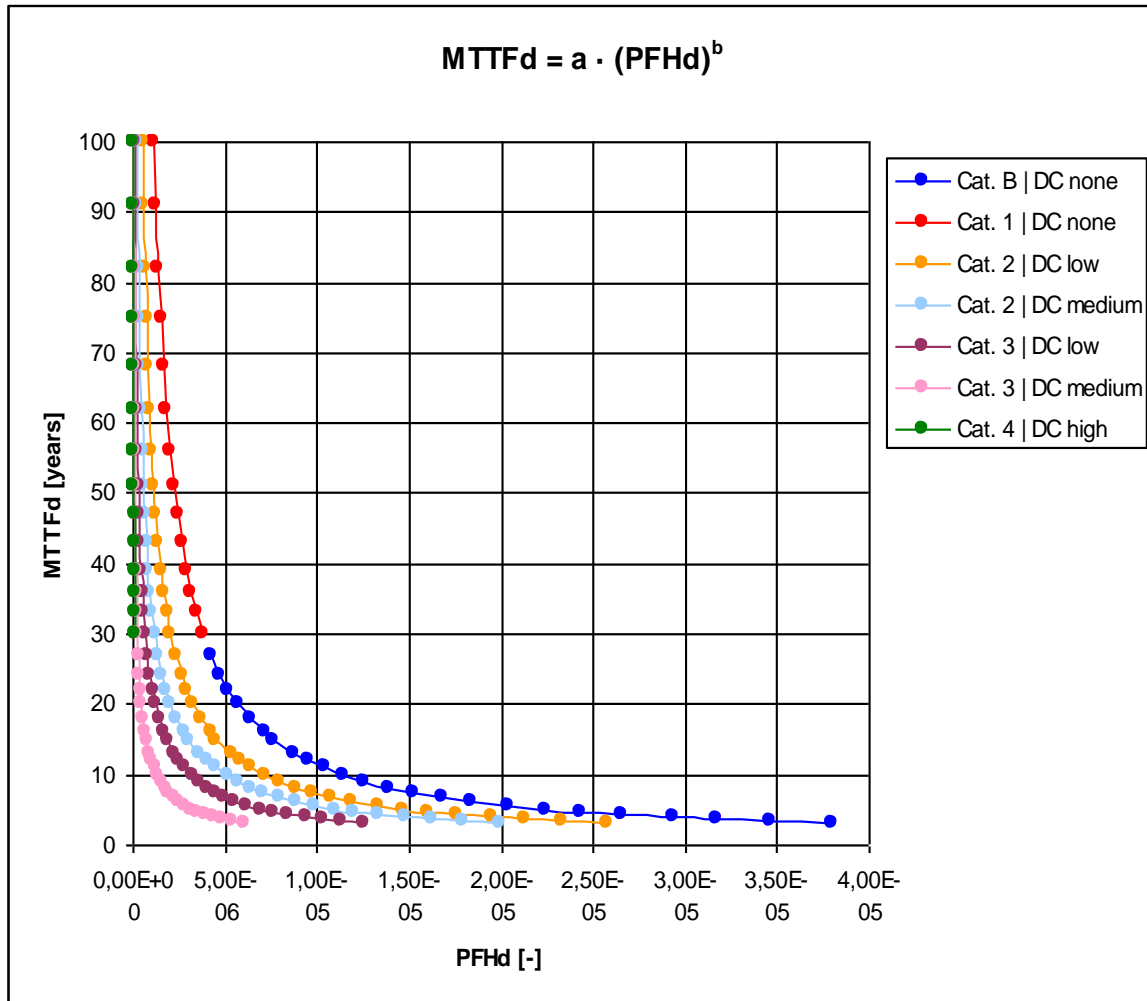


Figura 4.7. Representación del tiempo medio hasta fallo peligroso en función de la probabilidad de fallo peligroso, para una categoría y una cobertura de diagnóstico dadas.

A modo de resumen, se muestran en la Tabla 4.2. las ecuaciones representadas en la figura anterior, así como sus respectivos coeficientes de determinación y la categoría, cobertura de diagnóstico y niveles de fiabilidad a los que hacen referencia.

Cat.	DC	Ecuación	R^2	PL
B	ninguna	$MTTF_d = 0,0001 \cdot (PFH_d)^{-1,0002}$	1	a, b

1	ninguna	$MTTF_d = 0,0001 \cdot (PFH_d)^{-0,9985}$	1	b, c
2	baja	$MTTF_d = 0,0002 \cdot (PFH_d)^{-0,8975}$	0,9998	a, b, c, d
2	media	$MTTF_d = 0,0007 \cdot (PFH_d)^{-0,7796}$	0,9984	a, b, c, d
3	baja	$MTTF_d = 0,001 \cdot (PFH_d)^{-0,7191}$	0,9952	a, b, c, d
3	media	$MTTF_d = 0,0007 \cdot (PFH_d)^{-0,6998}$	0,9985	b, c, d, e
4	alta	$MTTF_d = 0,00002 \cdot (PFH_d)^{-0,89}$	1	e

Tabla 4.2. Relaciones entre el tiempo medio hasta fallo peligroso y la probabilidad de fallo peligroso según la categoría y la cobertura de diagnóstico.

Ejemplo 4.3. (MTTF_d de un SRP/CS)

Se parte del sistema presentado en el Ejemplo 4.1. y de los valores de probabilidad de fallo peligroso por hora de sus partes relativas a la seguridad extraídos de su análisis.

Conocidas la categoría del sistema (cat. 3) y la cobertura de diagnóstico del mismo (DC baja), la expresión que permite el cálculo del tiempo medio hasta fallo peligroso de sus partes relativas a la seguridad es:

$$MTTF_d = 0,001 \cdot (PFH_d)^{-0,7191}$$

De este modo se determina que:

$$MTTF_d(\text{entrada}) = 0,001 \cdot (9,63 \cdot 10^{-7})^{-0,7191} = 21,20 \text{ años}$$

$$MTTF_d(\text{lógico}) = 0,001 \cdot (1,44 \cdot 10^{-6})^{-0,7191} = 15,88 \text{ años}$$

$$MTTF_d(\text{salida}) = 0,001 \cdot (8,03 \cdot 10^{-7})^{-0,7191} = 24,16 \text{ años}$$

Estos valores son coherentes ya que, a mayor probabilidad de que un fallo peligroso tenga lugar en una hora, menor es el tiempo medio entre dos fallos peligrosos consecutivos.

4.2.6. Evaluación de las características de los SRP/CS.

Las características básicas de seguridad de los dispositivos hidráulicos, eléctricos o electrónicos se encuentran claramente delimitadas a partir de los cálculos detallados en los apartados anteriores. Sin embargo, los componentes mecánicos, electromecánicos o neumáticos, sujetos a desgaste dada la naturaleza mecánica de su

actividad, son caracterizados por una serie de parámetros adicionales que se expresan en base al número de los ciclos de operación del elemento y que deberán tenerse igualmente en consideración durante el proceso de selección de los dispositivos; éstos son: el número medio de ciclos hasta fallo peligroso del 10% de los componentes y el tiempo medio hasta el fallo peligroso del 10% de los componentes. Las expresiones (2.3) y (2.5) introducidas en el Capítulo 2 proporcionan las herramientas necesarias para la determinación de dichos parámetros, los cuales completarán la descripción analítica de los requerimientos mínimos a asegurar.

Ejemplo 4.4. (B_{10d} y T_{10d})

Se tiene un dispositivo electromecánico definido por un tiempo medio hasta fallo peligroso de 7386 años y una frecuencia de ciclos de operación estimada de 3520 ciclos por año.

El número de ciclos de operación hasta el fallo peligroso del 10% de los componentes se determina mediante el producto de los dos parámetros de partida y la aplicación del porcentaje correspondiente.

$$B_{10d} = 0,1 \cdot MTTF_d \cdot n_{op} = 0,1 \cdot 7386 \cdot 3520 = 2600000 \text{ ciclos}$$

A partir del resultado anterior puede entonces evaluarse directamente el tiempo medio hasta el fallo peligroso del 10% de los componentes a través de la utilización de la fórmula que lo define.

$$T_{10d} = \frac{B_{10d}}{n_{op}} = \frac{2600000}{3520} = 738,6 \text{ años}$$

4.2.7. Selección del componente de seguridad.

A partir del conjunto de parámetros evaluados en los apartados anteriores se seleccionan los dispositivos de seguridad pertinentes. Para ello, se hace uso de los catálogos de productos proporcionados por cualquier fabricante o distribuidor de componentes industriales de seguridad, en los que pueden encontrarse las características generales y técnicas de éstos, así como los aspectos relativos a la seguridad que atañen al análisis efectuado. La Figura 4.8. muestra la presentación de la información general de un modelo determinado de cortina de seguridad, extraída del catálogo de su fabricante.

General data

System part	Sender	Receiver
Resolution (depending on type)	14 mm / 30 mm	
Scanning range (depending on type)	-	0 m ... 8 m / 0 m ... 19 m
Protective field height (depending on type)	150 mm ... 1800 mm	
Safety related parameters	Type	Type 4 (IEC 61496)
	Safety integrity level	SIL3 (IEC 61508) SILCL3 (IEC 62061)
	Category	Category 4 (EN ISO 13849)
	Performance level	PL e (EN ISO 13849)
	PFHd (mean probability of a dangerous failure per hour)	1.5×10^{-8} (EN ISO 13849)
	T_M (Mission Time)	20 years (EN ISO 13849)
Response time (depending on type)	-	Max. 26 ms ¹⁾
Protection class	III	
Enclosure rating	IP 65	
Ambient operating temperature from ... to	0 °C ... +55 °C	
Storage temperature from ... to	-25 °C ... +70 °C	
Air humidity from ... to	15 % ... 95 %, non-condensing	
Housing cross-section	48 mm x 40 mm	
Vibration resistance	5 g (10 Hz ... 55 Hz), IEC 60068-2-6	
Shock resistance	10 g, 16 ms (IEC 60068-2-29)	

¹⁾ Without beam coding, without blanking. Other response times, see operating instructions.

Figura 4.8. Información general del modelo de cortina de seguridad C4000 Advanced, de la marca SICK. Extracto del catálogo del fabricante.

Los valores de los parámetros de seguridad que se han determinado representan valores admisibles. Esto significa, en el caso del tiempo medio hasta fallo peligroso, del número de ciclos hasta el fallo peligroso del 10% de los componentes o del tiempo medio hasta el fallo peligroso del 10% de los componentes, que productos caracterizados por valores de dichos parámetros iguales o superiores en magnitud a los hallados de manera analítica cubrirán las necesidades básicas de seguridad del sistema. Por el contrario, en el caso de la probabilidad de fallo peligroso por hora, aquellos productos definidos por un valor igual o inferior al valor calculado permitirán alcanzar dichos requerimientos mínimos.

5

Diseño y programación

En el capítulo anterior se ha presentado una metodología basada en la normativa EN ISO 13849-1 que permite la obtención de los parámetros de seguridad característicos de los distintos componentes pertenecientes a un sistema de control. Dichos dispositivos no se hallan aislados en el sistema, sino que se encuentran relacionados entre ellos tal que la variación de uno de sus parámetros tiene un impacto directo sobre las cualidades definitorias del resto de elementos integrados en la misma estructura. Ante la problemática que supone el control simultáneo de la totalidad de estas características, el uso de herramientas informáticas como Easy Java Simulations simplifican y agilizan el proceso de diseño y normalizan las diferentes tareas desempeñadas por sus usuarios.

5.1. Introducción.

Easy Java Simulations (EjsS) es una herramienta informática libre y de código abierto diseñada para la programación de simulaciones interactivas en Java o JavaScript, sin que para ello se requieran conocimientos avanzados de programación. La utilización de EjsS en el desarrollo de simulaciones facilita el conjunto de tareas de programación existentes, tanto desde un punto de vista técnico como desde uno conceptual. Desde un punto de vista técnico, se simplifica enormemente la creación del aspecto gráfico de la simulación, un proceso para el que normalmente se precisa de un nivel avanzado de conocimiento en infografía. Asimismo, desde un punto de vista conceptual, EjsS provee de una estructura sencilla para desarrollar los modelos de simulación, esto es, la descripción física del modelo estudiado. Sin embargo, esto no supone que la manipulación del programa esté exenta de esfuerzo, pues parte del trabajo seguirá dependiendo en gran medida del propio usuario. Éste se hace responsable del diseño de la vista del programa, así como de proporcionar las variables y algoritmos necesarios para completar la descripción del modelo de la simulación.

La ventana de EjsS que ejerce de interfaz de usuario se presenta en la imagen izquierda de la Figura 5.1. Pueden destacarse hasta cinco elementos clave. La zona inferior de la ventana está constituida por un área donde se muestran mensajes de aviso y de información emitidos por parte del programa. Por otro lado, una barra de herramientas en la región derecha de esta ventana permite realizar tareas básicas, tales como limpiar, abrir, guardar, buscar o comprimir ficheros, además de ejecutar las simulaciones abiertas en el programa. Inmediatamente arriba de esta barra de herramientas, un panel de información permite especificar determinados datos acerca del fichero (nombre, autor, idioma o comentarios, entre otros).

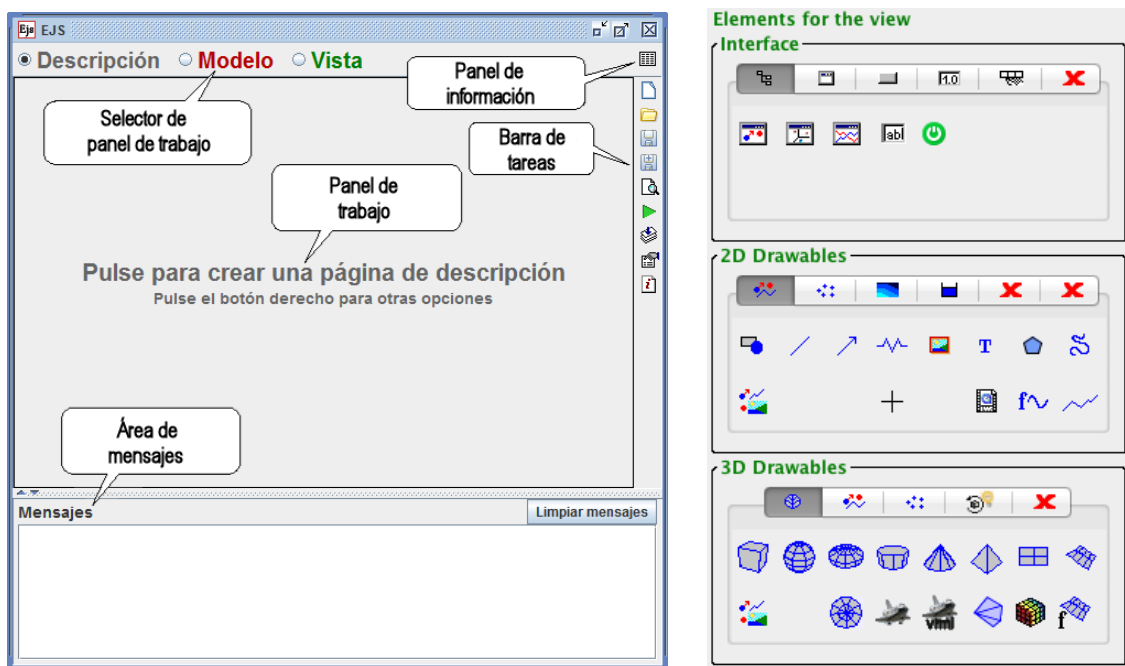


Figura 5.1. Interfaz de usuario de Easy Java Simulations.
Izda.: panel de “descripción”. Dcha.: elementos de la vista.

La barra de herramientas ubicada en la parte superior de la ventana habilita tres paneles de trabajo distintos para construir la simulación. Estos paneles se visualizan en la región central de la interfaz y corresponden a:

- **Descripción.** Panel que permite la creación y edición de textos en código HTML para la descripción de la simulación, la cual aparecerá en una ventana emergente al ejecutar la aplicación o como parte de la página web donde se ubica la simulación, al ejecutarla como applet.
- **Modelo.** Panel dedicado a la creación de la simulación, conteniendo la declaración e inicialización de las variables, así como la escritura de los algoritmos que describen el comportamiento de dichas variables con el tiempo o

según las relaciones impuestas por el propio usuario. Las variables empleadas pueden ser de cinco clases diferentes: booleanas, enteras, dobles, cadenas de caracteres u objetos de Java.

- Vista. Panel que permite la creación de la interfaz gráfica de la simulación, disponiendo para ello de elementos para el control de la simulación y para la visualización de los valores de salida, los cuales se encuentran organizados en tres grupos: elementos de la interfaz (ventanas, paneles y botones), elementos de dibujo 2D (elementos para paneles bidimensionales o con ejes 2D) y elementos de dibujo 3D (elementos para paneles tridimensionales o con ejes 3D).

Con todo esto, el producto final extraído de EjsS es una aplicación Java independiente o multiplataforma, o un applet capaz de ser visualizado en cualquier navegador web o dispositivo que sustente un entorno Java, incluido Android o iOS, pudiendo asimilar y leer datos directamente a través de la red y controlar éstos por el intermedio de scripts o archivos de órdenes incluidos en las páginas web.

El capítulo se organiza de la siguiente manera. En la Sección 5.2. se presenta la interfaz gráfica del programa desarrollado mediante EjsS. En los cinco apartados que se derivan a continuación, se ahonda en cada uno de los elementos y secciones que la constituyen y se detallan los algoritmos de cálculo implementados en ellos. La Sección 5.3. expone el procedimiento de diseño seguido por el programa interactivo para la selección de dispositivos de seguridad, presentando en los apartados que siguen los pasos de manera individual.

5.2. Entorno gráfico del programa.

La interfaz de usuario principal del programa se muestra en la imagen de la Figura 5.2. En ella, se presentan cuatro secciones diferentes, a las que se añade una quinta en una ventana adicional. La parte superior de la ventana está constituida por una barra de herramientas dividida en diversos bloques según la funcionalidad de las operaciones que incluye, donde se habilita la definición de los sistemas de control estudiados y el cálculo de su nivel de fiabilidad, así como la inserción de listas de dispositivos que constituirán una base de datos para la futura selección de las partes relativas a la seguridad del sistema de control. En la parte izquierda de la ventana, un diagrama de estructura de árbol ilustra la organización de la máquina estudiada. En el lado derecho, un área de trabajo permite manipular los datos de las distintas partes relativas a la seguridad por el intermedio de una serie de gráficos. Inmediatamente debajo de esta región, un recuadro posibilita la selección del sistema que se desea estudiar, el cual se visualizará en el panel de trabajo mencionado. En la parte inferior de la ventana, un área de mensajes muestra mensajes de aviso y notificaciones emitidas por el propio programa, que deben servir de guía al usuario.

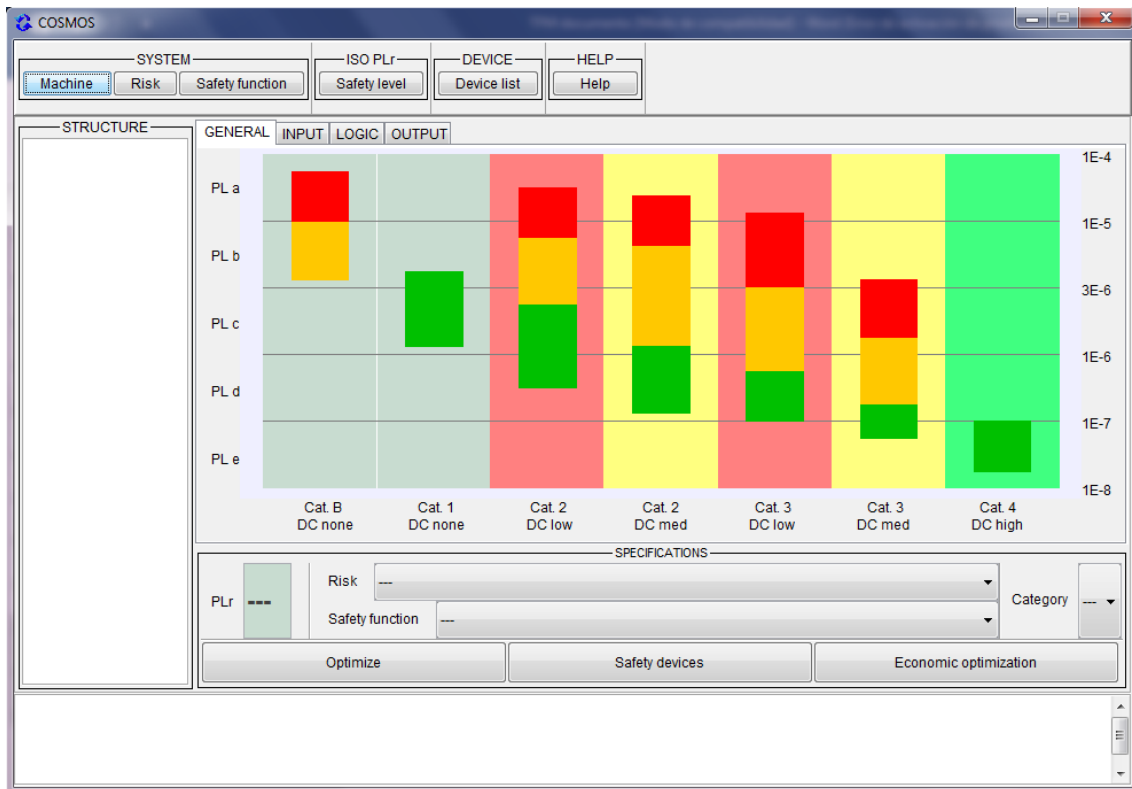


Figura 5.2. Interfaz de usuario del programa: ventana principal.

A continuación, se procede a exponer en detalle cada una de las partes del programa presentadas.

5.2.1. Barra de herramientas.

La barra de herramientas es un elemento que se ubica en la región superior de la ventana principal del programa y que ocupa la totalidad de la misma. Está compuesta por una serie de seis botones que permiten desempeñar tareas básicas, cuya funcionalidad está limitada prácticamente a las primeras etapas del estudio del sistema. Los botones se agrupan en diferentes bloques de uno o más elementos en base al propósito al que éstos sirven. Se distinguen los bloques de: (a) Definición del sistema, (b) Cálculo del nivel de fiabilidad, (c) Introducción de nuevos dispositivos y (d) Ayuda.

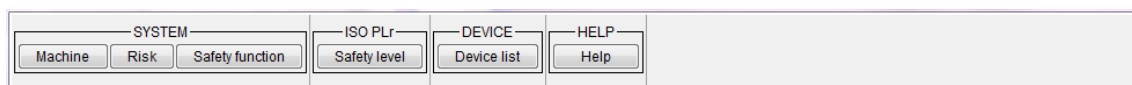


Figura 5.3. Barra de herramientas.

Definición del sistema.

El primer bloque representado en la barra de herramientas permite realizar la definición de la estructura principal del sistema estudiado. Ésta se efectúa en un proceso de tres pasos, en el que se especifican, según el orden en que se incluyen, (1) la máquina evaluada, (2) los riesgos teniendo lugar en la misma y (3) las funciones de seguridad que se derivan de los riesgos anteriores. Los botones que permiten realizar las acciones mencionadas reciben el nombre de: “Máquina”, “Riesgo” y “Función de seguridad”, respectivamente.

Máquina. Representa el elemento objeto de estudio. Una máquina, dado su funcionamiento, su manipulación o su presencia en determinados ambientes se ve expuesta a uno o varios riesgos que pueden influir en su actividad. Al presionar el botón “Máquina”, una ventana emergente (ver Figura 5.4.) aparece donde se pide especificar el nombre de la máquina en el campo disponible. La acción es validada accionando a continuación el botón “Añadir máquina” situado inmediatamente debajo del campo.

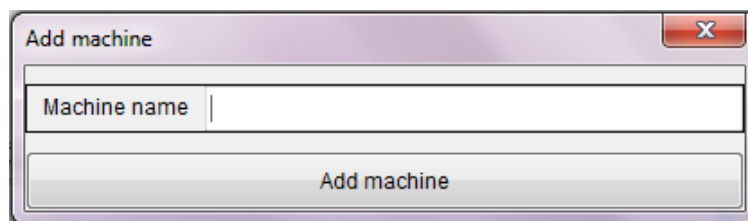


Figura 5.4. Ventana emergente: “Añadir máquina”.

La información de salida es de dos tipos. Por un lado, se tiene una cadena de caracteres representando el nombre de la máquina, el cual es visualizado en el diagrama de estructura de árbol en el lado izquierdo de la ventana principal. Por otro lado, se asigna un valor de 1 a una variable interna del programa con el nombre de *machine_v*. Esta variable, con valor inicial 0, sirve de indicación al programa y al usuario de la creación de un nuevo sistema. Cuando el valor es nulo, el programa entiende que ningún sistema está en curso de ser estudiado y, por tanto, bloquea todo intento de realizar cualquier acción que no se corresponda con identificar la máquina a evaluar. Cuando el valor es la unidad, no existe ninguna restricción inicial y podrá procederse a la identificación de los riesgos de la máquina.

Riesgo. Representa el peligro potencial al que se ve expuesta una máquina. Un riesgo está ligado a la existencia de una máquina y, del mismo modo, tiene asociada una o varias funciones de seguridad que evitan o reducen su impacto en el sistema en caso de suceso. Al presionar el botón “Riesgo”, una ventana emergente (ver Figura 5.5.) aparece

donde se pide especificar el nombre del riesgo en el campo disponible. La acción es validada accionando a continuación el botón “Añadir riesgo” situado inmediatamente debajo del campo.

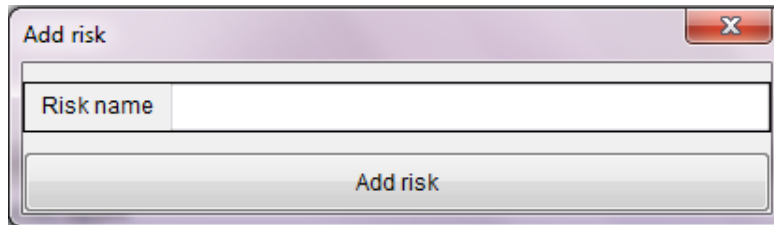


Figura 5.5. Ventana emergente: “Añadir riesgo”.

La información de salida es de dos tipos. Por un lado, se tiene una cadena de caracteres representando el nombre del riesgo, el cual es visualizado en el diagrama de estructura de árbol en el lado izquierdo de la ventana principal. Dicha cadena se almacena como un elemento de un vector columna de 100 filas; este hecho supone que el programa puede asimilar hasta un máximo de 100 riesgos diferentes. De manera análoga, una variable entera es asociada a dicho riesgo y retenida en un vector columna de idénticas dimensiones. La posición donde se ubica se corresponde con la posición que ocupa la cadena de caracteres que tiene asociada en su respectivo vector. Por consiguiente, la variable adopta un valor inicial nulo e incrementa unitariamente a medida que un nuevo riesgo es creado en el sistema, hasta alcanzar su máximo en 100.

Función de seguridad. Representa el medio a través del cual un equipo se mantiene bajo control y en un estado de seguridad ante una situación de peligro. Por tanto, una función de seguridad está relacionada inevitablemente a uno o varios riesgos. Al presionar el botón “Función de seguridad”, una ventana emergente (ver Figura 5.6.) aparece donde se pide especificar, por medio de sendas pestañas desplegadas, el riesgo previamente creado al que se asocia la función de seguridad, por un lado, y la función de seguridad deseada, en un segundo campo. El programa presenta una lista con doce funciones de seguridad diferentes, que pretenden cubrir la mayoría de los casos que pueden darse en un sistema; éstas son introducidas en el Apartado 3.2.1. del presente documento. La acción es validada accionando a continuación el botón “Añadir función de seguridad” situado en la parte inferior de la ventana emergente.

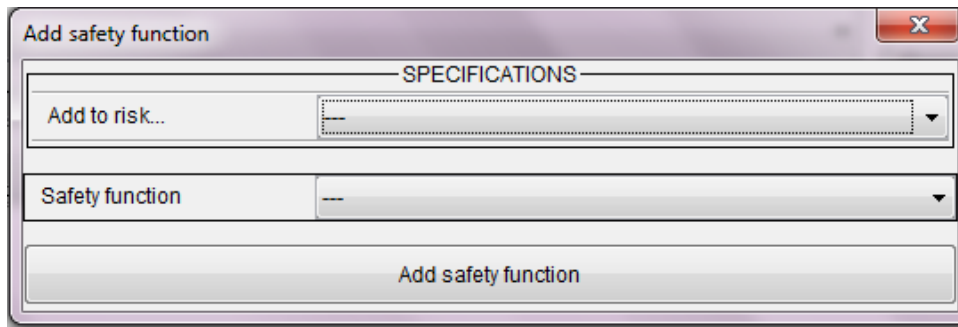


Figura 5.6. Ventana emergente: “Añadir función de seguridad”.

La información de salida es de dos tipos. Por una parte, se tiene una cadena de caracteres representando el nombre de la función de seguridad, la cual es visualizada en el diagrama de estructura de árbol en el lado izquierdo de la ventana principal. Dicha cadena se almacena como un elemento de una matriz de dimensión (100x12), donde las filas corresponden al número máximo de riesgos que pueden definirse en el programa y las columnas a las doce funciones de seguridad distintas que un riesgo puede tener asociadas. Cada función de seguridad tiene una columna específica asignada; por ejemplo, la función de seguridad “Impedir el acceso de manera permanente” está ligada a la primera columna de la matriz. De este modo, se evita definir una misma clase de función de seguridad varias veces en un mismo riesgo. Asimismo, a medida que se definen funciones de seguridad, se crea una cadena de caracteres conteniendo el conjunto de funciones de seguridad asociadas a un mismo riesgo, las cuales se encuentran separadas por un punto y coma entre ellas. Esta cadena se recoge en un vector columna de 100 filas, donde cada una de éstas corresponde con uno de los riesgos descritos. Su propósito es el de constituir una lista de funciones de seguridad pertenecientes a un mismo riesgo para futuras aplicaciones descritas en este capítulo.

Como puede observarse, el papel que cumplen los tres elementos expuestos (máquina, riesgo y función de seguridad) no es tanto el de proporcionar información acerca de cuál es el sistema con el que se desea trabajar, sino el de dotar de una cierta coherencia e imponer un orden específico al conjunto de datos que se introducirán y manipularán en el programa en sus distintas funcionalidades.

Cálculo del nivel de fiabilidad.

El segundo bloque representado en la barra de herramientas permite realizar el cálculo del nivel de prestaciones de una función de seguridad definida previamente en el programa. Éste se efectúa por medio del botón con nombre “Nivel de fiabilidad”. Al presionar el botón “Nivel de fiabilidad”, una ventana emergente (ver Figura 5.7.) aparece donde se pide especificar, por medio de sendas pestañas desplegadas, el riesgo y su correspondiente función de seguridad asociada para la que se quiere evaluar su

nivel de prestaciones. De igual modo, tres pestañas adicionales con los títulos de “Severidad (S)”, “Frecuencia (F)” y “Posibilidad (P)” permiten caracterizar el riesgo en base a su impacto, su recurrencia y su grado de evasión, respectivamente. Las opciones ofrecidas en cada situación se introducen en parejas con carácter opuesto; éstas son, según el orden de mención de los parámetros: leve/grave, raro/frecuente y evitable/inevitable. La acción es validada accionando a continuación el botón “Añadir nivel de fiabilidad” situado en la parte inferior de la ventana emergente.

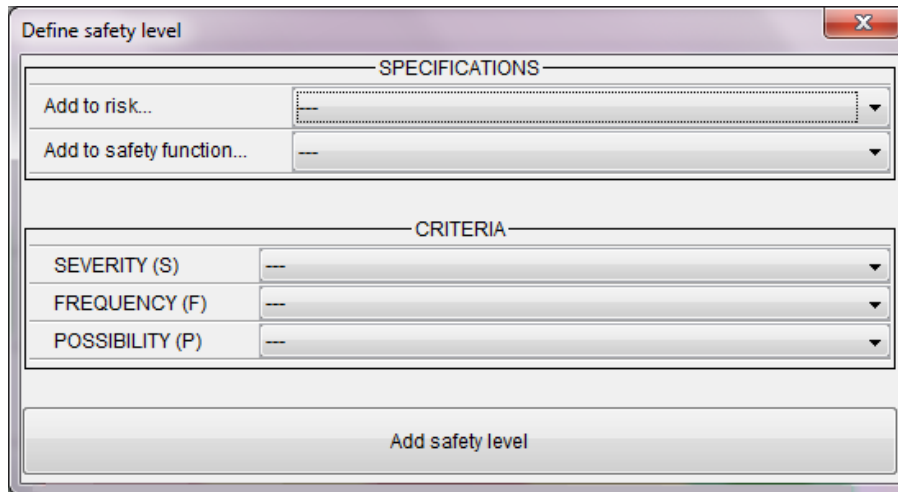


Figura 5.7. Ventana emergente: “Definir nivel de fiabilidad”.

El nivel de fiabilidad se determina como resultado de la combinación de los parámetros de severidad, frecuencia y posibilidad propios de cada riesgo. La Tabla 5.1. muestra el conjunto de ocho combinaciones posibles a partir de las opciones ofrecidas, así como el respectivo nivel de fiabilidad derivado de cada una de ellas. La información de salida consiste en una cadena de caracteres conteniendo la letra característica del nivel de prestaciones hallado. Esta variable se almacena como un elemento de una matriz de dimensiones (100x12x1), donde las filas corresponden al número máximo de riesgos que pueden definirse en el programa, las columnas a las doce funciones de seguridad distintas que un riesgo puede tener asociadas y la profundidad al nivel de fiabilidad ligado a la función de seguridad, única para cada conjunto riesgo + función de seguridad.

Severidad	Frecuencia	Posibilidad	PL
Leve	Raro	Evitable	a
Leve	Raro	Inevitable	b
Leve	Frecuente	Evitable	b
Leve	Frecuente	Inevitable	c
Grave	Raro	Evitable	c

Grave	Raro	Inevitable	d
Grave	Frecuente	Evitable	d
Grave	Frecuente	Inevitable	e

Tabla 5.1. Combinación de los parámetros severidad, frecuencia y posibilidad.

Introducción de nuevos dispositivos.

El programa elaborado permite la introducción de listas de dispositivos con sus respectivas características de seguridad y de precio, a partir de las cuales se realiza una selección de los mismos en base a las especificaciones definidas por el usuario en la parte de la interfaz detallada más adelante. Los ficheros que constituyen dichas listas de componentes de seguridad deben seguir un formato específico y su contenido cumplir con una estructura determinada.

En efecto, el archivo debe presentar en primer lugar una extensión *.xml o *.txt con tal de poder ser leído por el programa. Este archivo debe tener cabida para un total de 300 dispositivos de seguridad, de los cuales 100 de ellos son dispositivos de entrada, 100 son unidades lógicas y 100 son dispositivos de salida. Sin embargo, no es indispensable que la lista contenga la cantidad mencionada, siempre y cuando no se supere el límite de 300 elementos y se respete la proporción asignada a cada tipo de dispositivo. Para cada componente de seguridad registrado en dicha lista, las propiedades enumeradas son diez y corresponden, según el orden en que se presentan, a: (a) el nombre del dispositivo de seguridad, (b) el nombre del fabricante, (c) el nivel de fiabilidad, (d) el tipo de tecnología que representa, (e) la probabilidad de fallo peligroso por hora, (f) el tiempo medio hasta fallo peligroso, (g) el número de ciclos hasta el fallo peligroso del 10% de los componentes, (h) el número de ciclos de operación, (i) el tiempo medio hasta el fallo peligroso del 10% de los componentes y (j) su precio de venta. Las distintas características de un mismo componente de seguridad son separadas por una coma, mientras que los diferentes dispositivos se distinguen entre ellos mediante un punto y coma y un posterior salto de línea. En caso de no especificarse uno de los diez campos asignados a un componente de seguridad, se marcará bien con un “0” si es una propiedad de carácter numérico o bien con una “o” si ésta se corresponde con una cadena de caracteres. La característica que imperativamente debe definirse es la probabilidad de fallo peligroso por hora (quinto campo) o, en su defecto, el tiempo medio hasta fallo peligroso (sexto campo), ya que los cálculos internos del programa se fundamentan en dichos parámetros. El resto de variables, aunque no obligatorias, deben servir para acotar la selección del dispositivo.

Al presionar el botón “Lista de dispositivos”, una ventana emergente (ver Figura 5.8.) aparece donde se pide especificar en el campo disponible el nombre del fichero conteniendo la lista de dispositivos de seguridad. La acción es validada accionando a continuación el botón “Abrir”.

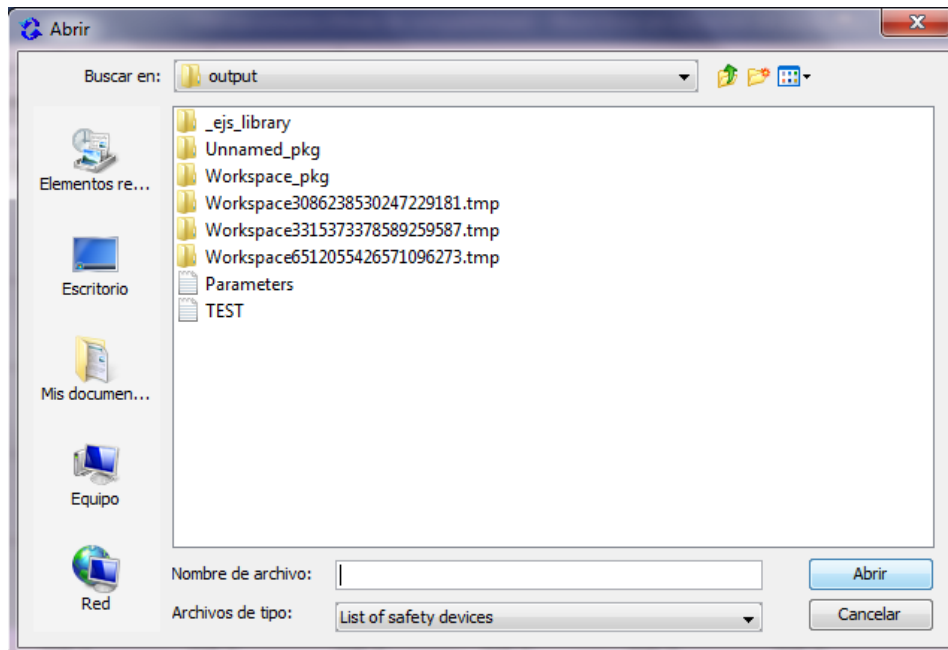


Figura 5.8. Ventana emergente: “Abrir lista de dispositivos”.

El programa lee entonces el contenido del archivo, el cual es asimilado como una única cadena de caracteres, y aplica las funciones y transformaciones necesarias para distinguir cada dispositivo y delimitar cada una de sus características, entendiéndolas ya sea como un elemento numérico o como uno alfabético, esto es, una cadena de caracteres. Estas cualidades extraídas son clasificadas en seis matrices en función del tipo de componente al que se refieren (dispositivo de entrada, unidad lógica o dispositivo de salida) y de la naturaleza del campo (numérico o alfabético). Esto supone, por ejemplo, que para el conjunto de dispositivos de entrada se tendrán dos matrices. La primera de ellas, de dimensión (100x4), contiene el nombre del componente de seguridad, su fabricante, el nivel de fiabilidad y la clase de tecnología implementada. En una segunda matriz de dimensión (100x6) se inscriben el resto de parámetros indicados en este apartado, tales como la probabilidad hasta fallo peligroso, el tiempo medio hasta fallo peligroso o el precio del componente de seguridad. Estas seis matrices resultantes son utilizadas posteriormente en el proceso de selección de los dispositivos de seguridad.

Ayuda.

La función “Ayuda” constituye el último bloque representado en la barra de herramientas y tiene como objetivo el servir de guía a todo usuario que se encuentre perdido en cualquier instante durante la utilización del programa. Al presionar el botón “Ayuda”, una ventana emergente (ver Figura 5.9.) aparece donde se recoge una serie de instrucciones para el manejo del programa en la forma de enunciados sencillos y cortos.

Dichas directrices se estructuran en tres bloques: (A) Definición del sistema, donde de exponen las pautas para concretar un sistema, partiendo del rango más general considerado por el programa hasta el nivel de fiabilidad de uno de sus riesgos; (B) Manipulación del sistema, donde de detalla brevemente cómo seleccionar y estudiar un sistema previamente definido; y (C) Visualización de los resultados, donde se listan las diferentes opciones para mostrar los productos que más se adecuan a los intereses del usuario.

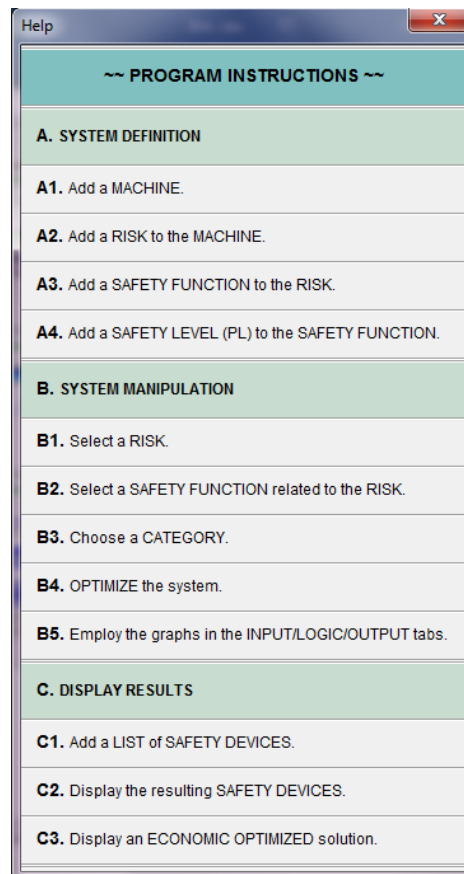


Figura 5.9. Ventana emergente: “Ayuda”.

5.2.2. Árbol de la estructura.

El árbol de la estructura es un elemento que ocupa la zona izquierda de la ventana principal del programa. En él, se muestra la organización del sistema creado, visualizando los diferentes elementos a medida que éstos son definidos en el programa. El orden en que se muestran e incluyen unos dentro de otros es: Máquina > Riesgo > Función de seguridad > PL. Como se ha indicado en los apartados anteriores, el programa solamente es capaz de asimilar una única máquina o entorno de trabajo, para la cual es posible especificar hasta 100 riesgos diferentes, cada uno de ellos suplidos por hasta 12 funciones de seguridad distintas caracterizadas por su respectivo nivel de

fiabilidad. Cada conjunto de riesgo – función de seguridad – nivel de fiabilidad constituye por sí solo un caso de estudio individual.

5.2.3. Área de trabajo.

El área de trabajo es un elemento que se ubica en la región central de la ventana principal del programa. Está compuesta por un área en su zona inferior que permite la selección del sistema a evaluar y por cuatro paneles de trabajo habilitados por sendas pestañas en su parte superior, en los que poder visualizar y manipular los parámetros característicos del sistema de control y de sus partes relativas a la seguridad. La explotación de esta sección supone el diseño previo del sistema y el subsiguiente cálculo de su nivel de fiabilidad.

Especificaciones del caso de estudio.

Como ya se ha comentado, la selección del sistema que se desea estudiar es llevada a cabo en la región inferior del área de trabajo, donde un bloque con el título de “Especificaciones” está dispuesto para tal uso. Esta tarea es desempeñada a través de la utilización de dos pestañas desplegadas donde especificar el conjunto del riesgo y su función de seguridad asociada a evaluar; las opciones mostradas en dichos campos se corresponden con los riesgos y funciones de seguridad previamente creados, tal que al abrir el programa ninguna opción se encuentra disponible. Al seleccionar un riesgo en el primer campo, automáticamente se mostrará en el segundo la totalidad de funciones de seguridad asociadas a éste; sin embargo, la acción inversa no es realizable. Con esto, el panel izquierdo del bloque muestra el nivel de fiabilidad requerido para el riesgo y función de seguridad escogidos, igualmente calculado con anterioridad a través del botón “Nivel de fiabilidad”.



Figura 5.10. Definición de las especificaciones del caso de estudio.

Una tercera pestaña en el lado derecho del bloque permite definir la categoría del sistema de control en base al nivel de prestaciones del sistema, para la selección de la cual es necesaria la elección previa del riesgo y su función de seguridad, sin la cual no

es posible mostrar las opciones del selector. La Tabla 5.2. muestra el conjunto de opciones de categoría disponibles para cada nivel de fiabilidad del sistema.

PL _r	Categoría
a	B, 2, 3
b	B, 2, 3
c	1, 2, 3
d	2, 3
e	4

Tabla 5.2. Opciones de categoría disponibles según el PL_r del sistema.

En la parte inferior del bloque se presentan tres botones cuya funcionalidad es la de lanzar el análisis del sistema y permitir la posterior selección de los dispositivos de seguridad.

Optimizar. Representa la orden de cálculo de las características individuales de las partes relativas a la seguridad de un sistema de control. Al presionar el botón “Optimizar”, se ejecuta una serie de operaciones internas que desembocan en el cálculo de la probabilidad de fallo peligroso por hora de cada una de las tres partes relativas a la seguridad consideradas: dispositivo de entrada, unidad lógica y dispositivo de salida, estudiadas en la segunda, tercera y cuarta pestaña del área de trabajo, respectivamente. El procedimiento de cálculo ya ha sido detallado en el Capítulo 5.

El parámetro de entrada corresponde al nivel de prestaciones del sistema, a partir del cual se extrae su respectiva probabilidad de fallo peligroso por hora y las correspondientes probabilidades individuales de los componentes que integra. Dado que se carece de información adicional acerca de la contribución a la seguridad de cada uno de estos componentes, el programa realiza una repartición equitativa inicial de la probabilidad de fallo peligroso por hora del sistema. En consecuencia, tomado como punto de partida la expresión (4.1), se determina que:

$$PFH_a(entrada) = PFH_a(lógico) = PFH_a(salida) = \frac{1}{3} \cdot PFH_a(total) \quad (5.1)$$

De estos parámetros se halla entonces el tiempo medio hasta fallo peligroso, el nivel de prestaciones, así como las características particulares de los elementos mecánicos (cuando aplicable) para cada parte relativa a la seguridad.

Dispositivo de seguridad. Representa la visualización del conjunto de elementos de seguridad cumpliendo con los requisitos de seguridad impuestos en la etapa de diseño. Al presionar el botón “Dispositivo de seguridad”, una ventana emergente aparece ofreciendo el listado de dichos componentes. La estructura, propiedades y utilización de la ventana emergente se detallarán en un apartado separado de este capítulo.

Optimización económica. Representa la selección automática de los dispositivos de seguridad a partir de su valor de probabilidad de fallo peligroso por hora y su precio de venta. Al presionar el botón “Optimización económica”, el programa accede a la base de datos de dispositivos de seguridad y escoge el conjunto de tres elementos (constituido por un componente de entrada, una unidad lógica y un componente de salida) que cumpla con la probabilidad de fallo peligroso por hora del sistema o su nivel de prestaciones mínimo requerido, al mismo tiempo que constituya la opción más barata de todas aquéllas viables. Ello no supone que la solución determinada sea la aconsejada para el sistema estudiado. La tarea del usuario es estudiar la propuesta y sopesar su implementación.

Pestaña 1: gráfico de las características del sistema.

La primera pestaña presentada permite visualizar y contextualizar el estudio del sistema a través de un gráfico (ver Figura 5.11.) basado en el diagrama de la Figura 3.4., el cual recoge la categoría, la cobertura de diagnóstico, el nivel de fiabilidad, la probabilidad de fallo peligroso por hora y el tiempo medio hasta fallo peligroso del sistema de control seleccionado. El programa toma como variables de entrada el nivel de prestaciones del sistema y su categoría, delimitada con anterioridad.

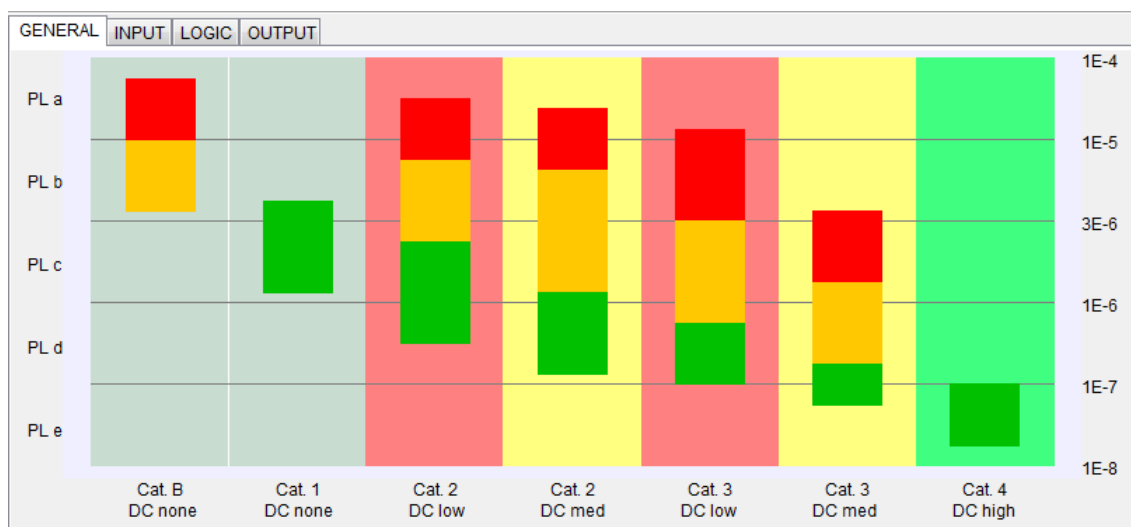


Figura 5.11. Pestaña 1: gráfico de las características del sistema.

Al presionar el botón “Optimizar”, se traza un punto negro en aquel lugar del diagrama donde coinciden los parámetros mencionados. El usuario puede entonces estimar a simple vista los parámetros característicos restantes a partir de la lectura del gráfico. Para más información acerca de cómo interpretar el diagrama, remitirse al apartado “Estimación del nivel de fiabilidad obtenido” en el Capítulo 3.

Pestañas 2, 3 y 4: estudio particular de las SRP/CS.

La segunda, tercera y cuarta pestaña constituyen los paneles de trabajo particulares del dispositivo de entrada, de la unidad lógica y del dispositivo de salida del sistema de control, respectivamente. El entorno de trabajo es idéntico para las tres pestañas expuestas y se segmenta en dos secciones claramente diferenciadas. En el lado izquierdo del panel un bloque proporciona el conjunto de parámetros que definen la parte relativa a la seguridad estudiada en la pestaña. El espacio restante de la pestaña es ocupado por dos gráficos describiendo sendas relaciones entre la probabilidad de fallo peligroso del 10% de los componentes y el número de ciclos de operación de los mismos, por un lado, y entre el tiempo medio hasta fallo peligroso y la probabilidad de fallo peligroso por hora, por el otro.

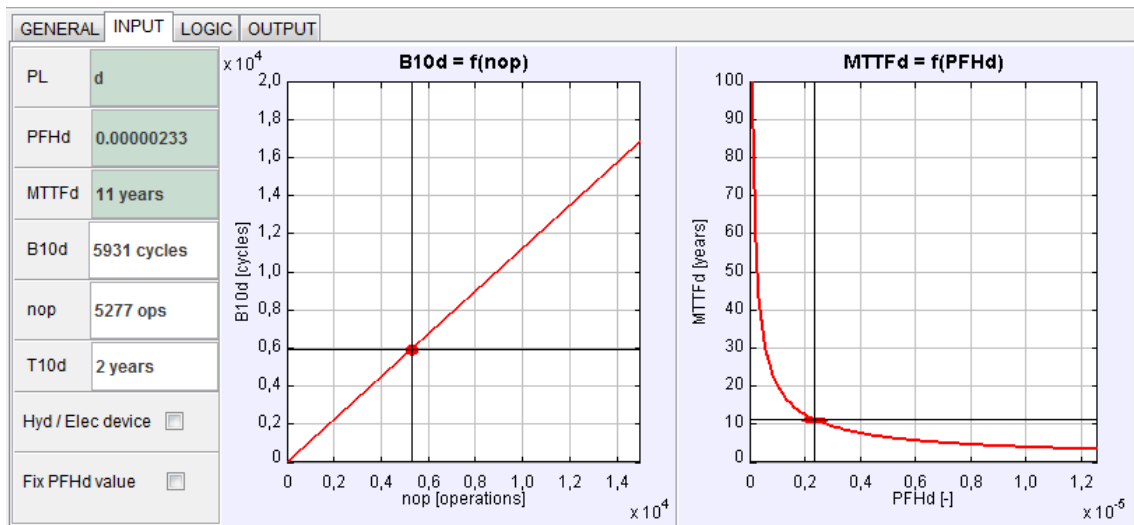


Figura 5.12. Panel de trabajo de una SRP/CS (dispositivo de entrada).

Parámetros del SRP/CS. El conjunto de variables que define a una parte relativa a la seguridad de un sistema de control es recogido y visualizado en el bloque mostrado a la izquierda de cada una de las pestañas concernientes al componente de seguridad. Los parámetros contemplados son: (a) el nivel de fiabilidad, (b) la probabilidad de fallo peligroso por hora, (c) el tiempo medio hasta fallo peligroso por hora, (d) el número de ciclos hasta el fallo peligroso del 10% de los componentes, (e) el número de ciclos de

operación por año del dispositivo y (f) el tiempo medio hasta el fallo peligroso del 10% de los mismos.

PL	d
PFHd	0.00000233
MTTFd	11 years
B10d	5931 cycles
nop	5277 ops
T10d	2 years
Hyd / Elec device	<input type="checkbox"/>
Fix PFHd value	<input type="checkbox"/>

Figura 5.13. Parámetros del SRP/CS.

En adición, se presentan dos casillas en la zona inferior del bloque. La primera de ellas sirve para especificar si el elemento evaluado corresponde con un dispositivo de naturaleza contraria a una mecánica, ya que, en este caso, no todos los parámetros indicados son aplicables (esto afecta en particular a las tres últimas variables listadas). Por consiguiente, al marcar la casilla, todos los campos concernientes a dispositivos de naturaleza hidráulica, eléctrica o electrónica son desactivados, así como el gráfico situado inmediatamente al lado del bloque de parámetros, al estar éste vinculado a los dispositivos mencionados. La segunda casilla permite fijar el valor de probabilidad de fallo peligroso por hora del segundo gráfico del panel, de manera que, al manipular el mismo gráfico en las pestañas de los otros dos componentes de seguridad, el primer gráfico no experimenta variación alguna y se mantendrá en el estado de su valor de probabilidad fijado.

Gráfico 1: relación entre B_{10d} y n_{op} . Representa la relación entre el número de ciclos hasta el fallo peligroso del 10% de los componentes con respecto del número de ciclos de operación por año de los mismos. Dicha relación se fundamenta en la fórmula (2.3) presentada en el Capítulo 2, de donde se deduce una expresión lineal de la forma:

$$B_{10d} = 0,1 \cdot MTTF_d \cdot n_{op} \quad (5.2)$$

Esta expresión utiliza en primera instancia el tiempo medio hasta fallo peligroso evaluado a través de los cálculos internos del programa tras presionar el botón “Optimizar” y más adelante el valor correspondiente de la variable al manipularla en el gráfico a su derecha. En este segundo caso, un valor de tiempo medio hasta fallo peligroso superior con respecto del original incrementará la pendiente de la recta; en el caso contrario, esto es, si la variable resulta inferior a su valor inicial, la pendiente se verá reducida.

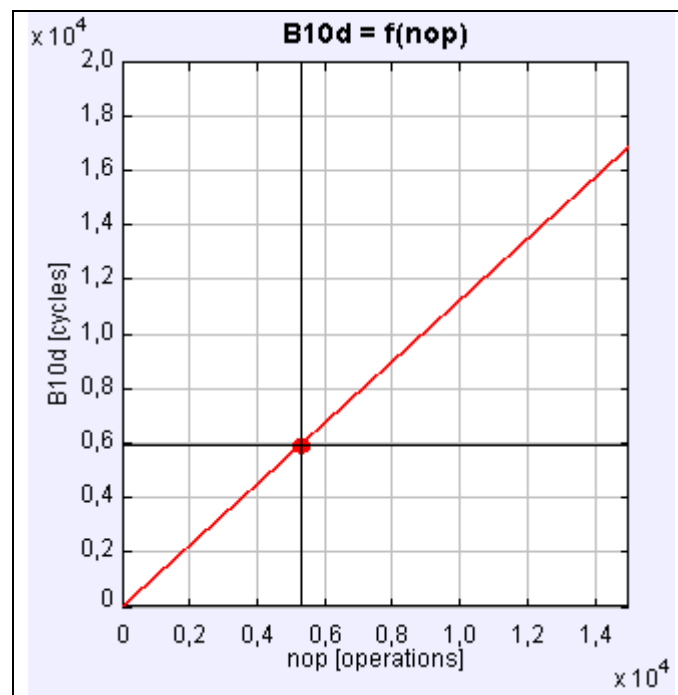


Figura 5.14. Relación $B_{10d} = f(n_{op})$.

Gráfico 2: relación entre $MTTF_d$ y PFH_d . Representa la relación entre el tiempo medio hasta fallo peligroso y la probabilidad de un fallo peligroso por hora. La curva trazada se corresponde con una de las opciones presentadas en la Tabla 4.2., para la cual se precisa conocer la categoría y la cobertura de diagnóstico del sistema; el primer parámetro es especificado por el usuario al seleccionar el sistema que se desea evaluar, mientras que el segundo es tomado en consideración por el propio programa de acuerdo a las aclaraciones indicadas en el Capítulo 4.

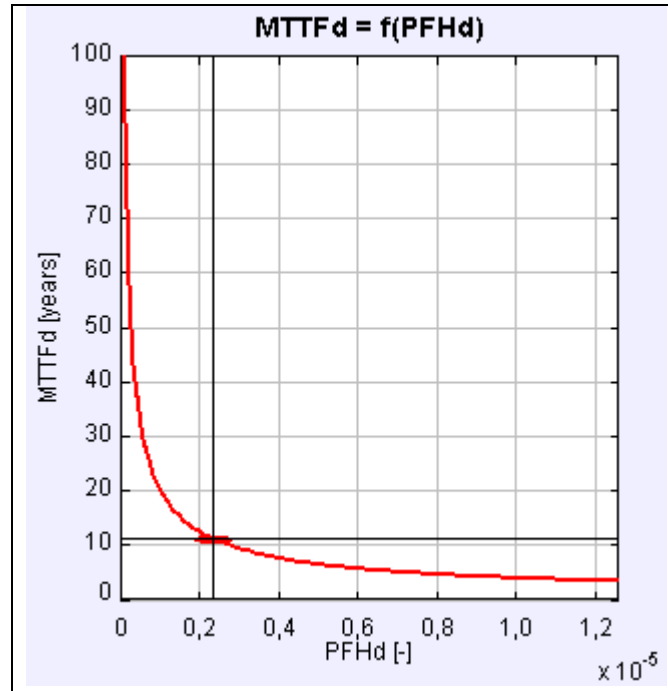


Figura 5.15. Relación $MTTF_d = f(PFH_d)$. Caso particular de un sistema de Cat. 3.

A diferencia del gráfico anterior, los parámetros de tiempo medio hasta fallo peligroso y de probabilidad de fallo peligroso por hora de cada parte relativa a la seguridad están ligados entre ellos. Este hecho supone que cualquier manipulación de dichas variables en cualquiera de las pestañas entraña una modificación automática de los mismos parámetros en los otros dos paneles, de tal manera que se asegura en todo momento el cumplimiento de las expresiones (2.1) y (2.7). Estas relaciones establecen que el tiempo medio hasta fallo peligroso y la probabilidad de fallo peligroso por hora de un sistema de control se expresen, respectivamente, como la composición de los tiempos medios y de las probabilidades individuales de sus partes relativas a la seguridad. Con tal propósito, las expresiones detalladas en (5.2) muestran el comportamiento del conjunto de probabilidades de fallo peligroso por hora de los componentes de seguridad ante una variación de su valor. La diferencia de valor advertida por un elemento se reparte de manera equitativa entre los otros dos dispositivos del sistema.

$$\begin{aligned}
 PFH_d^f(A) &= PFH_d^i(A) \pm \Delta PFH_d \\
 PFH_d^f(B) &= PFH_d^i(B) \mp \frac{1}{2} \cdot \Delta PFH_d \\
 PFH_d^f(C) &= PFH_d^i(C) \mp \frac{1}{2} \cdot \Delta PFH_d
 \end{aligned}
 \tag{5.2}$$

Los índices *i* y *f* señalan los valores inicial y final de probabilidad, respectivamente; A, B y C constituyen tres dispositivos de control cualquiera, cada uno de una tipología

diferente (entrada, lógico y salida); y ΔPFH_d corresponde a la variación de probabilidad de fallo peligroso por hora sufrida. El tiempo medio hasta fallo peligroso es actualizado directamente a partir de los nuevos valores de probabilidad, conocida la relación correspondiente de la Tabla 4.2.

Si, como se ha indicado previamente, se opta por fijar el valor de probabilidad de fallo peligroso por hora de una de las partes relativas a la seguridad, las ecuaciones anteriores se simplifican. Sea C el componente de seguridad cuya probabilidad de fallo peligroso por hora resta invariable. Por consiguiente, se establece que:

$$\begin{aligned} PFH_d^f(A) &= PFH_d^i(A) \pm \Delta PFH_d \\ PFH_d^f(B) &= PFH_d^i(B) \mp \Delta PFH_d \\ PFH_d^f(C) &= PFH_d^i(C) \end{aligned} \quad (5.3)$$

Cuando dos de las tres probabilidades de fallo peligroso por hora son fijadas, el efecto que se tiene sobre el sistema es el mismo que si las tres probabilidades se hubieran fijado, ya que, en caso de existir una variación en la única variable no fijada, ningún componente sería capaz de asimilar la diferencia de valor experimentada.

Cualquier manipulación de los gráficos que suponga una modificación de los parámetros definitorios de las partes relativas a la seguridad se verá reflejada en el panel de visualización en el lado izquierdo de su respectivo panel de trabajo.

5.2.4. Área de mensajes.

El área de mensajes es un elemento que se ubica en la región inferior de la ventana principal del programa y que ocupa la totalidad de la misma. En ella se muestran los mensajes emitidos por la propia interfaz informando acerca de la realización de una determinada acción, de la ejecución satisfactoria o no de ésta, así como de los posibles errores que pueden tener lugar en el transcurso de su operación, en particular, la falta de definición de ciertos parámetros necesarios para emprender acciones concretas o cálculos.

5.2.5. Área de dispositivos.

El área de dispositivos corresponde a una ventana emergente que resulta tras presionar sobre el botón “Dispositivo de seguridad”, ubicado en la zona inferior del área de especificaciones del caso de estudio, introducida con anterioridad. Esta nueva ventana presenta de manera general el conjunto de dispositivos que cumplen con los requerimientos de seguridad definidos por el usuario en el área de trabajo mencionada. Se estructura en cuatro pestañas. La primera de ellas (ver Figura 5.16.) indica la

cantidad de componentes de seguridad de cada tipo (dispositivo de entrada, unidad lógica o dispositivo de salida) que se adecuan a las características señaladas por el usuario. Las tres pestañas restantes están dedicadas a listar de manera separada dichos elementos según su tipo. Con tal de que la función del área de dispositivos pueda ejecutarse satisfactoriamente, se requiere que, previo a su uso, se introduzca en el programa una lista con componentes de seguridad con el formato adecuado a través del botón de la barra de herramientas “Lista de dispositivos”.

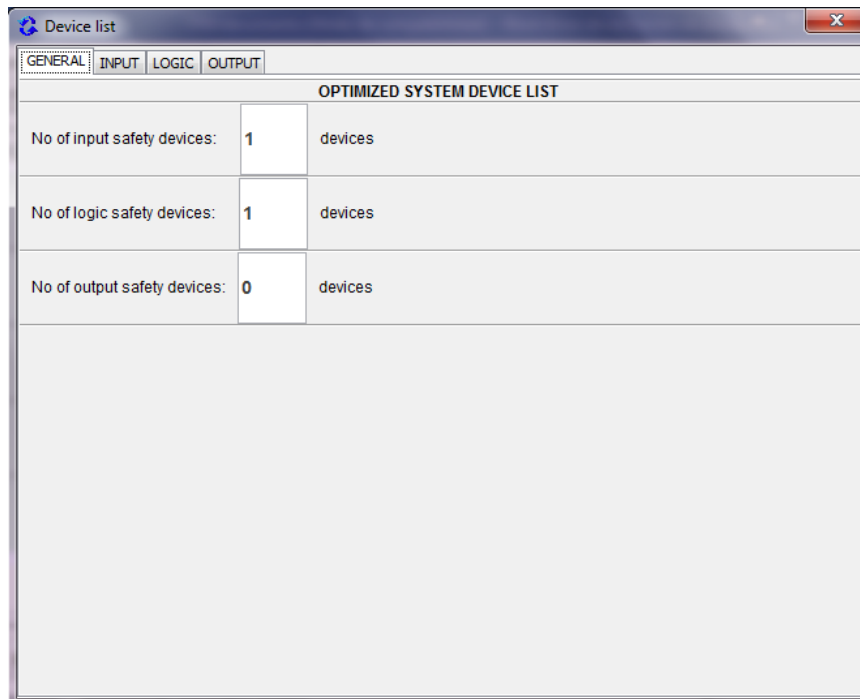


Figura 5.16. Ventana emergente: Dispositivos de seguridad.

Los componentes de seguridad son seleccionados en base a su probabilidad de fallo peligroso y su tiempo medio hasta fallo peligroso, por un lado, y a la tecnología implementada en ellos, por el otro. El gráfico ilustrado en la Figura 5.17. es representativo de este primer criterio. En efecto, partiendo de la gráfica de la Figura 5.15., dado un punto de la curva definido por un par de valores de probabilidad hasta fallo peligroso y de tiempo medio hasta fallo peligroso, se escogen aquellos componentes de seguridad cumpliendo con una probabilidad igual o inferior a la indicada (el dispositivo tiene menos tendencia a fallar) y con un tiempo medio igual o superior al descrito (la duración entre fallos consecutivos es mayor). Dividiendo la gráfica inicial en cuatro cuadrantes en torno al punto de referencia, los dispositivos de seguridad buscados se hallan en la región superior izquierda, la cual se muestra rayada en la imagen de la figura. Dichos elementos tienen como característica común la cobertura de los requisitos mínimos precisados para avalar la seguridad del sistema.

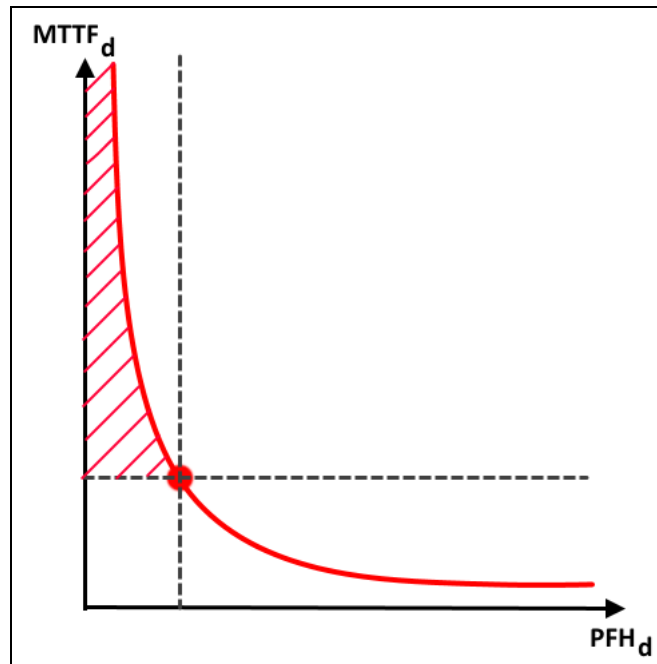


Figura 5.17. Criterio de selección de los dispositivos de seguridad.

En un segundo plano, la tecnología acota las opciones disponibles en la selección anterior, ya que, a pesar de que todos los componentes parten de un mismo análisis inicial, aquéllos de naturaleza mecánica, dado el desgaste que supone su actividad, disponen de una serie de parámetros adicionales dependientes del número de ciclos de operación, cuya manipulación a través del gráfico de la Figura 5.14. permite igualmente realizar una selección individualizada que no sería posible en componentes hidráulicos, eléctricos o electrónicos, al no existir tal relación. La clase de componente es especificada en el bloque de características en el lado izquierdo del panel de trabajo, para cada una de las partes relativas a la seguridad del sistema de control evaluadas.

En la ventana emergente presentada al principio de este apartado, dentro de las tres últimas pestañas dedicadas a la visualización particular de los componentes de seguridad, se dan diferentes criterios de ordenación de los elementos seleccionados que el usuario puede elegir mediante la opción pertinente en una barra de herramientas en su parte superior.

- Por defecto. Los componentes de seguridad siguen el mismo orden que el indicado en la lista de dispositivos introducida en el programa.
- PL. Los componentes de seguridad se ordenan de acuerdo a una implicación creciente en la seguridad del sistema, reflejada en su nivel de fiabilidad ($a > b > c > d > e$).

- PFH_d. Los componentes de seguridad se ordenan de acuerdo a un orden decreciente de su probabilidad de fallo peligroso por hora.
- MTTF_d. Los componentes de seguridad se ordenan de acuerdo a un orden decreciente de su tiempo medio hasta fallo peligroso.
- Tecnología. Los componentes de seguridad se ordenan de acuerdo a la tecnología que se implementa en ellos, siguiendo el orden arbitrario: hidráulico > neumático > mecánico > eléctrico > electrónico > electrónico programable.
- Precio. Los componentes de seguridad se ordenan de acuerdo a un orden decreciente de su precio de venta.

5.3. Procedimiento de diseño.

En los siguientes apartados se detalla el procedimiento que debe seguir el usuario del programa para definir satisfactoriamente su sistema de control y extraer de su correspondiente análisis un listado de componentes de seguridad cumpliendo con las características indicadas por éste. En general, el conjunto de pasos a cumplir es listado en la ventana emergente “Ayuda” de la Figura 5.9., presentada con anterioridad, a la que el usuario puede acceder presionando sobre el botón con el mismo nombre ubicado en la barra de herramientas. Un enfoque alternativo para la comprensión del proceso de diseño es ilustrado en la Figura 5.18., donde se representa de manera esquemática el desglose de los elementos de un sistema, desde el nivel más general comprendido por la herramienta informática, representado por una máquina o un entorno de trabajo, hasta las características particulares de cada una de las partes relativas a la seguridad que lo constituyen.

El procedimiento de diseño puede estructurarse en tres bloques principales: (A) Definición del sistema; (B) Manipulación del sistema; y (C) Visualización de los resultados. A continuación, se puntualiza en las particularidades de cada bloque.

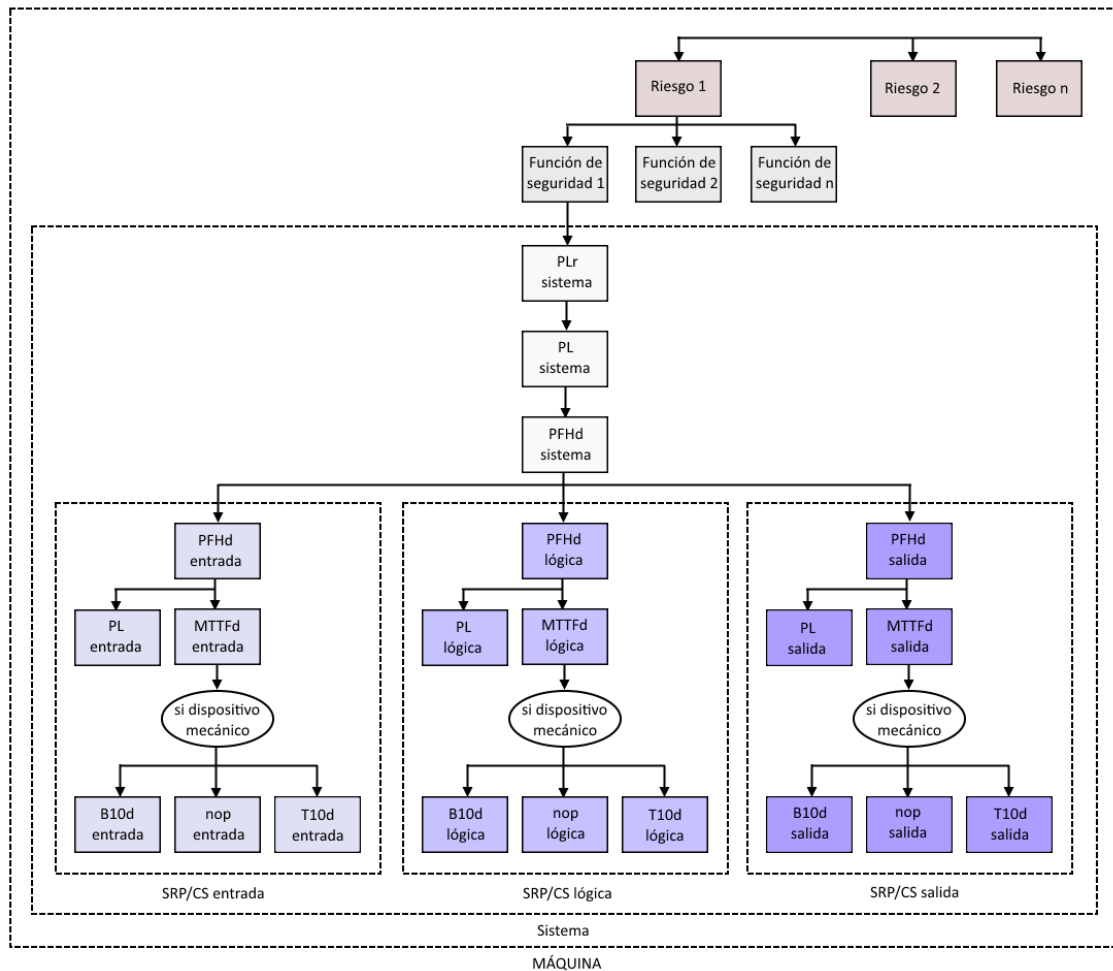


Figura 5.18. Esquema general de la metodología propuesta, basada en la norma EN ISO 13849-1.

5.3.1. Definición del sistema.

La definición del sistema incluye el conjunto de operaciones dedicadas a delimitar los elementos que constituyen el sistema de control a evaluar. Las acciones realizadas en este bloque deben ser efectuadas de manera sucesiva a como se indica en el presente apartado, ya que el programa no contempla un orden alternativo para la definición de dichos elementos.

Añadir una máquina. El primer paso consiste en la precisión de la máquina o entorno de trabajo que representa el sistema. Para ello, el usuario debe presionar sobre el botón “Máquina”. De este modo, aparece una ventana emergente donde puede concretarse el nombre del elemento en el campo disponible. Al escribir sobre el área de texto, ésta se torna de color amarillo. El usuario debe presionar primeramente la tecla “Enter” para validar el nombre y seguidamente el botón “Añadir máquina”, en la parte inferior de la

ventana emergente, para validar la acción y cerrar la ventana. El usuario solamente puede definir una sola máquina o entorno de trabajo.

Añadir un riesgo. El segundo paso consiste en la precisión del riesgo asociado a la máquina o al entorno de trabajo. El conjunto de operaciones seguidas es similar a como se ha procedido anteriormente. El usuario presiona sobre el botón “Riesgo”. De este modo, aparece una ventana emergente donde puede concretarse el nombre del riesgo en el campo disponible. Al escribir sobre el área de texto, ésta se torna de color amarillo. El usuario debe presionar primeramente sobre la tecla “Enter” para validar el nombre y seguidamente el botón “Añadir riesgo”, en la parte inferior de la ventana emergente, para validar la acción y cerrar la ventana. El usuario puede definir hasta un total de 100 riesgos para la máquina indicada precedentemente.

Añadir una función de seguridad. El tercer paso consiste en la precisión de la función de seguridad asociada a un riesgo. Para ello, el usuario debe presionar sobre el botón “Función de seguridad”. De este modo, aparece una ventana emergente donde se pide especificar dos campos en la forma de pestañas desplegadas. En el primer campo debe indicarse el riesgo, creado con anterioridad, al que debe hacer referencia la función de seguridad que se desea crear. En el segundo campo debe seleccionarse la función de seguridad que más se adecue a los intereses del usuario de una lista conteniendo 12 opciones diferentes. El usuario debe presionar a continuación el botón “Añadir función de seguridad”, en la parte inferior de la ventana emergente, para validar la acción y cerrar la ventana. El usuario puede definir hasta un total de 12 funciones de seguridad diferentes para cada uno de los riesgos indicados precedentemente.

Añadir un nivel de fiabilidad. El cuarto y último paso del bloque consiste en la precisión del nivel de fiabilidad de la función de seguridad asociada al riesgo. Para ello, el usuario debe presionar sobre el botón “Nivel de fiabilidad”. De este modo, aparece una ventana emergente donde se pide especificar cinco campos en la forma de pestañas desplegadas. En el primer campo debe indicarse el riesgo de la función de seguridad que se desea evaluar. En el segundo campo debe indicarse la función de seguridad para la que se quiere determinar su nivel de fiabilidad. En los campos restantes debe indicarse la severidad del riesgo, la frecuencia de su ocurrencia y la probabilidad de evitarse, escogiendo entre las dos opciones mostradas en cada caso. El usuario debe presionar seguidamente el botón “Añadir nivel de fiabilidad”, en la parte inferior de la ventana emergente, para validar la acción y cerrar la ventana. El usuario solamente puede definir un solo nivel de prestaciones para cada nivel de fiabilidad indicado precedentemente.

5.3.2. Manipulación del sistema.

Llegados a este punto, el sistema se halla completamente definido y podrá pasarse a una segunda fase de análisis llevada a cabo en un nivel de detalle superior e

individual para cada caso de estudio (riesgo + función de seguridad + nivel de fiabilidad) que desembocará en la selección final de las partes relativas a la seguridad del sistema de control. En aquellas situaciones en que dos o más riesgos compartan una misma función de seguridad, se ejecutarán sendas evaluaciones para cada uno de los riesgos involucrados.

Selección del riesgo. El primer paso consiste en la precisión del riesgo de la función de seguridad que se desea evaluar. Para ello, el usuario utiliza el campo en forma de pestaña desplegable con el nombre “Riesgo”, ubicado en el recuadro de la parte inferior de la interfaz con el título “Especificaciones”. Las opciones disponibles en la pestaña abarcan el conjunto de riesgos definidos por el usuario con anterioridad.

Selección de la función de seguridad asociada al riesgo. El segundo paso consiste en la precisión de la función de seguridad asociada al riesgo indicado anteriormente, que se desea evaluar. Para ello, el usuario utiliza el campo en forma de pestaña desplegable con el nombre “Función de seguridad”, ubicado en el recuadro de la parte inferior de la interfaz con el título “Especificaciones”. Las opciones disponibles en la pestaña abarcan el conjunto de funciones de seguridad definidas por el usuario con anterioridad. Una vez seleccionado el riesgo y la función de seguridad correspondiente, en el área de texto directamente a la izquierda de ambos campos se inscribe inmediatamente el nivel de fiabilidad correspondiente al conjunto de ambos elementos.

Selección de la categoría. El tercer paso consiste en la precisión de la estructura de la función de seguridad seleccionada. Para ello, el usuario utiliza el campo en forma de pestaña desplegable con el nombre “Categoría”, ubicado en el recuadro de la parte inferior de la interfaz con el título “Especificaciones”. Las opciones disponibles en la pestaña son dependientes del nivel de fiabilidad mostrado en el área de texto correspondiente.

Optimización del sistema. El cuarto paso consiste en la ejecución del proceso de optimización del programa. Para ello, el usuario debe presionar, posterior a la especificación de los campos anteriores, sobre el botón “Optimizar”, ubicado en el recuadro de la parte inferior de la interfaz con el título “Especificaciones”. Indicada la acción, se muestra en la pestaña del sistema general un punto sobre el gráfico posicionando el sistema con respecto del nivel de fiabilidad, la categoría, la cobertura de diagnóstico y el tiempo medio hasta fallo peligroso del mismo. En las tres pestañas concernientes a las partes relativas a la seguridad, se visualizan una serie de curvas mostrando la relación existente entre varios de sus parámetros.

Manipulación de los gráficos de las pestañas. El quinto y último paso del bloque consiste en el empleo de los gráficos disponibles en cada pestaña concernientes a las partes relativas a la seguridad. El usuario manipula la magnitud de las variables hasta alcanzar las características de seguridad deseadas por éste. La visualización de la variación de los parámetros puede observarse a través del área ubicada en la parte

izquierda de las pestañas, donde, aparte de reflejarse los valores de los dispositivos, el usuario puede indicar la naturaleza del mismo, así como la fijación de la probabilidad de fallo peligroso por hora en un valor concreto, inhabilitando el uso del gráfico derecho en la misma pestaña.

5.3.3. Visualización de los resultados.

Aunque el programa asista al diseño y selección de los componentes de seguridad de un sistema de control, es el propio usuario el responsable de usar sus conocimientos y experiencia para, siguiendo un criterio adecuado, seleccionar los dispositivos de seguridad apropiados. Partiendo del análisis definido en el apartado anterior, el programa permite visualizar los resultados de diferentes maneras en función de los intereses del usuario.

Añadir una lista de componentes de seguridad. El primer paso consiste en la inserción de un archivo conteniendo una lista de dispositivos de seguridad y sus características. Para ello, el usuario debe presionar sobre el botón “Lista de componentes”. De este modo, aparece una ventana emergente donde se pide especificar el nombre del fichero en el campo disponible; éste debe tener la extensión adecuada y su contenido debe adecuarse al formato especificado con anterioridad en el Apartado 5.2.1. El usuario debe presionar seguidamente el botón “Abrir”, en la parte inferior de la ventana emergente, para validar la acción y cerrar la ventana.

Visualizar los componentes de seguridad resultantes. El usuario puede visualizar los componentes de seguridad que resultan de la especificación de sus características. Para ello, el usuario debe presionar sobre el botón “Componentes de seguridad”, ubicado en el recuadro de la parte inferior de la interfaz con el título “Especificaciones”. De este modo, aparece una ventana emergente con diferentes pestañas, la primera de ellas indicando el número total de dispositivos cumpliendo con los requerimientos impuestos. En las pestañas restantes, se muestra el conjunto de componentes y sus propiedades de según su tipo (entrada, lógico o salida). El usuario puede seleccionar en la parte superior de cada pestaña el criterio que estime más óptimo para la ordenación de los elementos de la misma.

Visualizar una solución optimizada económicamente. El usuario puede visualizar el conjunto de tres componentes de seguridad que cumplen con las especificaciones de seguridad impuestas y, al mismo tiempo, minimizan la inversión de su coste. Para ello, el usuario debe presionar sobre el botón “Optimización económica”, ubicado en el recuadro de la parte inferior de la interfaz con el título “Especificaciones”. De este modo, aparece una ventana emergente donde se identifican los componentes de seguridad pertinentes, junto con sus características y su precio. En la parte superior de la ventana se indica la probabilidad de fallo peligroso por hora global, así como la inversión total realizada con los tres elementos.

6

Aplicación de la herramienta

Legados a este capítulo, se tiene un programa informático capaz de aplicar las directrices expuestas en la norma EN ISO 13849-1. Sin embargo, aún queda por comprobar su aplicabilidad real en los sistemas industriales. Con tal propósito, se han seleccionado dos ejemplos extraídos de la normativa y de manuales redactados por fabricantes de componentes de seguridad.

6.1. Introducción.

En este capítulo se recogen dos casos prácticos que ya han sido resueltos previamente y para los que se tienen resultados factibles y concretos. El objetivo de los siguientes apartados es la resolución de dichos ejemplos a través de la herramienta informática programada. La utilización del programa en sistemas de arquitectura dispar permite, a través de la comparación de los resultados obtenidos con sus homólogos teóricos, observar las capacidades y limitaciones del mismo, así como determinar la veracidad y la precisión con se hallan las características de las partes relativas a la seguridad del sistema de control, demostrando la viabilidad de su uso.

Los casos prácticos se han seleccionado de tal forma que abarquen el mayor espectro posible de situaciones. El primero de ellos, extraído del manual de seguridad de sistemas de control de la empresa ABB, corresponde a un sistema con múltiples entradas y múltiples salidas, compartiendo todas ellas una misma unidad lógica. En el segundo caso, escogido entre los ejemplos de los anexos de la norma EN ISO 13849-1, se trabaja con un sistema de doble canal compuesto por un número impar de elementos. En ambos casos, se deberán aplicar las transformaciones pertinentes sobre el sistema para favorecer su análisis.

El capítulo se organiza de la siguiente manera. En la Sección 6.2. se expone el primer ejemplo mencionado en esta introducción. El desarrollo del caso parte de una primera presentación del mismo, al cual le sigue el cálculo de los resultados teóricos y la aplicación del programa informático para la obtención de unos valores prácticos de

los parámetros de seguridad evaluados. En la Sección 6.3. se realiza el estudio del segundo ejemplo, empleando para ello una estructura análoga a la indicada en el caso anterior por lo que concierne a su planteamiento y posterior presentación de los resultados. En la Sección 6.4. se extraen diversas conclusiones a partir de la comparación de los valores determinados en los ejemplos precedentes con sus respectivos equivalentes teóricos.

6.2. Ejemplo 1: célula de trabajo.

6.2.1. Presentación del caso.

Una célula está compuesta por un robot (Máquina Q1), una prensa hidráulica (Máquina Q2) y una herramienta de mecanizado neumática (Máquina Q3). Con el fin de garantizar el funcionamiento seguro de las tres máquinas, se emplea el autómata de seguridad K1, el cual está ligado a una cortina fotoeléctrica F1, un interruptor bloqueado B1 y un botón de paro de emergencia G1. La implementación de dichos dispositivos permite el control de la entrada de personal en la célula y detener de forma segura el movimiento peligroso de los elementos de ésta si fuera necesario. El diagrama de bloques de la Figura 6.1. representa esquemáticamente la interconexión de los diferentes componentes de la célula.

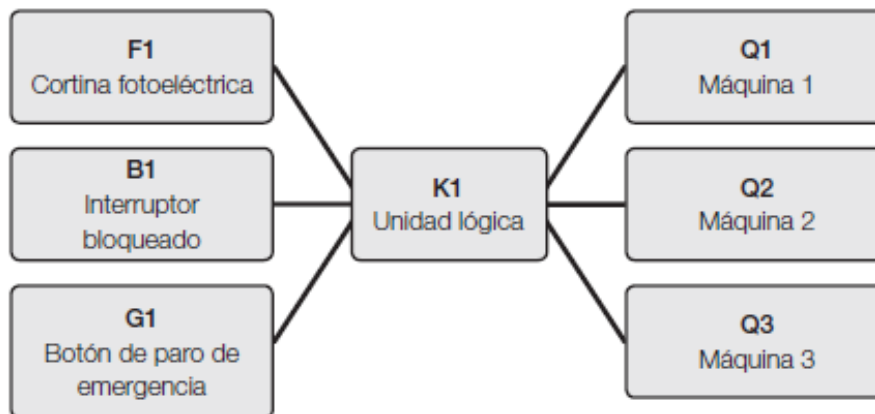


Figura 6.1. Diagrama de bloques de la interconexión de los elementos de la célula.

6.2.2. Resolución teórica.

Previo a la fase de cálculo, se delimita el número de funciones de seguridad a evaluar en base al número de dispositivos y de máquinas incluidos en la estructura. Cada dispositivo define una función de seguridad única para cada máquina que resulte

afectada por su actividad. Por consiguiente, en el caso propuesto, cada una de las tres máquinas indicadas en el esquema de la Figura 6.1. se ve controlada por cada uno de los tres componentes de seguridad conectados al autómata. Esto supone fijar hasta nueve funciones de seguridad diferentes. A modo de ejemplo, solamente se estudiará el conjunto de funciones de seguridad relativas al interruptor bloqueado B1, las cuales se ilustran en los diagramas de bloques de la Figura 6.2.

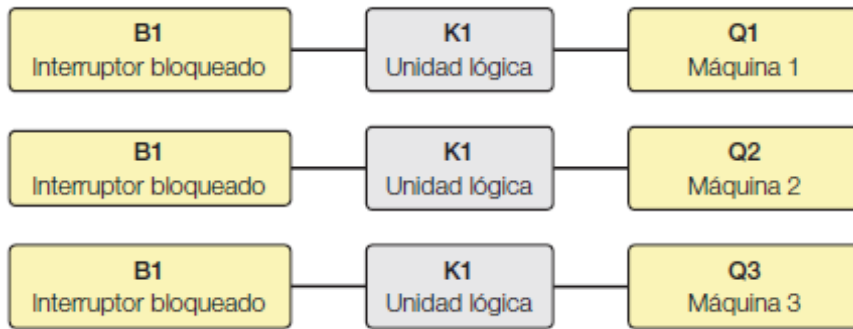


Figura 6.2. Funciones de seguridad relativas al componente B1.

Para cada función de seguridad definida, se realiza una evaluación de los riesgos asociados a las máquinas. Tanto en el caso del robot M1 como en el de la prensa hidráulica M2, se tienen sendos riesgos suponiendo un daño grave e irreversible (S2), con una ocurrencia frecuente (F2) y difícilmente evitables (P2). Con ello se concluye, haciendo uso del diagrama de árbol presentado en la Figura 3.3., un nivel de prestaciones requerido igual a “e”. La evaluación de la herramienta de mecanizado neumática M3 conduce a un nivel de fiabilidad requerido igual a “d”, debido a que, a pesar de ser un riesgo igualmente grave (S2) y frecuente (F2), se contempla al mismo tiempo la posibilidad de evitarlo (P1).

El análisis de la seguridad es llevado a cabo a partir de la probabilidad de fallo peligroso por hora característico de cada canal, el cual es hallado por adición de las probabilidades de fallo peligroso por hora de los componentes que integra. El valor resultante de dicha operación es identificado con un nivel de fiabilidad específico que debe igualarse o superar el nivel requerido. En la tabla que se muestra a continuación se listan las probabilidades de fallo peligroso por hora de los componentes de los casos de estudio considerados.

Componente	PFH _d [-]
B1 Interruptor bloqueado	$4,5 \cdot 10^{-9}$
K1 Unidad lógica	$2 \cdot 10^{-9}$
Q1 Máquina 1	$5,79 \cdot 10^{-8}$

Q2 Máquina 2	$8 \cdot 10^{-8}$
Q3 Máquina 3	$2 \cdot 10^{-7}$

Tabla 6.1. PFH_d de los componentes de las funciones de seguridad.

Para cada función de seguridad se determinan los siguientes resultados. En paréntesis se indica el nivel de fiabilidad requerido PL_r y el alcanzado por el canal PL.

- Robot M1 ($PL_r = e \mid PL = e$).

$$PFH_d(B1) + PFH_d(K1) + PFH_d(M1) = 4,5 \cdot 10^{-9} + 2 \cdot 10^{-9} + 5,79 \cdot 10^{-8} = 6,44 \cdot 10^{-8}$$

- Prensa hidráulica M2 ($PL_r = e \mid PL = e$).

$$PFH_d(B1) + PFH_d(K1) + PFH_d(M2) = 4,5 \cdot 10^{-9} + 2 \cdot 10^{-9} + 8 \cdot 10^{-8} = 8,65 \cdot 10^{-8}$$

- Herramienta de mecanizado neumática ($PL_r = d \mid PL = d$).

$$PFH_d(B1) + PFH_d(K1) + PFH_d(M3) = 4,5 \cdot 10^{-9} + 2 \cdot 10^{-9} + 2 \cdot 10^{-7} = 2,07 \cdot 10^{-7}$$

Puede observarse que, con los componentes de seguridad disponibles, se alcanza en todo momento el nivel de fiabilidad requerido para cada función de seguridad evaluada. Por tanto, la selección efectuada de los elementos de control es correcta.

6.2.3. Resolución experimental.

La determinación de las características de seguridad de los elementos del sistema por el intermedio del programa elaborado debe abordarse de manera separada para cada conjunto de tres elementos definido.

- Robot M1. El nivel de fiabilidad requerido es concretado a partir de las indicaciones del riesgo detalladas en el enunciado. La única estructura que permite alcanzar un nivel de prestaciones “e” corresponde a una categoría 4, caracterizada por una cobertura de diagnóstico alta. De acuerdo a los criterios descritos con anterioridad en el Capítulo 4, un sistema de tales características tiene asociado un valor límite de probabilidad de fallo peligroso por hora de $9,54 \cdot 10^{-8}$, el cual, como se observa, describe una situación más crítica que la del modelo teórico. La repartición de dicho valor entre los diferentes componentes de seguridad es efectuada conforme a lo ilustrado en la Tabla 6.2.

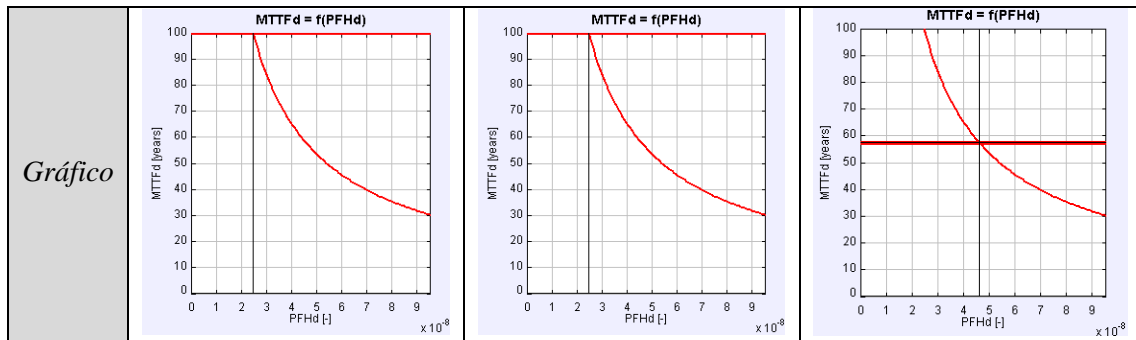
	Entada	Lógico	Salida
PL	e	e	e
PFH _d	2·10 ⁻⁸	2·10 ⁻⁸	5·10 ⁻⁸
MTTF _d	100 años	100 años	58 años
T _{10d}	10 años	-	6 años
Gráfico			

Tabla 6.2. Solución analítica de la función de seguridad relativa a M1.

Las características del dispositivo de salida son logradas. Sin embargo, el modelo escogido impide que los elementos puedan alcanzar una probabilidad de fallo peligroso superior a $2,47 \cdot 10^{-8}$, ya que en dicho caso el tiempo medio hasta fallo peligroso supera los 100 años. Por consiguiente, los valores indicados para el componente de entrada y para la unidad lógica son valores de referencia; cualquier dispositivo con una probabilidad inferior al valor señalado será válido y la función de seguridad estará asegurada.

- Prensa hidráulica. De manera análoga a como se ha procedido en la función de seguridad relativa al robot M1, se parte de un sistema con un nivel de fiabilidad “e”, una categoría 4 y una cobertura de diagnóstico alta. De ello se extrae el mismo límite de probabilidad de fallo peligroso por hora de $9,54 \cdot 10^{-8}$. La repartición de dicho valor entre los diferentes componentes de seguridad es efectuada conforme a lo ilustrado en la Tabla 6.3.

	Entada	Lógico	Salida
PL	e	e	e
PFH _d	2·10 ⁻⁸	2·10 ⁻⁸	5·10 ⁻⁸
MTTF _d	100 años	100 años	58 años
T _{10d}	10 años	-	-



Gráfico

Tabla 6.3. Solución analítica de la función de seguridad relativa a M2.

Nuevamente se está ante una situación donde los valores hallados no se corresponden exactamente con sus homólogos teóricos. Como se ha explicado en la función de seguridad precedente, la probabilidad de fallo peligroso por hora teórica de los componentes de seguridad de entrada y lógico no son alcanzables con el modelo en que se sustenta el programa y, por tanto, representan sendos valores máximos a partir de los cuales puede hacerse una selección más exhaustiva. En el caso del componente de salida, no puede obtenerse una probabilidad superior a $5 \cdot 10^{-8}$; de ser así, se superaría la probabilidad de fallo peligroso por hora asignada al sistema y los tiempos medio hasta fallo peligroso de los componentes de entrada y lógico superarían los 100 años. De lo contrario, si no existiera dicho límite, podrían conseguirse el conjunto de valores originales.

- Herramienta de mecanizado neumática. El nivel de fiabilidad requerido es concretado a partir de las indicaciones del riesgo detalladas en el enunciado. La estructura seleccionada se corresponde con una de categoría 3, que en el programa es caracterizada por una cobertura de diagnóstico baja. De acuerdo a los criterios descritos con anterioridad en el Capítulo 4, un sistema de tales características tiene asociado un valor límite de probabilidad de fallo peligroso por hora de $1 \cdot 10^{-6}$, el cual, como se observa, describe una situación más crítica que la del modelo teórico, con una probabilidad de $2,07 \cdot 10^{-7}$. La repartición de dicho valor entre los diferentes componentes de seguridad es efectuada conforme a lo ilustrado en la Tabla 6.4.

	Entada	Lógico	Salida
PL	d	d	d
PFH_d	$7 \cdot 10^{-7}$	$1 \cdot 10^{-7}$	$2 \cdot 10^{-7}$
$MTTF_d$	27 años	100 años	65 años
T_{10d}	3 años	-	7 años

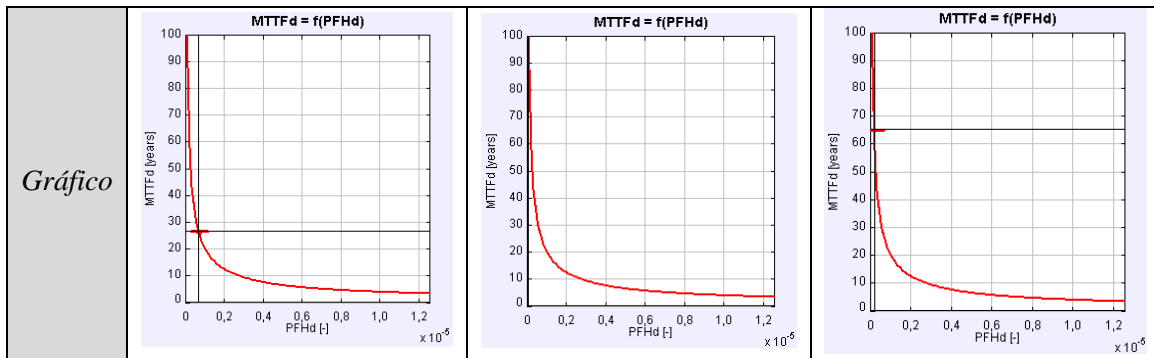


Tabla 6.4. Solución analítica de la función de seguridad relativa a M3.

De manera similar a como ocurre con las otras dos funciones de seguridad del sistema de control, las características teóricas de los componentes de seguridad solamente son alcanzables en uno de los tres elementos del canal, esto es, únicamente en aquel dispositivo cuya probabilidad de fallo peligroso por hora es lo suficientemente alta como para estar comprendida dentro del rango admisible de valores que la arquitectura y la cobertura de diagnóstico limitan. Para los otros dos dispositivos, se dispone de un valor de probabilidad que debe ser tomado como referencia para la selección de elementos teniendo una probabilidad de fallo peligroso por hora igual o inferior al valor indicado. A pesar de ello, y al igual que sucede en los dos casos expuestos anteriormente, los valores teóricos son siempre inferiores a los resultados deducidos por el programa informático y, por tanto, cumplen con el criterio de seguridad establecido.

6.3. Ejemplo 2: sistema redundante.

6.3.1. Presentación de caso.

Un circuito de mando para el control del enclavamiento de un resguardo está compuesto de dos canales redundantes, uno electromecánico y otro electrónico programable. El primer canal comprende un interruptor de puerta (SW1B) dotado de contactos de apertura forzada, el cual se encuentra conectado a un contactor (K1B) permitiendo desactivar la alimentación del motor eléctrico de la máquina. El segundo canal comprende un segundo interruptor de puerta (SW2) de naturaleza electrónica programable conectado a un autómata industrial programable (PLC) controlando un convertidor de corriente (CC) a través del cual es posible desconectar la alimentación del motor de la máquina. Un detector de rotación (RS) ajeno al canal es utilizado para comprobar el convertidor de corriente. El diagrama de bloques de la Figura 6.3. representa esquemáticamente la interconexión de los diferentes componentes del circuito de mando.

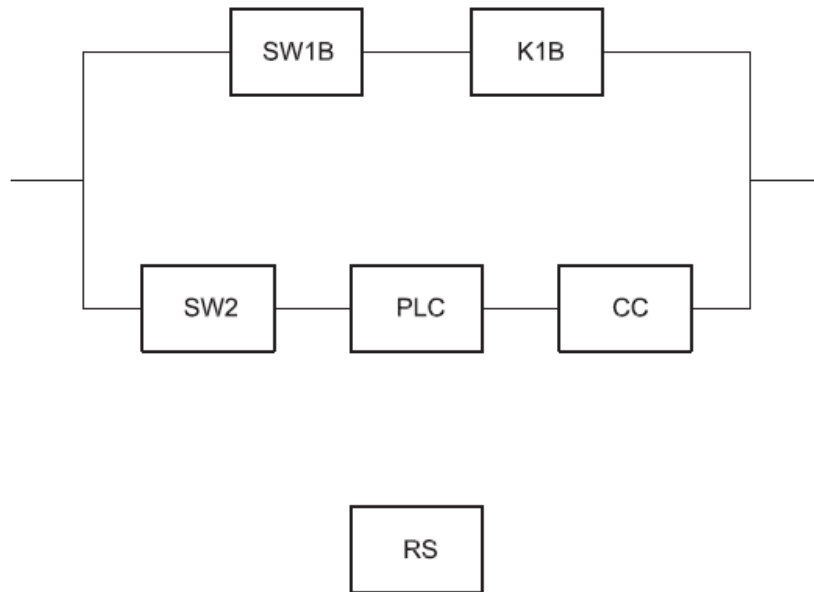


Figura 6.3. Diagrama de bloques de los elementos del sistema.

La función de seguridad del sistema se establece tal que el movimiento peligroso es detenido al abrirse el resguardo, una vez desconectada la alimentación del motor eléctrico de la máquina.

6.3.2. Resolución teórica.

El riesgo que se desea controlar es considerado como grave (S2), de frecuencia rara y duración de exposición corta (F1), y evitable en determinadas condiciones (P1). Partiendo del diagrama de árbol de decisiones de la Figura 3.3., la combinación de las tres condiciones definidas conlleva a un nivel de fiabilidad requerido “c”.

El análisis de la seguridad es llevado a cabo a partir de la lectura del diagrama de la Figura 3.4., en la que el nivel de fiabilidad resulta de la combinación de la arquitectura del sistema, su cobertura de diagnóstico y su tiempo medio hasta fallo peligroso equivalente. Para ello, se dispone de las características de los componentes de la función de seguridad suministradas por el fabricante, las cuales son listadas en la tabla que se muestra a continuación.

Componente	MTTF _d [años]	DC [%]
SW1B Interruptor de puerta	-	-
K1B Contactor	30	99
SW2 Interruptor de puerta	20	60

PLC Autómata industrial programable	20	30
CC Convertidor de corriente	20	90
RS Detector de rotación	-	-

Tabla 6.5. Características de los componentes de la función de seguridad.

El tiempo medio hasta fallo peligroso del sistema de control se deduce mediante los tiempos medio hasta fallo peligroso individuales de sus canales.

- Canal 1 ($MTTF_d = 30$ años). Dadas las características funcionales del interruptor SW1B, el contactor K1B es el único elemento que contribuye a la seguridad del canal y que, por tanto, es considerado en el cálculo del tiempo.

$$\frac{1}{MTTF_d(C1)} = \frac{1}{MTTF_d(K1B)} = \frac{1}{30} = \frac{0,03}{\text{años}}$$

- Canal 2 ($MTTF_d = 6,67$ años).

$$\frac{1}{MTTF_d(C2)} = \frac{1}{MTTF_d(SW2)} + \frac{1}{MTTF_d(PLC)} + \frac{1}{MTTF_d(CC)} = \frac{1}{20} + \frac{1}{20} + \frac{1}{20} = \frac{0,15}{\text{años}}$$

Debido a que se trata de un sistema de doble canal donde ambos canales no son idénticos, el tiempo medio hasta fallo peligroso de un sistema de canal individual equivalente puede evaluarse mediante la expresión (2.2) presentada en el Capítulo 2. El valor obtenido del cálculo corresponde a 20 años.

Por lo que respecta a la cobertura de diagnóstico del circuito de mando, ésta es determinada en base a los tiempos medio hasta fallo peligroso y las coberturas de diagnóstico de los dispositivos que lo componen. Por aplicación de la expresión (2.6) del Capítulo 2, se halla una cobertura diagnóstico media de 67,1%.

$$DC_{avg} = \frac{\frac{DC(SW2)}{MTTF_d(SW2)} + \frac{DC(K1B)}{MTTF_d(K1B)} + \frac{DC(PLC)}{MTTF_d(PLC)} + \frac{DC(CC)}{MTTF_d(CC)}}{\frac{1}{MTTF_d(SW2)} + \frac{1}{MTTF_d(K1B)} + \frac{1}{MTTF_d(PLC)} + \frac{1}{MTTF_d(CC)}} = 67,1\%$$

Puede constatar que la combinación de la estructura del sistema, pudiéndose asociar a una categoría 3, junto con un tiempo medio hasta fallo peligroso medio y una cobertura de diagnóstico baja equivale, efectivamente, a un nivel de prestaciones “c”. Por tanto, la selección efectuada de los elementos de control es correcta.

6.3.3. Resolución experimental.

La determinación de las características de seguridad de los elementos del sistema de control por el intermedio del programa elaborado es abordada de manera separada para cada canal del mismo. A diferencia del primer ejemplo, donde cada dispositivo de entrada y de salida definían una función de seguridad única, en este caso los dos canales del sistema son comprendidos en una misma función de seguridad. El nivel de fiabilidad requerido de dicha función de seguridad es concretado a partir de las indicaciones de su riesgo detalladas en el enunciado. De manera similar, la categoría de ésta y su cobertura de diagnóstico son directamente dadas en la descripción del circuito de mando. De acuerdo a los criterios descritos con anterioridad en el Capítulo 4, un sistema con un nivel de prestaciones requerido “c”, una categoría 3 y una cobertura de diagnóstico baja tiene asociado un valor límite de probabilidad de fallo peligroso por hora de $3 \cdot 10^{-6}$. Dado que el análisis de la seguridad es realizado en el ejemplo en base al tiempo medio hasta fallo peligroso de los componentes de seguridad, se manipulan las gráficas pertinentes del programa hasta alcanzar la probabilidad de fallo peligroso por hora asociada al valor deseado de tiempo medio hasta fallo peligroso.

- **Canal 1.** La repartición del valor límite de probabilidad de fallo peligroso por hora entre los diferentes componentes del canal es efectuada conforme a lo ilustrado en la Tabla 6.5.

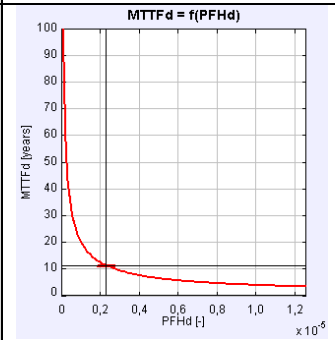
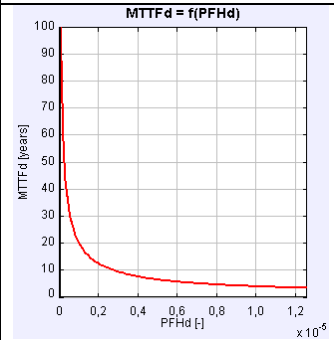
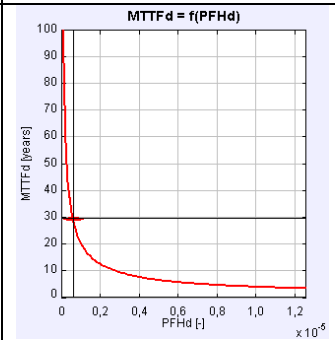
	Entada	Lógico	Salida
<i>PL</i>	c	d	d
<i>PFH_d</i>	$2,29 \cdot 10^{-6}$	$1 \cdot 10^{-7}$	$6,1 \cdot 10^{-7}$
<i>MTTF_d</i>	11 años	100 años	30 años
<i>T_{10d}</i>	-	-	3 años
Gráfico			

Tabla 6.5. Solución analítica del canal 1.

Las características del dispositivo del canal, identificado en el programa como un componente de salida, son alcanzadas. Sin embargo, la única manera de indicar al programa la ausencia de uno o varios componentes es minimizando el

aporte de probabilidad de fallo peligroso por hora (o, lo que es lo mismo, aproximar a un valor máximo el tiempo medio hasta fallo peligroso del componente en cuestión). Cuando se pretende emprender tal acción una vez definidas las características del dispositivo de seguridad, se observa que no es posible ya que, al minimizar la contribución de probabilidad de fallo peligroso por hora de un componente, en el elemento que resta tiende a aumentar en la misma cantidad con tal de compensar su efecto y asegurar el requerimiento de seguridad. Por consiguiente, el uso del programa informático no es muy adecuado en aquellos canales con un solo componente.

- Canal 2. La repartición del valor límite de probabilidad de fallo peligroso por hora entre los diferentes componentes del canal es efectuada conforme a lo ilustrado en la Tabla 6.6.

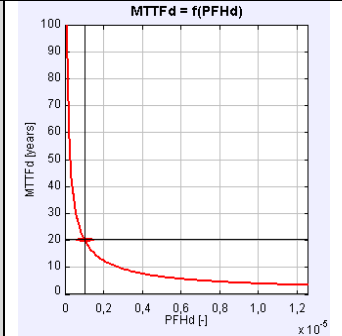
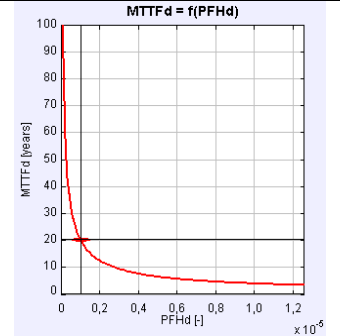
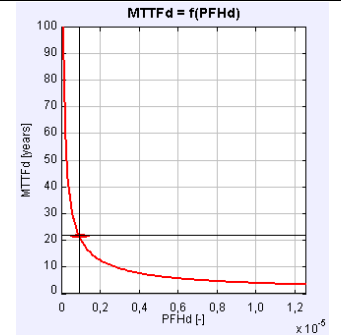
	Entada	Lógico	Salida
<i>PL</i>	c	c	d
<i>PFH_d</i>	$1,01 \cdot 10^{-6}$	$1,01 \cdot 10^{-6}$	$9,3 \cdot 10^{-7}$
<i>MTTF_d</i>	20 años	20 años	22 años
<i>T_{10d}</i>	-	-	-
Gráfico			

Tabla 6.6. Solución analítica del canal 2.

Contrariamente a como ocurre en el análisis del canal anterior, el tiempo medio hasta fallo peligroso de los componentes de seguridad integrados es alcanzable con un buen grado de precisión, experimentando una leve variación en uno de sus elementos.

6.4. Conclusiones.

La lectura de los resultados obtenidos de los ejemplos anteriores permite extraer una serie de conclusiones por lo que respecta al uso y al funcionamiento de la herramienta informática programada. En general, puede apreciarse que los valores

obtenidos son más estrictos que los evaluados de manera teórica, es decir, se hallan probabilidades de fallo peligroso por hora superiores a los valores de referencia y tiempos medio hasta fallo peligroso menores. Las fuentes de error son varias.

En primer lugar, hay que indicar las propias limitaciones del modelo, reflejadas en los gráficos que el usuario del programa manipula. Si bien en dichas gráficas el tiempo medio hasta fallo peligroso está acotado entre 0 y 100 años (y, por ende, la probabilidad de fallo peligroso por hora está delimitada en este intervalo), los componentes de seguridad que se estudian pueden superar el valor máximo indicado y alcanzar valores de en torno a varios miles de años, en particular, los dispositivos de naturaleza eléctrica o electrónica. No obstante, la normativa EN ISO 13849-1 fija un máximo de 100 años con tal de limitar la dependencia de las partes relativas a la seguridad en la fiabilidad y poder abarcar así otros aspectos de su arquitectura.

Por otro lado, el programa considera casos extremos a la hora de asegurar el cumplimiento del nivel de fiabilidad requerido de un sistema. Para un nivel de fiabilidad determinado, existe un amplio espectro de valores de probabilidad de fallo peligroso por hora. De este intervalo, la herramienta informática asume el máximo valor posible, el cual se corresponde con el peor de los casos dados para una categoría y cobertura de diagnóstico dadas. Sin embargo, el sistema evaluado puede no corresponderse a este caso extremo y tener en realidad una probabilidad mucho menor, aunque igualmente dentro del rango de valores admisibles. De este modo, al partir de una probabilidad de fallo peligroso por hora mayor, la contribución de los componentes de seguridad en el canal será igualmente superior a la que se esperaría de no asumirse una situación límite, ya que deben suplirse de manera necesaria la exigencia de seguridad impuesta.

Finalmente, la cobertura de diagnóstico está limitada a una opción en aquellos casos donde existen varios niveles que pueden definirse; este fenómeno es propio de los sistemas de categoría 2 ó 3. Por consiguiente, existe una serie de situaciones que no son contempladas por el programa y que pueden inducir a un error o a variaciones en los resultados mostrados por la herramienta, al modelizarse el sistema con unas características que no se corresponden con las reales. Como se ha descrito en el Capítulo 2, la cobertura de diagnóstico es un parámetro dependiente del componente de seguridad. Así, el hecho de predefinir una cobertura de diagnóstico baja en lugar de una media es equivalente a limitar las aptitudes de los dispositivos empleados en el sistema de control evaluado.

En conclusión, el análisis de la seguridad del sistema es llevado a cabo al nivel del propio sistema y al de sus componentes y optimiza, en cualquier caso, los parámetros relativos a la seguridad de ambos. Los valores de las características resultantes del estudio sirven, pues, de referencia para una posterior selección de los elementos del sistema de control, los instrumentos necesarios para la cual se encuentran directamente incorporados en el programa.

7

Conclusiones

Con este Trabajo de Fin de Máster se ha desarrollado una herramienta informática interactiva para el diseño de la seguridad de sistemas industriales, basándose para ello en la normativa EN ISO 13849-1. Este programa puede emplearse directamente a situaciones reales, como mostrado en los diversos ejemplos contenidos este informe, pero también abre una ventana a la integración de nuevas funcionalidades y a su interacción con otras herramientas del mismo campo de aplicación. En este capítulo se introducen algunos casos.

7.1. Resumen y consideraciones.

La conciencia por un entorno de trabajo seguro, junto con los recientes y abruptos avances tecnológicos, ha conducido a la emisión de una legislación particular orientada a la seguridad en máquinas con el fin de regular su implantación y su uso. En la actualidad, los estándares encargados de regular la seguridad en máquinas son las normas EN ISO 13849-1 e IEC 62061, siendo la primera de las dos apta para cualquier tipo de dispositivo mientras que la segunda está limitada a componentes de naturaleza eléctrica o electrónica. Diversos programas informáticos han plasmado las directrices descritas en los estándares mencionadas. En ellos, los usuarios introducen los componentes de seguridad implementados en su sistema y la herramienta determina automáticamente su nivel de fiabilidad o de integridad, según la situación, y lo compara con su nivel requerido.

El programa desarrollado con la aplicación de EjsS sigue el funcionamiento opuesto al descrito por las herramientas mencionadas, apoyándose en el estándar EN ISO 13849-1. En efecto, el usuario no parte de un conjunto de componentes de seguridad, de los que se comprueba su validez y conformidad con respecto a la norma, sino que éstos constituyen sendas salidas del proceso de diseño. En una primera etapa, el usuario concibe un sistema caracterizado por un nivel de fiabilidad requerido. A partir de dicha variable alfabética se deduce la situación más desfavorable para el sistema planteado, la cual está caracterizada por un valor de probabilidad de fallo peligroso por

hora determinado. Mediante la manipulación de distintas gráficas, el usuario puede definir la manera en que los dispositivos de seguridad de un canal del sistema asimilan los requisitos de seguridad impuestos, definir las características generales de éstos y extraer en consecuencia un listado con los elementos permitiendo alcanzar tales propiedades. La nueva herramienta logra así optimizar la solución de seguridad con la selección que realiza, contrariamente a como ocurre con los programas expuestos anteriormente.

Por consiguiente, los resultados obtenidos al utilizar el programa desarrollado resultan evidentemente más estrictos que los evaluados de manera teórica, como se ha comprobado con los ejemplos ilustrados en el Capítulo 6. Este fenómeno se debe a la existencia de un amplio rango de valores de probabilidad de fallo peligroso por hora para un único nivel de fiabilidad determinado. Al basar la selección de los componentes en una situación extrema, las características de los mismos serán más exigentes que en cualquier otro caso para poder atender el requisito mínimo de seguridad, traduciéndose en probabilidades de fallo peligroso por hora bajas y tiempos medio hasta fallo peligroso elevados. No obstante, cabe destacar que la solución ofrecida por el programa no es más que una propuesta y que es el propio usuario quien, basándose en su experiencia y buen criterio, debe considerar los resultados y aplicarlos de la manera que crea mejor conveniente.

7.2. Oportunidades de mejora y desarrollo.

La herramienta informática desarrollada, como cualquier programa informático, no es un producto completamente terminado, sino que puede considerarse un punto de partida sobre el que incorporar nuevas funcionalidades y proponer mejoras para su materialización en futuras versiones. A continuación, se presentan algunas de las posibles direcciones que el desarrollo de las siguientes versiones puede seguir.

- Compatibilidad con otros programas. Como se ha comentado con anterioridad, el programa presentado en esta memoria sigue un proceso de diseño contrario al de los softwares convencionales de su mismo campo de aplicación. De este modo, una vía de desarrollo abierta es la de su empleo como una herramienta complementaria de la que extraer soluciones de diseño que puedan ser posteriormente testadas en programas informáticos más sofisticados como, por ejemplo, SISTEMA. Con este propósito, es conveniente que las bases de datos sobre las que se sustentan ambos entornos sean compatibles tanto en su formato como en la estructura y forma de su contenido. En la actualidad, son muchos los fabricantes de dispositivos de seguridad que ponen a disposición de los interesados archivos conteniendo bibliotecas con sus gamas de productos, de la que los programas mencionados ya se benefician.

- Admisión de criterios alternativos. El programa considera actualmente sólo las indicaciones del estándar EN ISO 13849-1, al ser éste de aplicación relativamente sencilla y de carácter más versátil que el resto de normativas de su campo de estudio. Sin embargo, la norma IEC 62061, aunque de rango de aplicación más estrecho, puede resultar en determinadas situaciones más sugerente de utilizar según la estructura del sistema evaluado. Por este motivo, se propone para futuras versiones del programa la implementación de las reglas definidas por este segundo estándar para que el usuario escoja el criterio que estime más conveniente para su uso.
- Compatibilidad entre componentes. Los dispositivos hallados de la selección llevada a cabo por el software elaborado pueden no ser compatibles entre ellos y la solución propuesta no realizable. Este problema puede solucionarse, entre otras maneras, incluyendo en la lista de dispositivos introducida en el programa un último campo indicando para qué tipo de aplicaciones o funciones de seguridad es óptima su utilización.

Bibliografía

- [1] International Standardisation Organization (ISO). *Safety of machinery – Safety-related parts of control systems – Part 1: General principles for design (ISO 13849-1:2006)* [pdf]. 2ª ed. Ginebra (Suiza): ISO, 1 de noviembre de 2006. 103 p.
- [2] SICK AG. *Guide for Safe Machinery – Six steps for a safe machine* [pdf]. Waldkirch (Alemania): SICK AG, 7 de julio de 2015. Ref.: 8007988.
- [3] HEDBERG, Johan; SÖDERBERG, Andreas; TEGEHALL, Jan. *How to design safe machine control systems – a guideline to EN ISO 13849-1* [pdf]. Borås (Suecia): SP Technical Research Institute of Sweden, 2011. 78 p. ISBN: 978-91-87017-14-8.
- [4] HAUKE, Michael, et al. *BGIA Report 2/2008. Functional safety of machine control – Application of EN ISO 13849-1* [pdf]. Berlín (Alemania): Institut für Arbeitsschutz der Deutschen Gesetzlichen Unfallversicherung, junio de 2009. ISBN: 978-3-88383-793-2. ISSN: 1869-3491.
- [5] Schneider Electric Industries SAS. *Preventa solutions for efficient machine safety – Safety Legislation and Standards* [pdf]. Rueil-Malmaison (Francia): Schneider Electric Industries SAS, enero de 2015. 26 p. DIA3ED2150103EN.
- [6] GUTIÉRREZ, B. *Seguridad en maquinaria EN ISO 13849-1* [pdf]. Seis Maquinaria S.L.L. 67 p.
- [7] Rockwell Automation, Inc. *Introduction to Functional Safety of Control Systems*. En: Principles, standards and implementation [página web]. Rockwell Automation, Inc., 2016 [Consulta el 15 de marzo de 2016]. Disponible en: <http://www.ab.com/en/epub/catalogs/3377539/5866177/3378076/7555769/Introduction.html>
- [8] Rockwell Automation, Inc. *Diseño del sistema según ISO / EN 13849 y SISTEMA*. En: Principios, estándares e implementación [página web]. Rockwell Automation, Inc., 2016 [Consulta el 15 de marzo de 2016]. Disponible en: <http://www.ab.com/en/epub/catalogs/3377539/5866177/3378076/7555769/Sistema.html>

com/es/epub/catalogs/3377539/5866177/3378076/7565826/Dise-o-del-subsistema-y-combinaciones.html

- [9] Rockwell Automation, Inc. *Medidas de protección y equipo complementario* [página web]. Rockwell Automation, Inc., 2016 [Consulta el 15 de marzo de 2016]. Disponible en: <http://www.ab.com/es/epub/catalogs/3377539/5866177/3378076/7131359/print.html>
- [10] ABB AB. *ABB Safety Handbook. Machine Safety – Jokab Safety products* [pdf]. Suecia: ABB AB, 2013. 411 p. 2TLC172001C0202.
- [11] ReeR. *Safety Related Parts of Control Systems, Part 1: General principles for design* [página web]. En: ISO 13849-1 PL. Turin (Italia): ReeR, 2016. Disponible en: <http://www.reer.it/reer/en/safety-guide/item/262-iso-13849-1-pl-en>
- [12] SICK, AG. *Product Catalog Industrial Safety Systems. Safety solutions and services for machines and systems* [pdf]. Waldkirch (Alemania): SICK AG, 30 de julio de 2010. 1054 p. Ref: 8010889.
- [13] BADIOLA, Fco. Javier; UNZUETA, Ibon; Centro Nacional de Verificación de Maquinaria. *Máquinas: diseño de las partes de los sistemas de mando relativas a la seguridad* [pdf]. En: Notas Técnicas de Prevención 946. Madrid (España): Instituto Nacional de Seguridad e Higiene en el Trabajo (INSHT), 2012. NIPO: 272-13-015-4.

Programas de cálculo

- [1] ESQUEMBRE, Francisco. *Easy Java Simulations* (EjsS), version 5.2, build 151201. Murcia (España): Universidad de Murcia, 2016.
- [2] Microsoft Corporation. *MS Excel 2016*. Estados Unidos: Microsoft Corporation, 2016.

Lista de símbolos, abreviaciones y siglas

a, b, c, d, e	Denotación de los niveles de fiabilidad.
B, 1, 2, 3, 4	Denotación de las categorías.
B10d	Número de ciclos hasta el fallo peligroso del 10% de los componentes.
c	Medio de monitorización cruzada.
Cat.	Categoría.
CCF	Fallo de causa común.
CEN	Comité Europeo de Normalización.
DC	Cobertura de diagnóstico.
DCavg	Cobertura de Diagnóstica media.
dop	Número medio de días de operación por año.
EPI	Equipo de Protección Individual.
F, F1, F2	Frecuencia y/o duración de la exposición a un peligro.
hop	Número medio de horas de operación por día.
I, I1, I2	Dispositivo de entrada.
i, j	Índices de cuenta.
Iab, Ibc, im	Medio de interconexión.
IEC	International Electrotechnical Commission.
ISO	International Standardization Organization.
K	Contactador.
L, L1, L2	Dispositivo lógico.
m	Medio de monitorización.
MTTFd	Tiempo medio hasta fallo peligroso.
n	Número de elementos.
nop	Tiempo medio de operaciones anuales.

O, O1, O2	Dispositivo de salida.
OTE	Salida del equipo de verificación.
P, P1, P2	Probabilidad de evitar o reducir un peligro.
PFHd	Probabilidad de fallo peligroso por hora.
PL	Nivel de fiabilidad o nivel de prestación.
PLr	Nivel de fiabilidad o nivel de prestación requerido.
S, S1, S2	Severidad de un daño.
SIL	Nivel de integridad de un sistema.
SRP	Parte relativa a la seguridad.
SRP/CS	Parte relativa a la seguridad de un sistema de control.
tcycle	Tiempo medio entre el comienzo de dos ciclos consecutivos.
T	Ciclo de vida de un componente.
T10d	Tiempo medio hasta el fallo peligroso del 10% de los componentes.
TE	Equipo de verificación.

Anexo A. Tablas

A1. Cobertura de diagnóstico (DC).

Las Tablas A1.1., A1.2. y A1.3. representan distintos valores de cobertura de diagnóstico alcanzadas por una parte relativa a la seguridad de un sistema de control, en base a las medidas implementadas y según la naturaleza del dispositivo o la unidad afectada (entrada, lógico o de salida, respectivamente). El valor de la cobertura de diagnóstico para las medidas con indicación (*) es dependiente de la aplicación.

Medida	DC
Estímulo cíclico de comprobación mediante cambio dinámico de las señales de entrada.	90%
Prueba de verosimilitud. Por ejemplo, utilización de contactos normalmente abiertos y normalmente cerrados, guiados mecánicamente.	99%
Control cruzado de señales de entrada sin comprobación dinámica.	0% a 99%*
Control cruzado de señales de entrada con comprobación dinámica, en el caso de que los cortocircuitos no sean detectables (para I/O múltiples).	90%
Control cruzado de las señales de entrada y de los resultantes intermedios en la lógica (L) y el control temporal y lógico del soporte lógico y del flujo del programa y detección de defectos estáticos y de cortocircuitos (para I/O múltiples).	99%
Control indirecto. Por ejemplo, control mediante presostato, control de la posición eléctrica de los accionadores.	90% a 99%*
Control directo. Por ejemplo, control directo de la posición eléctrica de los distribuidores de mando, control de los dispositivos electromecánicos mediante elementos de contacto unidos mecánicamente.	99%
Detección de fallos mediante el proceso.	0% a 99%*
Control de algunas características del sensor. Por ejemplo, tiempo de respuesta, gama de las señales analógicas (resistencia eléctrica o capacidad).	60%

Tabla A1.1. Estimación de la cobertura de diagnóstico para elementos de entrada.

Medida	DC
Control indirecto. Por ejemplo, control mediante el presostato, control de la posición eléctrica de los accionadores.	90% a 99%*
Control directo. Por ejemplo, control directo de la posición eléctrica de los distribuidores de mando, control de los dispositivos electromecánicos mediante elementos de contacto unidos mecánicamente.	99%
Control temporal simple de la lógica. Por ejemplo, temporizador como perro guardián, cuando los puntos de disparo se encuentran en el programa de la lógica.	60%
Control temporal y lógico de la lógica mediante el perro guardián, cuando el equipo de comprobación realiza pruebas de verosimilitud del comportamiento de la lógica.	90%
Autocontroles a la puesta en marcha para detectar los defectos latentes en partes de la lógica. Por ejemplo, memorias de programa y de datos, puertos de entrada/salida, interfaces.	90%
Comprobación de la capacidad de reacción del dispositivo de control (por ejemplo, perro guardián) por el canal principal a la puesta en marcha, o cuando se solicita la función de seguridad o una señal externa la solicita a través de un dispositivo de entrada.	90%
Principio dinámico: todos los componentes de la lógica deben cambiar de estado MARCHA-PARADA-MARCHA cuando se solicita la función de seguridad. Por ejemplo, circuito de enclavamiento realizado mediante relés.	99%
Memoria invariable: firma de una palabra (8bit).	90%
Memoria invariable: firma de dos palabras (16bits).	99%
Memoria variable: comprobación de la RAM utilizando datos redundantes. Por ejemplo, banderas, marcadores, constantes, temporizadores y comparación mutua de dichos datos.	60%
Memoria variable: comprobación de la legibilidad y aptitud para la lectura de las células de memoria utilizadas.	60%
Memoria variable: control de la RAM con un código de Hamming modificado o autocontrol de la RAM.	99%
Tratamiento por unidad: autocontrol mediante soporte lógico.	60% a 99%*
Tratamiento por unidad: tratamiento codificado.	90% a 99%*
Detección de defectos mediante el proceso.	0% a 99%*

Tabla A1.2. Estimación de la cobertura de diagnóstico para elementos lógicos.

Medida	DC
Control de las salidas mediante un canal sin comprobación dinámica.	0% a 99%*
Control cruzado de las señales de salida sin comprobación dinámica.	0% a 99%*
Control cruzado de las señales de salida con comprobación dinámica, sin detección de cortocircuitos (para I/O múltiples).	90%
Control cruzado de las señales de entrada y de los resultantes intermedios en la lógica (L) y el control temporal y lógico del soporte lógico y del flujo del programa y detección de defectos estáticos y de cortocircuitos (para I/O múltiples).	99%
Vía de parada redundante sin control de accionados.	0%
Vía de parada redundante con control de uno de los accionadores, mediante lógica o un equipo de comprobación.	90%
Vía de parada redundante con control de uno de los accionadores, mediante lógica y un equipo de comprobación.	99%
Control indirecto. Por ejemplo, control mediante un presostato, control de la posición eléctrica de los accionadores.	90% a 99%*
Detección de defectos mediante el proceso.	0% a 99%*
Control directo. Por ejemplo, control de la posición eléctrica de los distribuidores de mando, control de los dispositivos electromecánicos mediante elementos de contacto unidos mecánicamente.	99%

Tabla A1.3. Estimación de la cobertura de diagnóstico para elementos de salida.

A2. Probabilidad de fallo peligroso por hora (PFH_d).

La Tabla A2.1. muestra valores de probabilidad de fallo peligroso por hora en función del tiempo medio hasta fallo peligroso de cada canal en un sistema de control. Dicho valor viene además determinado por la arquitectura y la cobertura del canal.

MTTF _d [years]	PFH _d [-]						
	Cat. B	Cat. 1	Cat. 2	Cat. 2	Cat. 3	Cat. 3	Cat. 4
	DC none	DC none	DC low	DC medium	DC low	DC medium	DC high
3	3,80E-05		2,58E-05	1,99E-05	1,26E-05	6,09E-06	
3,3	3,46E-05		2,33E-05	1,79E-05	1,13E-05	5,41E-06	
3,6	3,17E-05		2,13E-05	1,62E-05	1,03E-05	4,86E-06	
3,9	2,93E-05		1,95E-05	1,48E-05	9,37E-06	4,40E-06	
4,3	2,65E-05		1,76E-05	1,33E-05	8,39E-06	3,89E-06	
4,7	2,43E-05		1,60E-05	1,20E-05	7,58E-06	3,48E-06	
5,1	2,24E-05		1,47E-05	1,10E-05	6,91E-06	3,15E-06	
5,6	2,04E-05		1,33E-05	9,87E-06	6,21E-06	2,80E-06	
6,2	1,84E-05		1,19E-05	8,80E-06	5,53E-06	2,47E-06	
6,8	1,68E-05		1,08E-05	7,93E-06	4,98E-06	2,20E-06	
7,5	1,52E-05		9,75E-06	7,10E-06	4,45E-06	1,95E-06	
8,2	1,39E-05		8,87E-06	6,43E-06	4,02E-06	1,74E-06	
9,1	1,25E-05		7,94E-06	5,71E-06	3,57E-06	1,53E-06	
10	1,14E-05		7,18E-06	5,14E-06	3,21E-06	1,36E-06	
11	1,04E-05		6,44E-06	4,53E-06	2,81E-06	1,18E-06	
12	9,51E-06		5,84E-06	4,04E-06	2,49E-06	1,04E-06	
13	8,78E-06		5,33E-06	3,64E-06	2,23E-06	9,21E-07	
15	7,61E-06		4,53E-06	3,01E-06	1,86E-06	7,44E-07	
16	7,13E-06		4,21E-06	2,77E-06	1,67E-06	6,67E-07	
18	6,34E-06		3,68E-06	2,37E-06	1,41E-06	5,67E-07	
20	5,71E-06		3,26E-06	2,06E-06	1,22E-06	4,85E-07	
22	5,19E-06		2,93E-06	1,82E-06	1,07E-06	4,21E-07	
24	4,76E-06		2,65E-06	1,62E-06	9,47E-07	3,70E-07	
27	4,23E-06		2,32E-06	1,39E-06	8,04E-07	3,10E-07	
30		3,80E-06	2,06E-06	1,21E-06	6,94E-07	2,65E-07	9,54E-08
33		3,46E-06	1,85E-06	1,06E-06	5,94E-07	2,30E-07	8,57E-08
36		3,17E-06	1,67E-06	9,39E-07	5,16E-07	2,01E-07	7,77E-08
39		2,93E-06	1,53E-06	8,40E-07	4,53E-07	1,78E-07	7,11E-08
43		2,65E-06	1,37E-06	7,34E-07	3,87E-07	1,54E-07	6,37E-08
47		2,43E-06	1,24E-06	6,49E-07	3,35E-07	1,34E-07	5,76E-08
51		2,24E-06	1,13E-06	5,80E-07	2,93E-07	1,19E-07	5,26E-08
56		2,04E-06	1,02E-06	5,10E-07	2,52E-07	1,03E-07	4,73E-08
62		1,84E-06	9,06E-07	4,43E-07	2,13E-07	8,84E-08	4,22E-08
68		1,68E-06	8,17E-07	3,90E-07	1,84E-07	7,68E-08	3,80E-08
75		1,52E-06	7,31E-07	3,40E-07	1,57E-07	6,62E-08	3,41E-08
82		1,39E-06	6,61E-07	3,04E-07	1,35E-07	5,79E-08	3,08E-08
91		1,25E-06	5,88E-07	2,61E-07	1,14E-07	4,94E-08	2,74E-08
100		1,14E-06	5,28E-07	2,29E-07	1,01E-07	4,29E-08	2,47E-08

Tabla A2.1. Valores de probabilidad de fallo peligroso por hora admisibles en función del tiempo medio hasta fallo peligroso.

Anexo B. Código del programa

B1. Botón “Máquina”.

Nombre	Valor inicial	Tipo	Dimensión
machine_input	-	String	-
machine_v	0	int	-
machine_n	-	String	-

Tabla B1.1. Parámetros del botón “Máquina”.

```
//Add machine button
public void addMachine () {
    if (AddMachineButton == true) {
        machine_v = machine_v + 1;
        if (machine_v == 1) {
            machine_n = machine_input;
            _view.Operations.append("> Machine '" + machine_n + "' was
created." + "\n");
            _view.System_structure.setText("M. " + machine_n);
        }
        else if (machine_v > 1) {
            _view.Operations.append("> Error: a machine already exists!" +
"\n");
            machine_v = 1;
        }
    }
    showMachineDialog = false;
}
```

B2. Botón “Riesgo”.

Nombre	Valor inicial	Tipo	Dimensión
risk_input	-	String	-
risk_v	0	int	[100]
risk_n	“o”	String	[100]
i	0	int	-

risk_select	“---;”	String	-
-------------	--------	--------	---

Tabla B2.1. Parámetros del botón “Riesgo”.

```

// Add RISK button.
public void addRisk () {
    if (AddRiskButton == true) {
        if (machine_v == 1) {
            risk_n[i] = risk_input;
            risk_v[i] = i + 1;
            _view.Operations.append("> Risk " + risk_v[i] + ": '" +
risk_n[i] + "' was created." + "\n");
            _view.System_structure.setText("M. " + machine_n);
            for (r = 0; r < i + 1; r++) {
                sf_count = 0;
                _view.System_structure.append("\n" + "    R" + risk_v[r] + ".
" + risk_n[r]);
                for (sf = 0; sf < 11; sf++) {
                    if (!secfun_n[r][sf].equals("o")) {
                        sf_count = sf_count + 1;
                        _view.System_structure.append("\n" + "        SF" +
String.valueOf(r + 1) + "." + String.valueOf(sf_count) + ". " +
secfun_n[r][sf]);
                        if (!secllev_n[r][sf][0].equals("o")) {
                            _view.System_structure.append("\n" + "            SL. PL " +
secllev_n[r][sf][0]);
                        }
                        else {
                            sf = sf + 0;
                        }
                    }
                }
                else {
                    sf = sf + 0;
                }
            }
            risk_select = risk_select + risk_n[i] + ";";
            i = i + 1;
        }
        else {
            _view.Operations.append("> Error: create a machine first!" +
"\n");
        }
    }
    showRiskDialog = false;
}

```


B3. Botón “Función de seguridad”.

Nombre	Valor inicial	Tipo	Dimensión
secfun_R	-	String	-
secfun_input	“---“	String	-
secfun_n	“o”	String	[100] [12]
j	0	int	-
k	0	int	-
secfun_option	“---;”	String	[100]
risk_select	“---”	String	-
r	0	int	-

Tabla B3.1. Parámetros del botón “Función de seguridad”.

```
// Add SAFETY FUNCTION button.
public void addSecFun () {
    if (AddSecFunButton == true) {
        if (machine_v == 1) {
            if (i > 0) {
                for (k = 0; k < 100; k++) {
                    if (risk_n[k].equals(secfun_R)) {
// error
                        if (secfun_input.equals("---")) {
                            _view.Operations.append("> Error: select a safety
function!" + "\n");
                        }
// j = 0
                        else if (secfun_input.equals("Permanently preventing
access")) {
                            if (secfun_n[k][0].equals("o")) {
                                secfun_n[k][0] = secfun_input;
                                _view.Operations.append("> Safety Function: '" +
secfun_n[k][0] + "' was created for risk '" + secfun_R + "'." + "\n");
                                if (secfun_option[k] == "") {
                                    secfun_option[k] = "---;" + secfun_n[k][0] + ";";
                                }
                                else {
                                    secfun_option[k] = secfun_option[k] + secfun_n[k][0]
+ ";";
                                }
                            }
                        }
                        else {
                            _view.Operations.append("> Error: Safety function
already exists!" + "\n");
                        }
                    }
                }
// j = 1
                else if (secfun_input.equals("Temporarily preventing
```

```

access")) {
    if (secfun_n[k][0].equals("o")) {
        secfun_n[k][1] = secfun_input;
        _view.Operations.append("> Safety Function: '" +
secfun_n[k][1] + "' was created for risk '" + secfun_R + "'. " + "\n");
        if (secfun_option[k] == "") {
            secfun_option[k] = "---;" + secfun_n[k][1] + ";";
        }
        else {
            secfun_option[k] = secfun_option[k] + secfun_n[k][1]
+ ";";
        }
    }
    else {
        _view.Operations.append("> Error: Safety function
already exists!" + "\n");
    }
}
// j = 2
    else if (secfun_input.equals("Retaining parts / substances
/ radiation")) {
        if (secfun_n[k][0].equals("o")) {
            secfun_n[k][2] = secfun_input;
            _view.Operations.append("> Safety Function: '" +
secfun_n[k][2] + "' was created for risk '" + secfun_R + "'. " + "\n");
            if (secfun_option[k] == "") {
                secfun_option[k] = "---;" + secfun_n[k][2] + ";";
            }
            else {
                secfun_option[k] = secfun_option[k] + secfun_n[k][2]
+ ";";
            }
        }
        else {
            _view.Operations.append("> Error: Safety function
already exists!" + "\n");
        }
    }
// j = 3
    else if (secfun_input.equals("Initiating a stop")) {
        if (secfun_n[k][0].equals("o")) {
            secfun_n[k][3] = secfun_input;
            _view.Operations.append("> Safety Function: '" +
secfun_n[k][3] + "' was created for risk '" + secfun_R + "'. " + "\n");
            if (secfun_option[k] == "") {
                secfun_option[k] = "---;" + secfun_n[k][3] + ";";
            }
            else {
                secfun_option[k] = secfun_option[k] + secfun_n[k][3]
+ ";";
            }
        }
        else {

```

```
        _view.Operations.append("> Error: Safety function
already exists!" + "\n");
    }
}
// j = 4
    else if (secfun_input.equals("Avoiding an unepcted
startup")) {
        if (secfun_n[k][0].equals("o")) {
            secfun_n[k][4] = secfun_input;
            _view.Operations.append("> Safety Function: '" +
secfun_n[k][4] + "' was created for risk '" + secfun_R + "'." + "\n");
            if (secfun_option[k] == "") {
                secfun_option[k] = "---;" + secfun_n[k][4] + ";";
            }
            else {
                secfun_option[k] = secfun_option[k] + secfun_n[k][4]
+ ";";
            }
        }
        else {
            _view.Operations.append("> Error: Safety function
already exists!" + "\n");
        }
    }
// j = 5
    else if (secfun_input.equals("Preventing start")) {
        if (secfun_n[k][0].equals("o")) {
            secfun_n[k][5] = secfun_input;
            _view.Operations.append("> Safety Function: '" +
secfun_n[k][5] + "' was created for risk '" + secfun_R + "'." + "\n");
            if (secfun_option[k] == "") {
                secfun_option[k] = "---;" + secfun_n[k][5] + ";";
            }
            else {
                secfun_option[k] = secfun_option[k] + secfun_n[k][5]
+ ";";
            }
        }
        else {
            _view.Operations.append("> Error: Safety function
already exists!" + "\n");
        }
    }
// j = 6
    else if (secfun_input.equals("Combination of initiating a
stop / preventing start")) {
        if (secfun_n[k][0].equals("o")) {
            secfun_n[k][6] = secfun_input;
            _view.Operations.append("> Safety Function: '" +
secfun_n[k][6] + "' was created for risk '" + secfun_R + "'." + "\n");
            if (secfun_option[k] == "") {
                secfun_option[k] = "---;" + secfun_n[k][6] + ";";
            }
        }
    }
}
```

```

        else {
            secfun_option[k] = secfun_option[k] + secfun_n[k][6]
+ ";";
        }
    }
    else {
        _view.Operations.append("> Error: Safety function
already exists!" + "\n");
    }
}
// j = 7
    else if (secfun_input.equals("Enabling material
throughput")) {
        if (secfun_n[k][0].equals("o")) {
            secfun_n[k][7] = secfun_input;
            _view.Operations.append("> Safety Function: '" +
secfun_n[k][7] + "' was created for risk '" + secfun_R + "'. " + "\n");
            if (secfun_option[k] == "") {
                secfun_option[k] = "---;" + secfun_n[k][7] + ";";
            }
            else {
                secfun_option[k] = secfun_option[k] + secfun_n[k][7]
+ ";";
            }
        }
        else {
            _view.Operations.append("> Error: Safety function
already exists!" + "\n");
        }
    }
// j = 8
    else if (secfun_input.equals("Monitoring machine
parameters")) {
        if (secfun_n[k][0].equals("o")) {
            secfun_n[k][8] = secfun_input;
            _view.Operations.append("> Safety Function: '" +
secfun_n[k][8] + "' was created for risk '" + secfun_R + "'. " + "\n");
            if (secfun_option[k] == "") {
                secfun_option[k] = "---;" + secfun_n[k][8] + ";";
            }
            else {
                secfun_option[k] = secfun_option[k] + secfun_n[k][8]
+ ";";
            }
        }
        else {
            _view.Operations.append("> Error: Safety function
already exists!" + "\n");
        }
    }
// j = 9
    else if (secfun_input.equals("Disabling safety functions
manually and for a limited time")) {

```

```

        if (secfun_n[k][0].equals("o")) {
            secfun_n[k][9] = secfun_input;
            _view.Operations.append("> Safety Function: '" +
secfun_n[k][9] + "' was created for risk '" + secfun_R + "'." + "\n");
            if (secfun_option[k] == "") {
                secfun_option[k] = "---;" + secfun_n[k][9] + ";";
            }
            else {
                secfun_option[k] = secfun_option[k] + secfun_n[k][9]
+ ";";
            }
        }
        else {
            _view.Operations.append("> Error: Safety function
already exists!" + "\n");
        }
    }
    // j = 10
    else if (secfun_input.equals("Emergency stop")) {
        if (secfun_n[k][0].equals("o")) {
            secfun_n[k][10] = secfun_input;
            _view.Operations.append("> Safety Function: '" +
secfun_n[k][10] + "' was created for risk '" + secfun_R + "'." +
"\n");

            if (secfun_option[k] == "") {
                secfun_option[k] = "---;" + secfun_n[k][10] + ";";
            }
            else {
                secfun_option[k] = secfun_option[k] +
secfun_n[k][10] + ";";
            }
        }
        else {
            _view.Operations.append("> Error: Safety function
already exists!" + "\n");
        }
    }
    // j = 11
    else if (secfun_input.equals("Safety-relevant indications
and alarms")) {
        if (secfun_n[k][0].equals("o")) {
            secfun_n[k][11] = secfun_input;
            _view.Operations.append("> Safety Function: '" +
secfun_n[k][11] + "' was created for risk '" + secfun_R + "'." +
"\n");

            if (secfun_option[k] == "") {
                secfun_option[k] = "---;" + secfun_n[k][11] + ";";
            }
            else {
                secfun_option[k] = secfun_option[k] +
secfun_n[k][11] + ";";
            }
        }
    }
}

```

```

        else {
            _view.Operations.append("> Error: Safety function
already exists!" + "\n");
        }
    }
    // breaking the loop
    break;
    // closing statements
    }
    else {
        k = k + 0;
    }
}
else {
    _view.Operations.append("> Error: create a risk first!" +
"\n");
}
else {
    _view.Operations.append("> Error: create a machine first!" +
"\n");
    _view.System_structure.setText("M. " + machine_n);
    for (r = 0; r < i; r++) {
        sf_count = 0;
        _view.System_structure.append("\n" + "    R" + risk_v[r] + ". "
+ risk_n[r]);
        for (sf = 0; sf < 11; sf++) {
            if (!secfun_n[r][sf].equals("o")) {
                sf_count = sf_count + 1;
                _view.System_structure.append("\n" + "        SF" +
String.valueOf(r + 1) + "." + String.valueOf(sf_count) + ". " +
secfun_n[r][sf]);
                if (!seclev_n[r][sf][0].equals("o")) {
                    _view.System_structure.append("\n" + "            SL. PL " +
seclev_n[r][sf][0]);
                }
            }
            else {
                sf = sf + 0;
            }
        }
        else {
            sf = sf + 0;
        }
    }
}
}
showSecFunDialog = false;
}

```

B4. Botón “Nivel de fiabilidad”.

Nombre	Valor inicial	Tipo	Dimensión
seclev_R	“---“	String	-
seclev_SF	“---“	String	-
sevISO_input	“---“	String	-
freqISO_input	“---“	String	-
posISO_input	“---“	String	-
seclev_v	0	int	[100] [12] [1]
seclev_n	“o”	String	[100] [12] [1]
m	0	int	-
n	0	int	-
sf	0	int	-
sf_count	0	int	-

Tabla B4.1. Parámetros del botón “Nivel de fiabilidad”.

```

public void addSecLev () {
    if (AddSecLevButton == true) {
        if (machine_v == 1) {
            // Determination of the risk "n" and the safety function "n".
            for (m = 0; m < 100; m++) {
                if (risk_n[m].equals(seclev_R)) {
                    if (seclev_SF.equals("---")){
                        _view.Operations.append("> Error: specify safety
function!" + "\n");
                    }
                    for (n = 0; n < 12; n++) {
                        if (secfun_n[m][n].equals(seclev_SF)) {
                            // Determination of PLr for risk "m" and safety function "n".
                            if (sevISO_input.equals("---")) {
                                _view.Operations.append("> Error: select a severity!"
+ "\n");
                            }
                            else if (freqISO_input.equals("---")){
                                _view.Operations.append("> Error: select a frequency!"
+ "\n");
                            }
                            else if (posISO_input.equals("---")) {
                                _view.Operations.append("> Error: select a
possibility!" + "\n");
                            }
                            else if (sevISO_input.equals("Minor") &&
freqISO_input.equals("Rare") && posISO_input.equals("Avoidable")) {
                                seclev_n[m][n][0] = "a";
                                _view.Operations.append("> Safety level '" +
seclev_n[m][n][0] + "' was assigned to safety function '" +

```

```

secfun_n[m][n] + "' from risk '" + risk_n[m] + "'.'" + "\n");
    }
    else if (sevISO_input.equals("Minor") &&
freqISO_input.equals("Rare") && posISO_input.equals("Unavoidable")) {
        seclev_n[m][n][0] = "b";
        _view.Operations.append("> Safety level '" +
seclev_n[m][n][0] + "' was assigned to safety function '" +
secfun_n[m][n] + "' from risk '" + risk_n[m] + "'.'" + "\n");
    }
    else if (sevISO_input.equals("Minor") &&
freqISO_input.equals("Frequent") && posISO_input.equals("Avoidable"))
{
        seclev_n[m][n][0] = "b";
        _view.Operations.append("> Safety level '" +
seclev_n[m][n][0] + "' was assigned to safety function '" +
secfun_n[m][n] + "' from risk '" + risk_n[m] + "'.'" + "\n");
    }
    else if (sevISO_input.equals("Minor") &&
freqISO_input.equals("Frequent") &&
posISO_input.equals("Unavoidable")) {
        seclev_n[m][n][0] = "c";
        _view.Operations.append("> Safety level '" +
seclev_n[m][n][0] + "' was assigned to safety function '" +
secfun_n[m][n] + "' from risk '" + risk_n[m] + "'.'" + "\n");
    }
    else if (sevISO_input.equals("Serious") &&
freqISO_input.equals("Rare") && posISO_input.equals("Avoidable")) {
        seclev_n[m][n][0] = "c";
        _view.Operations.append("> Safety level '" +
seclev_n[m][n][0] + "' was assigned to safety function '" +
secfun_n[m][n] + "' from risk '" + risk_n[m] + "'.'" + "\n");
    }
    else if (sevISO_input.equals("Serious") &&
freqISO_input.equals("Rare") && posISO_input.equals("Unavoidable")) {
        seclev_n[m][n][0] = "d";
        _view.Operations.append("> Safety level '" +
seclev_n[m][n][0] + "' was assigned to safety function '" +
secfun_n[m][n] + "' from risk '" + risk_n[m] + "'.'" + "\n");
    }
    else if (sevISO_input.equals("Serious") &&
freqISO_input.equals("Frequent") && posISO_input.equals("Avoidable"))
{
        seclev_n[m][n][0] = "d";
        _view.Operations.append("> Safety level '" +
seclev_n[m][n][0] + "' was assigned to safety function '" +
secfun_n[m][n] + "' from risk '" + risk_n[m] + "'.'" + "\n");
    }
    else if (sevISO_input.equals("Serious") &&
freqISO_input.equals("Frequent") &&
posISO_input.equals("Unavoidable")) {
        seclev_n[m][n][0] = "e";
        _view.Operations.append("> Safety level '" +
seclev_n[m][n][0] + "' was assigned to safety function '" +

```



```
secfun_n[m][n] + "' from risk '" + risk_n[m] + "'.'" + "\n");
    }
    break;
}
else {
    n = n + 0;
}
}
break;
}
else {
    m = m + 0;
}
}
}
else {
    _view.Operations.append("> Error: create a machine first!" +
"\n");
}
_view.System_structure.setText("M. " + machine_n);
for (r = 0; r < i; r++) {
    sf_count = 0;
    _view.System_structure.append("\n" + "    R" + risk_v[r] + ". "
+ risk_n[r]);
    for (sf = 0; sf < 11; sf++) {
        if (!secfun_n[r][sf].equals("o")) {
            sf_count = sf_count + 1;
            _view.System_structure.append("\n" + "        SF" +
String.valueOf(r + 1) + "." + String.valueOf(sf_count) + ". " +
secfun_n[r][sf]);
            if (!seclev_n[r][sf][0].equals("o")) {
                _view.System_structure.append("\n" + "            SL. PL " +
seclev_n[r][sf][0]);
            }
            else {
                sf = sf + 0;
            }
        }
        else {
            sf = sf + 0;
        }
    }
}
}
showSecLevDialog = false;
}
```

B5. Botón “Lista de dispositivos”.

Nombre	Valor inicial	Tipo	Dimensión
Device	-	String	[300]
d	0	int	-
ListName	-	String	-
ListContent	-	String	-
Variable	-	String	[10]

Tabla B5.1. Parámetros del botón “Lista de dispositivos”.

```

public void introList () {
    if (SecDevListButton == true) {
        String ListName =
fileChooser.chooseFilename(_view.Main_window, false);
        ListName = ListName.replaceAll("\\\\", "/");
        String ListContent = resourceFinder.getString(ListName);
        ListContent = ListContent.replace("+", "");
        ListContent = ListContent.replace("\n", "");
        Device = ListContent.split(";", 300);
        for (d = 0; d < 300; d++) {
            Variable = Device[d].split(",", 10);
            if (d < 100) {
                DevList_input_n[d][0] = Variable[0];
                DevList_input_n[d][1] = Variable[1];
                DevList_input_n[d][2] = Variable[2];
                DevList_input_n[d][3] = Variable[3];
                DevList_input_v[d][0] = Double.parseDouble(Variable[4]);
                DevList_input_v[d][1] = Double.parseDouble(Variable[5]);
                DevList_input_v[d][2] = Double.parseDouble(Variable[6]);
                DevList_input_v[d][3] = Double.parseDouble(Variable[7]);
                DevList_input_v[d][4] = Double.parseDouble(Variable[8]);
                DevList_input_v[d][5] = Double.parseDouble(Variable[9]);
            }
            else if ((d > 99) && (d < 200)) {
                DevList_logic_n[d-100][0] = Variable[0];
                DevList_logic_n[d-100][1] = Variable[1];
                DevList_logic_n[d-100][2] = Variable[2];
                DevList_logic_n[d-100][3] = Variable[3];
                DevList_logic_v[d-100][0] = Double.parseDouble(Variable[4]);
                DevList_logic_v[d-100][1] = Double.parseDouble(Variable[5]);
                DevList_logic_v[d-100][2] = Double.parseDouble(Variable[6]);
                DevList_logic_v[d-100][3] = Double.parseDouble(Variable[7]);
                DevList_logic_v[d-100][4] = Double.parseDouble(Variable[8]);
                DevList_logic_v[d-100][5] = Double.parseDouble(Variable[9]);
            }
            else if (d > 199) {
                DevList_output_n[d-200][0] = Variable[0];
                DevList_output_n[d-200][1] = Variable[1];
            }
        }
    }
}

```

```

        DevList_output_n[d-200][2] = Variable[2];
        DevList_output_n[d-200][3] = Variable[3];
        DevList_output_v[d-200][0] = Double.parseDouble(Variable[4]);
        DevList_output_v[d-200][1] = Double.parseDouble(Variable[5]);
        DevList_output_v[d-200][2] = Double.parseDouble(Variable[6]);
        DevList_output_v[d-200][3] = Double.parseDouble(Variable[7]);
        DevList_output_v[d-200][4] = Double.parseDouble(Variable[8]);
        DevList_output_v[d-200][5] = Double.parseDouble(Variable[9]);
    }
}
}
_view.Operations.append("> Device list successfully loaded." +
"\n");
}

```

B6. Botón “Optimización”.

Nombre	Valor inicial	Tipo	Dimensión
p	0	int	-
q	0	int	-
PLr	“---“	String	-
PFHr	0	double	-
channel	“---“	String	-
channel_v	-	int	-
category	“---“	String	-
category_input	“---“	String	-
secfun_optim	“---“	String	-
optim_R	“---“	String	-
optim_SF	“---“	String	-
scale_x_max	0.000038	double	-

Tabla B6.1. Parámetros del botón “Optimización”.

```

public void optimize () {
// Determination of PFHr (total).
if (PLr.equals("a")){
    if (category.equals("B")){
        PFHr = 3.8*Math.pow(10,-5);
        PLr_x = 2;
        PLr_y = 19;
        showPLrpoint = true;
    }
else if (category.equals("2")){
        PFHr = 2.58*Math.pow(10,-5);
        PLr_x = 10;
        PLr_y = 18;
}
}
}

```

```
        showPLRpoint = true;
    }
    else if (category.equals("3")){
        PFHr = 1.26*Math.pow(10,-5);
        PLr_x = 18;
        PLr_y = 16.5;
        showPLRpoint = true;
    }
    else {
        showPLRpoint = false;
    }
}
else if (PLr.equals("b")){
    if (category.equals("B")){
        PFHr = 1.14*Math.pow(10,-5);
        PLr_x = 2;
        PLr_y = 16;
        showPLRpoint = true;
    }
    else if (category.equals("1")){
        PFHr = 3.8*Math.pow(10,-6);
        PLr_x = 6;
        PLr_y = 13;
        showPLRpoint = true;
    }
    else if (category.equals("2")){
        PFHr = 7.18*Math.pow(10,-6);
        PLr_x = 10;
        PLr_y = 15;
        showPLRpoint = true;
    }
    else if (category.equals("3")){
        PFHr = 3.21*Math.pow(10,-6);
        PLr_x = 18;
        PLr_y = 16;
        showPLRpoint = true;
    }
    else {
        showPLRpoint = false;
    }
}
else if (PLr.equals("c")){
    if (category.equals("1")){
        PFHr = 3*Math.pow(10,-6);
        PLr_x = 6;
        PLr_y = 12;
        showPLRpoint = true;
    }
    else if (category.equals("2")){
        PFHr = 3*Math.pow(10,-6);
        PLr_x = 10;
        PLr_y = 12;
        showPLRpoint = true;
    }
}
```

```
    }
    else if (category.equals("3")){
        PFHr = 3*Math.pow(10,-6);
        PLr_x = 18;
        PLr_y = 12;
        showPLrpoint = true;
    }
    else {
        showPLrpoint = false;
    }
}
else if (PLr.equals("d")){
    if (category.equals("2")){
        PFHr = 1*Math.pow(10,-6);
        PLr_x = 10;
        PLr_y = 8;
        showPLrpoint = true;
    }
    else if (category.equals("3")){
        PFHr = 1*Math.pow(10,-6);
        PLr_x = 18;
        PLr_y = 8;
        showPLrpoint = true;
    }
    else {
        showPLrpoint = false;
    }
}
else if (PLr.equals("e")){
    if (category.equals("4")){
        PFHr = 9.54*Math.pow(10,-8);
        PLr_x = 26;
        PLr_y = 4;
        showPLrpoint = true;
    }
    else {
        showPLrpoint = false;
    }
}
else {
    _view.Operations.append("> Error: no PLr found!" + "\n");
}
//Determination of channel_v.
if (category.equals("B") || category.equals("1") ||
category.equals("2") || category.equals("3") || category.equals("4")){
    channel_v = 1;
}
else {
    //channel_v = 0;
    _view.Operations.append("> Error: category not selected!" + "\n");
}
// Determination of PFHd (device).
PFHd_input = PFHr/3;
```

```

PFHd_logic = PFHr/3;
PFHd_output = PFHr/3;
PFHd0_input = PFHd_input;
PFHd0_logic = PFHd_logic;
PFHd0_output = PFHd_output;
fix_in = PFHd_input;
fix_lo = PFHd_logic;
fix_out = PFHd_output;
// Determination of PL_input.
if (PFHd_input > 1*Math.pow(10,-5)){
    PL_input = "a";
}
else if (PFHd_input <= 1*Math.pow(10,-5) && PFHd_input >
3*Math.pow(10,-6)){
    PL_input = "b";
}
else if (PFHd_input <= 3*Math.pow(10,-6) && PFHd_input >
1*Math.pow(10,-6)){
    PL_input = "c";
}
else if (PFHd_input <= 1*Math.pow(10,-6) && PFHd_input >
1*Math.pow(10,-7)){
    PL_input = "d";
}
else if (PFHd_input <= 1*Math.pow(10,-7)){
    PL_input = "e";
}
// Determination of MTTFd_input value [years].
if (category.equals("B")) {
    MTTFd_input = 1.14*0.0001*Math.pow(PFHd_input,-1.0002);
    if (MTTFd_input > 100){
        MTTFd_input = 100;
        PFHd_input = 3.80*Math.pow(10,-5);
    }
    PFHd_in_max = 3.80*Math.pow(10,-5);
    PFHd_in_min = 4.23*Math.pow(10,-6);
    B10d_y_in = "0.1*1.14*0.0001*Math.pow(PFHd_input,-1.0002)*nop_in";
    analyticExp_in = "1.14*0.0001*Math.pow(PFHd_in,-1.0002)";
    showMTTFd_in = true;
    scale_x_max = 0.000038;
    _view.Operations.append("> The system was correctly optimized." +
"\n");
}
else if (category.equals("1")) {
    MTTFd_input = 1.16*0.0001*Math.pow(PFHd_input,-0.9985);
    if (MTTFd_input > 100) {
        MTTFd_input = 100;
        PFHd_input = 3.80*Math.pow(10,-6);
    }
    PFHd_in_max = 3.80*Math.pow(10,-6);
    PFHd_in_min = 1.14*Math.pow(10,-6);
    B10d_y_in = "0.1*1.16*0.0001*Math.pow(PFHd_input,-0.9985)*nop_in";
    analyticExp_in = "1.16*0.0001*Math.pow(PFHd_in,-0.9985)";
}

```

```

    showMTTFd_in = true;
    scale_x_max = 0.0000038;
    _view.Operations.append("> The system was correctly optimized." +
"\n");
}
else if (category.equals("2")) {
    MTTFd_input = 0.0002*Math.pow(PFHd_input,-0.8975);
    if (MTTFd_input > 100) {
        MTTFd_input = 100;
        PFHd_input = 2.58*Math.pow(10,-5);
    }
    PFHd_in_max = 2.58*Math.pow(10,-5);
    PFHd_in_min = 5.28*Math.pow(10,-7);
    B10d_y_in = "0.1*0.0002*Math.pow(PFHd_input,-0.8975)*nop_in";
    analyticExp_in = "0.0002*Math.pow(PFHd_in,-0.8975)";
    showMTTFd_in = true;
    scale_x_max = 0.0000258;
    _view.Operations.append("> The system was correctly optimized." +
"\n");
}
else if (category.equals("3")) {
    MTTFd_input = 0.001*Math.pow(PFHd_input,-0.7191);
    if (MTTFd_input > 100) {
        MTTFd_input = 100;
        PFHd_input = 1.26*Math.pow(10,-5);
    }
    PFHd_in_max = 1.26*Math.pow(10,-5);
    PFHd_in_min = 1.01*Math.pow(10,-7);
    B10d_y_in = "0.1*0.001*Math.pow(PFHd_input,-0.7191)*nop_in";
    analyticExp_in = "0.001*Math.pow(PFHd_in,-0.7191)";
    showMTTFd_in = true;
    scale_x_max = 0.0000126;
    _view.Operations.append("> The system was correctly optimized." +
"\n");
}
else if (category.equals("4")) {
    MTTFd_input = 0.85*0.00002*Math.pow(PFHd_input,-0.89);
    if (MTTFd_input > 100){
        MTTFd_input = 100;
        PFHd_input = 9.54*Math.pow(10,-8);
    }
    PFHd_in_max = 9.54*Math.pow(10,-8);
    PFHd_in_min = 2.47*Math.pow(10,-8);
    B10d_y_in = "0.1*0.85*0.00002*Math.pow(PFHd_input,-0.89)*nop_in";
    analyticExp_in = "0.85*0.00002*Math.pow(PFHd_in,-0.89)";
    showMTTFd_in = true;
    scale_x_max = 0.0000000954;
    _view.Operations.append("> The system was correctly optimized." +
"\n");
}
// Determination of the curve B10-nop (input).
if (hyd_input == false){
    nop_input = 7500;
}

```

```

    B10d_input = 0.1*MTTFd_input*nop_input;
    T10d_input = B10d_input/nop_input;
    showB10d_in = true;
}
else if (hyd_input == true){
    nop_input = 0;
    B10d_input = 0;
    T10d_input = 0;
    showB10d_in = false;
}
if (MTTFd_input == 0){
    B10d_input = 0;
    nop_input = 0;
    T10d_input = 0;
    showB10d_in = false;
}

// Determination of PL_logic.
if (PFHd_logic > 1*Math.pow(10,-5)){
    PL_logic = "a";
}
else if (PFHd_logic <= 1*Math.pow(10,-5) && PFHd_logic >
3*Math.pow(10,-6)){
    PL_logic = "b";
}
else if (PFHd_logic <= 3*Math.pow(10,-6) && PFHd_logic >
1*Math.pow(10,-6)){
    PL_logic = "c";
}
else if (PFHd_logic <= 1*Math.pow(10,-6) && PFHd_logic >
1*Math.pow(10,-7)){
    PL_logic = "d";
}
else if (PFHd_logic <= 1*Math.pow(10,-7)){
    PL_logic = "e";
}

// Determination of MTTFd_logic value [years].
if (category.equals("B")) {
    MTTFd_logic = 1.14*0.0001*Math.pow(PFHd_logic,-1.0002);
    if (MTTFd_logic > 100) {
        MTTFd_logic = 100;
        PFHd_logic = 3.80*Math.pow(10,-5);
    }
    PFHd_lo_max = 3.80*Math.pow(10,-5);
    PFHd_lo_min = 4.23*Math.pow(10,-6);
    B10d_y_lo = "0.1*1.14*0.0001*Math.pow(PFHd_logic,-1.0002)*nop_lo";
    analyticExp_lo = "1.14*0.0001*Math.pow(PFHd_lo,-1.0002)";
    showMTTFd_lo = true;
    scale_x_max = 0.000038;
}
else if (category.equals("1")) {
    MTTFd_logic = 1.16*0.0001*Math.pow(PFHd_logic,-0.9985);
    if (MTTFd_logic > 100) {

```



```

    MTTFd_logic = 100;
    PFHd_logic = 3.80*Math.pow(10,-6);
}
PFHd_lo_max = 3.80*Math.pow(10,-6);
PFHd_lo_min = 1.16*Math.pow(10,-6);
B10d_y_lo = "0.1*1.16*0.0001*Math.pow(PFHd_logic,-0.9985)*nop_lo";
analyticExp_lo = "1.16*0.0001*Math.pow(PFHd_lo,-0.9985)";
showMTTFd_lo = true;
scale_x_max = 0.0000038;
}
else if (category.equals("2")) {
    MTTFd_logic = 0.0002*Math.pow(PFHd_logic,-0.8975);
    if (MTTFd_logic > 100) {
        MTTFd_logic = 100;
        PFHd_logic = 2.58*Math.pow(10,-5);
    }
    PFHd_lo_max = 2.58*Math.pow(10,-5);
    PFHd_lo_min = 5.28*Math.pow(10,-7);
    B10d_y_lo = "0.1*0.0002*Math.pow(PFHd_logic,-0.8975)*nop_lo";
    analyticExp_lo = "0.0002*Math.pow(PFHd_lo,-0.8975)";
    showMTTFd_lo = true;
    scale_x_max = 0.0000258;
}
else if (category.equals("3")) {
    MTTFd_logic = 0.001*Math.pow(PFHd_logic,-0.7191);
    if (MTTFd_logic > 100) {
        MTTFd_logic = 100;
        PFHd_logic = 1.26*Math.pow(10,-5);
    }
    PFHd_lo_max = 1.26*Math.pow(10,-5);
    PFHd_lo_min = 1.01*Math.pow(10,-7);
    B10d_y_lo = "0.1*0.001*Math.pow(PFHd_logic,-0.7191)*nop_lo";
    analyticExp_lo = "0.001*Math.pow(PFHd_lo,-0.7191)";
    showMTTFd_lo = true;
    scale_x_max = 0.0000126;
}
else if (category.equals("4")) {
    MTTFd_logic = 0.85*0.00002*Math.pow(PFHd_logic,-0.89);
    if (MTTFd_logic > 100) {
        MTTFd_logic = 100;
        PFHd_logic = 9.54*Math.pow(10,-8);
    }
    PFHd_lo_max = 9.54*Math.pow(10,-8);
    PFHd_lo_min = 2.47*Math.pow(10,-8);
    B10d_y_lo = "0.1*0.85*0.00002*Math.pow(PFHd_logic,-0.89)*nop_lo";
    analyticExp_lo = "0.85*0.00002*Math.pow(PFHd_lo,-0.89)";
    showMTTFd_lo = true;
    scale_x_max = 0.0000000954;
}
// Determination of the curve B10-nop (logic).
if (hyd_logic == false){
    nop_logic = 7500;
    B10d_logic = 0.1*MTTFd_logic*nop_logic;
}

```

```

    T10d_logic = B10d_logic/nop_logic;
    showB10d_lo = true;
}
else if (hyd_logic == true){
    nop_logic = 0;
    B10d_logic = 0;
    T10d_logic = 0;
    showB10d_lo = false;
}
if (MTTFd_logic == 0){
    B10d_logic = 0;
    nop_logic = 0;
    T10d_logic = 0;
    showB10d_lo = false;
}

// Determination of PL_output.
if (PFHd_output > 1*Math.pow(10,-5)){
    PL_output = "a";
}
else if (PFHd_output <= 1*Math.pow(10,-5) && PFHd_output >
3*Math.pow(10,-6)){
    PL_output = "b";
}
else if (PFHd_output <= 3*Math.pow(10,-6) && PFHd_output >
1*Math.pow(10,-6)){
    PL_output = "c";
}
else if (PFHd_output <= 1*Math.pow(10,-6) && PFHd_output >
1*Math.pow(10,-7)){
    PL_output = "d";
}
else if (PFHd_output <= 1*Math.pow(10,-7)){
    PL_output = "e";
}
// Determination of MTTFd_output value [years].
if (category.equals("B")) {
    MTTFd_output = 1.14*0.0001*Math.pow(PFHd_output,-1.0002);
    if (MTTFd_output > 100) {
        MTTFd_output = 100;
        PFHd_output = 3.80*Math.pow(10,-5);
    }
    PFHd_out_max = 3.80*Math.pow(10,-5);
    PFHd_out_min = 4.23*Math.pow(10,-6);
    B10d_y_out = "0.1*1.14*0.0001*Math.pow(PFHd_output,-
1.0002)*nop_out";
    analyticExp_out = "1.14*0.0001*Math.pow(PFHd_out,-1.0002)";
    showMTTFd_out = true;
    scale_x_max = 0.000038;
}
else if (category.equals("1")) {
    MTTFd_output = 1.16*0.0001*Math.pow(PFHd_output,-0.9985);
    if (MTTFd_output > 100) {

```

```

        MTTFd_output = 100;
        PFHd_output = 3.80*Math.pow(10,-6);
    }
    PFHd_out_max = 3.80*Math.pow(10,-6);
    PFHd_out_min = 1.14*Math.pow(10,-6);
    B10d_y_out = "0.1*1.16*0.0001*Math.pow(PFHd_output,-
0.9985)*nop_out";
    analyticExp_out = "1.16*0.0001*Math.pow(PFHd_out,-0.9985)";
    showMTTFd_out = true;
    scale_x_max = 0.0000038;
}
else if (category.equals("2")) {
    MTTFd_output = 0.0002*Math.pow(PFHd_output,-0.8975);
    if (MTTFd_output > 100) {
        MTTFd_output = 100;
        PFHd_output = 2.58*Math.pow(10,-5);
    }
    PFHd_out_max = 2.58*Math.pow(10,-5);
    PFHd_out_min = 5.28*Math.pow(10,-7);
    B10d_y_out = "0.1*0.0002*Math.pow(PFHd_output,-0.8975)*nop_out";
    analyticExp_out = "0.0002*Math.pow(PFHd_out,-0.8975)";
    showMTTFd_out = true;
    scale_x_max = 0.0000258;
}
else if (category.equals("3")) {
    MTTFd_output = 0.001*Math.pow(PFHd_output,-0.7191);
    if (MTTFd_output > 100) {
        MTTFd_output = 100;
        PFHd_output = 1.26*Math.pow(10,-5);
    }
    PFHd_out_max = 1.26*Math.pow(10,-5);
    PFHd_out_min = 1.01*Math.pow(10,-7);
    B10d_y_out = "0.1*0.001*Math.pow(PFHd_output,-0.7191)*nop_out";
    analyticExp_out = "0.001*Math.pow(PFHd_out,-0.7191)";
    showMTTFd_out = true;
    scale_x_max = 0.0000126;
}
else if (category.equals("4")) {
    MTTFd_output = 0.85*0.00002*Math.pow(PFHd_output,-0.89);
    if (MTTFd_output > 100) {
        MTTFd_output = 100;
        PFHd_output = 9.54*Math.pow(10,-8);
    }
    PFHd_out_max = 9.54*Math.pow(10,-8);
    PFHd_out_min = 2.47*Math.pow(10,-8);
    B10d_y_out = "0.1*0.85*0.00002*Math.pow(PFHd_output,-
0.89)*nop_out";
    analyticExp_out = "0.85*0.00002*Math.pow(PFHd_out,-0.89)";
    showMTTFd_out = true;
    scale_x_max = 0.0000000954;
}
// Determination of the curve B10-nop (output).
if (hyd_output == false){

```

```

    nop_output = 7500;
    B10d_output = 0.1*MTTFd_output*nop_output;
    T10d_output = B10d_output/nop_output;
    showB10d_out = true;
}
else if (hyd_output == true){
    nop_output = 0;
    B10d_output = 0;
    T10d_output = 0;
    showB10d_out = false;
}
if (MTTFd_output == 0){
    B10d_output = 0;
    nop_output = 0;
    T10d_output = 0;
    showB10d_out = false;
}
PFHr = PFHd_input + PFHd_logic + PFHd_output;
}

```

B7. Botón “Dispositivos de seguridad”.

Nombre	Valor inicial	Tipo	Dimensión
DevList_input_n	“o”	String	[100] [4]
DevList_logic_n	“o”	String	[100] [4]
DevList_output_n	“o”	String	[100] [4]
DevList_input_v	0	double	[100] [6]
DevList_logic_v	0	double	[100] [6]
DevList_output_v	0	double	[100] [6]
counter_in	-	int	-
counter_lo	-	int	-
counter_out	-	int	-
dev_in	0	int	-
dev_lo	0	int	-
dev_out	0	int	-
inputprint_n	“o”	String	[100] [4]
inputprint_v	0	double	[100] [6]
logicprint_n	“o”	String	[100] [4]
logicprint_v	0	double	[100] [6]
outputprint_n	“o”	String	[100] [4]
outputprint_v	0	double	[100] [6]
copy_in	0	int	-
copy_lo	0	int	-
copy_out	0	int	-
inputcopy_n	“o”	String	[100] [4]

inputcopy_v	0	double	[100] [6]
logiccopy_n	“o”	String	[100] [4]
logiccopy_v	0	double	[100] [6]
outputcopy_n	“o”	String	[100] [4]
outputcopy_v	0	double	[100] [6]

Tabla B7.1. Parámetros del botón “Dispositivos de seguridad”.

```

public void deviceList () {
    if (DevListButton = true) {

// Input device matrix.
    for (copy_in = 0; copy_in < 100; copy_in++) {
        if (!DevList_input_n[copy_in][0].equals("o")) {
            inputcopy_n[copy_in][0] = DevList_input_n[copy_in][0];
            inputcopy_n[copy_in][1] = DevList_input_n[copy_in][1];
            inputcopy_n[copy_in][2] = DevList_input_n[copy_in][2];
            inputcopy_n[copy_in][3] = DevList_input_n[copy_in][3];
            inputcopy_v[copy_in][0] = DevList_input_v[copy_in][0];
            inputcopy_v[copy_in][1] = DevList_input_v[copy_in][1];
            inputcopy_v[copy_in][2] = DevList_input_v[copy_in][2];
            inputcopy_v[copy_in][3] = DevList_input_v[copy_in][3];
            inputcopy_v[copy_in][4] = DevList_input_v[copy_in][4];
            inputcopy_v[copy_in][5] = DevList_input_v[copy_in][5];
            if (DevList_input_v[copy_in][0] == 0) {
                if (category.equals("B")) {
                    inputcopy_v[copy_in][0] =
0.0001*Math.pow(inputcopy_v[copy_in][1],-1.0002);
                }
                else if (category.equals("1")) {
                    inputcopy_v[copy_in][0] =
0.0001*Math.pow(inputcopy_v[copy_in][1],-0.9985);
                }
                else if (category.equals("2")) {
                    inputcopy_v[copy_in][0] =
0.0002*Math.pow(inputcopy_v[copy_in][1],-0.8975);
                }
                else if (category.equals("3")) {
                    inputcopy_v[copy_in][0] =
0.001*Math.pow(inputcopy_v[copy_in][1],-0.7191);
                }
                else if (category.equals("4")) {
                    inputcopy_v[copy_in][0] =
0.00002*Math.pow(inputcopy_v[copy_in][1],-0.89);
                }
            }
        }
    }
    else {
        break;
    }
}
    
```

```

    }
// Logic device matrix.
for (copy_lo = 0; copy_lo < 100; copy_lo++) {
    if (!DevList_logic_n[copy_lo][0].equals("o")) {
        logiccopy_n[copy_lo][0] = DevList_logic_n[copy_lo][0];
        logiccopy_n[copy_lo][1] = DevList_logic_n[copy_lo][1];
        logiccopy_n[copy_lo][2] = DevList_logic_n[copy_lo][2];
        logiccopy_n[copy_lo][3] = DevList_logic_n[copy_lo][3];
        logiccopy_v[copy_lo][0] = DevList_logic_v[copy_lo][0];
        logiccopy_v[copy_lo][1] = DevList_logic_v[copy_lo][1];
        logiccopy_v[copy_lo][2] = DevList_logic_v[copy_lo][2];
        logiccopy_v[copy_lo][3] = DevList_logic_v[copy_lo][3];
        logiccopy_v[copy_lo][4] = DevList_logic_v[copy_lo][4];
        logiccopy_v[copy_lo][5] = DevList_logic_v[copy_lo][5];
        if (DevList_logic_v[copy_lo][0] == 0) {
            if (category.equals("B")) {
                logiccopy_v[copy_lo][0] =
0.0001*Math.pow(logiccopy_v[copy_lo][1],-1.0002);
            }
            else if (category.equals("1")) {
                logiccopy_v[copy_lo][0] =
0.0001*Math.pow(logiccopy_v[copy_lo][1],-0.9985);
            }
            else if (category.equals("2")) {
                logiccopy_v[copy_lo][0] =
0.0002*Math.pow(logiccopy_v[copy_lo][1],-0.8975);
            }
            else if (category.equals("3")) {
                logiccopy_v[copy_lo][0] =
0.001*Math.pow(logiccopy_v[copy_lo][1],-0.7191);
            }
            else if (category.equals("4")) {
                logiccopy_v[copy_lo][0] =
0.00002*Math.pow(logiccopy_v[copy_lo][1],-0.89);
            }
        }
    }
    else {
        break;
    }
}
// Output device matrix.
for (copy_out = 0; copy_out < 100; copy_out++) {
    if (!DevList_output_n[copy_out][0].equals("o")) {
        outputcopy_n[copy_out][0] = DevList_output_n[copy_out][0];
        outputcopy_n[copy_out][1] = DevList_output_n[copy_out][1];
        outputcopy_n[copy_out][2] = DevList_output_n[copy_out][2];
        outputcopy_n[copy_out][3] = DevList_output_n[copy_out][3];
        outputcopy_v[copy_out][0] = DevList_output_v[copy_out][0];
        outputcopy_v[copy_out][1] = DevList_output_v[copy_out][1];
        outputcopy_v[copy_out][2] = DevList_output_v[copy_out][2];
        outputcopy_v[copy_out][3] = DevList_output_v[copy_out][3];
        outputcopy_v[copy_out][4] = DevList_output_v[copy_out][4];
    }
}

```

```
        outputcopy_v[copy_out][5] = DevList_output_v[copy_out][5];
        if (DevList_output_v[copy_out][0] == 0) {
            if (category.equals("B")) {
                outputcopy_v[copy_out][0] =
0.0001*Math.pow(outputcopy_v[copy_out][1],-1.0002);
            }
            else if (category.equals("1")) {
                outputcopy_v[copy_out][0] =
0.0001*Math.pow(outputcopy_v[copy_out][1],-0.9985);
            }
            else if (category.equals("2")) {
                outputcopy_v[copy_out][0] =
0.0002*Math.pow(outputcopy_v[copy_out][1],-0.8975);
            }
            else if (category.equals("3")) {
                outputcopy_v[copy_out][0] =
0.001*Math.pow(outputcopy_v[copy_out][1],-0.7191);
            }
            else if (category.equals("4")) {
                outputcopy_v[copy_out][0] =
0.00002*Math.pow(outputcopy_v[copy_out][1],-0.89);
            }
        }
    }
    else {
        break;
    }
}

// Counters.
    if ((counter_in != 0) || (counter_lo != 0) || (counter_out != 0))
{
    counter_in = 0;
    counter_lo = 0;
    counter_out = 0;
}
    _view.Inputdev_list.setText("");

// Selection.
// Input device.
    for (dev_in = 0; dev_in < 100; dev_in++) {
        if (!DevList_input_n[dev_in][0].equals("o")) {
            if (((inputcopy_n[dev_in][3].equals("Mechanical")) ||
(inputcopy_n[dev_in][3].equals("Pneumatic"))) && (hyd_input == false))
{
                if ((inputcopy_v[dev_in][0] <= PFHd_input) &&
((inputcopy_v[dev_in][2] >= B10d_input) || (inputcopy_v[dev_in][4] >=
T10d_input))) {
                    inputprint_n[counter_in][0] = DevList_input_n[dev_in][0];
                    inputprint_n[counter_in][1] = DevList_input_n[dev_in][1];
                    inputprint_n[counter_in][2] = DevList_input_n[dev_in][2];
                    inputprint_n[counter_in][3] = DevList_input_n[dev_in][3];
                    inputprint_v[counter_in][0] = DevList_input_v[dev_in][0];
```

```

        inputprint_v[counter_in][1] = DevList_input_v[dev_in][1];
        inputprint_v[counter_in][2] = DevList_input_v[dev_in][2];
        inputprint_v[counter_in][3] = DevList_input_v[dev_in][3];
        inputprint_v[counter_in][4] = DevList_input_v[dev_in][4];
        inputprint_v[counter_in][5] = DevList_input_v[dev_in][5];
        counter_in = counter_in + 1;
        _view.Inputdev_list.append(String.valueOf(counter_in) + ".
" + DevList_input_n[dev_in][0] + " | " + DevList_input_n[dev_in][1] +
" | PL " + DevList_input_n[dev_in][2] + " | PFHd " +
String.valueOf(DevList_input_v[dev_in][0]) + " | MTTFd " +
String.valueOf(DevList_input_v[dev_in][1]) + " | Technology " +
DevList_input_n[dev_in][3] + " | B10d " +
String.valueOf(DevList_input_v[dev_in][2]) + " | nop " +
String.valueOf(DevList_input_v[dev_in][3]) + " | T10d " +
String.valueOf(DevList_input_v[dev_in][4]) + " | Price " +
String.valueOf(DevList_input_v[dev_in][5]) + " €" + "\n");
        _view.Inputdev_list.append("-----
-----
---" + "\n");
    }
}
else {
    if ((inputcopy_v[dev_in][0] <= PFHd_input) && (hyd_input ==
true)) {
        inputprint_n[counter_in][0] = DevList_input_n[dev_in][0];
        inputprint_n[counter_in][1] = DevList_input_n[dev_in][1];
        inputprint_n[counter_in][2] = DevList_input_n[dev_in][2];
        inputprint_n[counter_in][3] = DevList_input_n[dev_in][3];
        inputprint_v[counter_in][0] = DevList_input_v[dev_in][0];
        inputprint_v[counter_in][1] = DevList_input_v[dev_in][1];
        inputprint_v[counter_in][2] = DevList_input_v[dev_in][2];
        inputprint_v[counter_in][3] = DevList_input_v[dev_in][3];
        inputprint_v[counter_in][4] = DevList_input_v[dev_in][4];
        inputprint_v[counter_in][5] = DevList_input_v[dev_in][5];
        counter_in = counter_in + 1;
        _view.Inputdev_list.append(String.valueOf(counter_in) + ".
" + DevList_input_n[dev_in][0] + " | " + DevList_input_n[dev_in][1] +
" | PL " + DevList_input_n[dev_in][2] + " | PFHd " +
String.valueOf(DevList_input_v[dev_in][0]) + " | MTTFd " +
String.valueOf(DevList_input_v[dev_in][1]) + " | Technology " +
DevList_input_n[dev_in][3] + " | B10d " +
String.valueOf(DevList_input_v[dev_in][2]) + " | nop " +
String.valueOf(DevList_input_v[dev_in][3]) + " | T10d " +
String.valueOf(DevList_input_v[dev_in][4]) + " | Price " +
String.valueOf(DevList_input_v[dev_in][5]) + " €" + "\n");
        _view.Inputdev_list.append("-----
-----
---" + "\n");
    }
}
}
}

```



```

        else {
            break;
        }
    }
}
// Logic device.
for (dev_lo = 0; dev_lo < 100; dev_lo++) {
    if (!DevList_logic_n[dev_lo][0].equals("o")) {
        if (((logiccopy_n[dev_lo][3].equals("Mechanical")) ||
(logiccopy_n[dev_lo][3].equals("Pneumatic"))) && (hyd_logic == false))
{
            if ((logiccopy_v[dev_lo][0] <= PFHd_logic) &&
((logiccopy_v[dev_lo][2] >= B10d_logic) || (logiccopy_v[dev_lo][4] >=
T10d_logic))) {
                logicprint_n[counter_lo][0] = DevList_logic_n[dev_lo][0];
                logicprint_n[counter_lo][1] = DevList_logic_n[dev_lo][1];
                logicprint_n[counter_lo][2] = DevList_logic_n[dev_lo][2];
                logicprint_n[counter_lo][3] = DevList_logic_n[dev_lo][3];
                logicprint_v[counter_lo][0] = DevList_logic_v[dev_lo][0];
                logicprint_v[counter_lo][1] = DevList_logic_v[dev_lo][1];
                logicprint_v[counter_lo][2] = DevList_logic_v[dev_lo][2];
                logicprint_v[counter_lo][3] = DevList_logic_v[dev_lo][3];
                logicprint_v[counter_lo][4] = DevList_logic_v[dev_lo][4];
                logicprint_v[counter_lo][5] = DevList_logic_v[dev_lo][5];
                counter_lo = counter_lo + 1;
                _view.Logicdev_list.append(String.valueOf(counter_lo) + ".
" + DevList_logic_n[dev_lo][0] + " | " + DevList_logic_n[dev_lo][1] +
" | PL " + DevList_logic_n[dev_lo][2] + " | PFHd " +
String.valueOf(DevList_logic_v[dev_lo][0]) + " | MTTFd " +
String.valueOf(DevList_logic_v[dev_lo][1]) + " | Technology " +
DevList_logic_n[dev_lo][3] + " | B10d " +
String.valueOf(DevList_logic_v[dev_lo][2]) + " | nop " +
String.valueOf(DevList_logic_v[dev_lo][3]) + " | T10d " +
String.valueOf(DevList_logic_v[dev_lo][4]) + " | Price " +
String.valueOf(DevList_logic_v[dev_lo][5]) + " €" + "\n");
                _view.Logicdev_list.append("-----
-----
-----
---" + "\n");
            }
        }
    }
    else {
        if ((logiccopy_v[dev_lo][0] <= PFHd_logic) && (hyd_logic ==
true)) {
            logicprint_n[counter_lo][0] = DevList_logic_n[dev_lo][0];
            logicprint_n[counter_lo][1] = DevList_logic_n[dev_lo][1];
            logicprint_n[counter_lo][2] = DevList_logic_n[dev_lo][2];
            logicprint_n[counter_lo][3] = DevList_logic_n[dev_lo][3];
            logicprint_v[counter_lo][0] = DevList_logic_v[dev_lo][0];
            logicprint_v[counter_lo][1] = DevList_logic_v[dev_lo][1];
            logicprint_v[counter_lo][2] = DevList_logic_v[dev_lo][2];
            logicprint_v[counter_lo][3] = DevList_logic_v[dev_lo][3];
            logicprint_v[counter_lo][4] = DevList_logic_v[dev_lo][4];
            logicprint_v[counter_lo][5] = DevList_logic_v[dev_lo][5];

```

```

        counter_lo = counter_lo + 1;
        _view.Logicdev_list.append(String.valueOf(counter_lo) + ".
" + DevList_logic_n[dev_lo][0] + " | " + DevList_logic_n[dev_lo][1] +
" | PL " + DevList_logic_n[dev_lo][2] + " | PFHd " +
String.valueOf(DevList_logic_v[dev_lo][0]) + " | MTTFd " +
String.valueOf(DevList_logic_v[dev_lo][1]) + " | Technology " +
DevList_logic_n[dev_lo][3] + " | B10d " +
String.valueOf(DevList_logic_v[dev_lo][2]) + " | nop " +
String.valueOf(DevList_logic_v[dev_lo][3]) + " | T10d " +
String.valueOf(DevList_logic_v[dev_lo][4]) + " | Price " +
String.valueOf(DevList_logic_v[dev_lo][5]) + " €" + "\n");
        _view.Logicdev_list.append("-----
-----
---" + "\n");
    }
}
else {
    break;
}
}
// Output device.
for (dev_out = 0; dev_out < 100; dev_out++) {
    if (!DevList_output_n[dev_out][0].equals("o")) {
        if (((outputcopy_n[dev_out][3].equals("Mechanical")) ||
(outputcopy_n[dev_out][3].equals("Pneumatic")))) && (hyd_output ==
false)) {
            if ((outputcopy_v[dev_out][0] <= PFHd_output) &&
((outputcopy_v[dev_out][2] >= B10d_output) ||
(outputcopy_v[dev_out][4] >= T10d_output)) {
                outputprint_n[counter_out][0] =
DevList_output_n[dev_out][0];
                outputprint_n[counter_out][1] =
DevList_output_n[dev_out][1];
                outputprint_n[counter_out][2] =
DevList_output_n[dev_out][2];
                outputprint_n[counter_out][3] =
DevList_output_n[dev_out][3];
                outputprint_v[counter_out][0] =
DevList_output_v[dev_out][0];
                outputprint_v[counter_out][1] =
DevList_output_v[dev_out][1];
                outputprint_v[counter_out][2] =
DevList_output_v[dev_out][2];
                outputprint_v[counter_out][3] =
DevList_output_v[dev_out][3];
                outputprint_v[counter_out][4] =
DevList_output_v[dev_out][4];
                outputprint_v[counter_out][5] =
DevList_output_v[dev_out][5];
                counter_out = counter_out + 1;
                _view.Outputdev_list.append(String.valueOf(counter_out) +

```


ecoCompany_in	-	String	-
ecoPL_in	-	String	-
ecoPFHd_in	-	double	-
ecoMTTFd_in	-	double	-
ecoTechnology_in	-	String	-
ecoB10d_in	-	double	-
econop_in	-	double	-
ecoT10d_in	-	double	-
ecoPrice_in	-	double	-
ecoName_lo	-	String	-
ecoCompany_lo	-	String	-
ecoPL_lo	-	String	-
ecoPFHd_lo	-	double	-
ecoMTTFd_lo	-	double	-
ecoTechnology_lo	-	String	-
ecoB10d_lo	-	double	-
econop_lo	-	double	-
ecoT10d_lo	-	double	-
ecoPrice_lo	-	double	-
ecoName_out	-	String	-
ecoCompany_out	-	String	-
ecoPL_out	-	String	-
ecoPFHd_out	-	double	-
ecoMTTFd_out	-	double	-
ecoTechnology_out	-	String	-
ecoB10d_out	-	double	-
econop_out	-	double	-
ecoT10d_out	-	double	-
ecoPrice_out	-	double	-
ecoPFHd_total	-	double	-
ecoPrice_total	-	double	-

B8.1. Parámetros del botón “Optimización económica”.

```

public void economicOpt () {
    if (EconomicButton == true) {
        if (PFHr == 0) {
            _view.Operations.append("> Error: optimize your system first!" +
"\n");
            showEconOpticDialog = false;
        } // if PFHr == 0
        else {
            priceref = 1*Math.pow(10,10);
        }
    }
}

```

```

    PFHd_eco = PFHr;

// MATRICES.
// Input device matrix.
    for (ecount_in = 0; ecount_in < 100; ecount_in++) {
        if (!DevList_input_n[ecount_in][0].equals("o")) {
            EcoList_input_v[ecount_in][0] =
DevList_input_v[ecount_in][0];
            EcoList_input_v[ecount_in][1] =
DevList_input_v[ecount_in][1];
            if (DevList_input_v[ecount_in][0] == 0) {
                if (category.equals("B")) {
                    EcoList_input_v[ecount_in][0] =
0.0001*Math.pow(EcoList_input_v[ecount_in][1],-1.0002);
                }
                else if (category.equals("1")) {
                    EcoList_input_v[ecount_in][0] =
0.0001*Math.pow(EcoList_input_v[ecount_in][1],-0.9985);
                }
                else if (category.equals("2")) {
                    EcoList_input_v[ecount_in][0] =
0.0002*Math.pow(EcoList_input_v[ecount_in][1],-0.8975);
                }
                else if (category.equals("3")) {
                    EcoList_input_v[ecount_in][0] =
0.001*Math.pow(EcoList_input_v[ecount_in][1],-0.7191);
                }
                else if (category.equals("4")) {
                    EcoList_input_v[ecount_in][0] =
0.00002*Math.pow(EcoList_input_v[ecount_in][1],-0.89);
                }
            }
        }
        else {
            break;
        }
    }
// Logic device matrix.
    for (ecount_lo = 0; ecount_lo < 100; ecount_lo++) {
        if (!DevList_logic_n[ecount_lo][0].equals("o")) {
            EcoList_logic_v[ecount_lo][0] =
DevList_logic_v[ecount_lo][0];
            EcoList_logic_v[ecount_lo][1] =
DevList_logic_v[ecount_lo][1];
            if (DevList_logic_v[ecount_lo][0] == 0) {
                if (category.equals("B")) {
                    EcoList_logic_v[ecount_lo][0] =
0.0001*Math.pow(EcoList_logic_v[ecount_lo][1],-1.0002);
                }
                else if (category.equals("1")) {
                    EcoList_logic_v[ecount_lo][0] =
0.0001*Math.pow(EcoList_logic_v[ecount_lo][1],-0.9985);
                }
            }
        }
    }

```

```
        else if (category.equals("2")) {
            EcoList_logic_v[ecount_lo][0] =
0.0002*Math.pow(EcoList_logic_v[ecount_lo][1],-0.8975);
        }
        else if (category.equals("3")) {
            EcoList_logic_v[ecount_lo][0] =
0.001*Math.pow(EcoList_logic_v[ecount_lo][1],-0.7191);
        }
        else if (category.equals("4")) {
            EcoList_logic_v[ecount_lo][0] =
0.00002*Math.pow(EcoList_logic_v[ecount_lo][1],-0.89);
        }
    }
}
else {
    break;
}
}
// Output device matrix.
for (ecount_out = 0; ecount_out < 100; ecount_out++) {
    if (!DevList_output_n[ecount_out][0].equals("o")) {
        EcoList_output_v[ecount_out][0] =
DevList_output_v[ecount_out][0];
        EcoList_output_v[ecount_out][1] =
DevList_output_v[ecount_out][1];
        if (DevList_output_v[ecount_out][0] == 0) {
            if (category.equals("B")) {
                EcoList_output_v[ecount_out][0] =
0.0001*Math.pow(EcoList_output_v[ecount_out][1],-1.0002);
            }
            else if (category.equals("1")) {
                EcoList_output_v[ecount_out][0] =
0.0001*Math.pow(EcoList_output_v[ecount_out][1],-0.9985);
            }
            else if (category.equals("2")) {
                EcoList_output_v[ecount_out][0] =
0.0002*Math.pow(EcoList_output_v[ecount_out][1],-0.8975);
            }
            else if (category.equals("3")) {
                EcoList_output_v[ecount_out][0] =
0.001*Math.pow(EcoList_output_v[ecount_out][1],-0.7191);
            }
            else if (category.equals("4")) {
                EcoList_output_v[ecount_out][0] =
0.00002*Math.pow(EcoList_output_v[ecount_out][1],-0.89);
            }
        }
    }
}
else {
    break;
}
}
```

```

// SELECTION.
for (row_eco_in = 0; row_eco_in < 100; row_eco_in++) {
    if (!DevList_input_n[row_eco_in][0].equals("o")) {
        for(row_eco_lo = 0; row_eco_lo < 100; row_eco_lo++) {
            if (!DevList_logic_n[row_eco_lo][0].equals("o")) {
                for(row_eco_out = 0; row_eco_out < 100; row_eco_out++) {
                    if (!DevList_output_n[row_eco_out][0].equals("o")) {
                        if (((EcoList_input_v[row_eco_in][0] +
EcoList_logic_v[row_eco_lo][0] + EcoList_output_v[row_eco_out][0]) <=
PFHd_eco) && ((DevList_input_v[row_eco_in][5] +
DevList_logic_v[row_eco_lo][5] + DevList_output_v[row_eco_out][5]) <=
priceref)) {
                            PFHd_eco = EcoList_input_v[row_eco_in][0] +
EcoList_logic_v[row_eco_lo][0] + EcoList_output_v[row_eco_out][0];
                            priceref = DevList_input_v[row_eco_in][5] +
DevList_logic_v[row_eco_lo][5] + DevList_output_v[row_eco_out][5];
                            econ_in = row_eco_in;
                            econ_lo = row_eco_lo;
                            econ_out = row_eco_out;
                        }
                    }
                }
            }
        }
    }
}

// VISUALIZATION
if (((EcoList_input_v[econ_in][0] + EcoList_logic_v[econ_lo][0]
+ EcoList_output_v[econ_out][0]) <= PFHd_eco) &&
((DevList_input_v[econ_in][5] + DevList_logic_v[econ_lo][5] +
DevList_output_v[econ_out][5]) <= priceref) &&
(EcoList_input_v[econ_in][0] > 0 ) && (EcoList_logic_v[econ_lo][0] >
0) && (EcoList_output_v[econ_out][0] > 0)) {
    _view.Operations.append("> An economically optimized solution
was found!" + "\n");

    ecoName_in = DevList_input_n[econ_in][0];
    ecoCompany_in = DevList_input_n[econ_in][1];
    ecoPL_in = DevList_input_n[econ_in][2];
    ecoPFHd_in = DevList_input_v[econ_in][0];
    ecoMTTFd_in = DevList_input_v[econ_in][1];
    ecoTechnology_in = DevList_input_n[econ_in][3];
    ecoB10d_in = DevList_input_v[econ_in][2];
}

```



```

econop_in = DevList_input_v[econ_in][3];
ecoT10d_in = DevList_input_v[econ_in][4];
ecoPrice_in = DevList_input_v[econ_in][5];

ecoName_lo = DevList_logic_n[econ_lo][0];
ecoCompany_lo = DevList_logic_n[econ_lo][1];
ecoPL_lo = DevList_logic_n[econ_lo][2];
ecoPFHd_lo = DevList_logic_v[econ_lo][0];
ecoMTTFd_lo = DevList_logic_v[econ_lo][1];
ecoTechnology_lo = DevList_logic_n[econ_lo][3];
ecoB10d_lo = DevList_logic_v[econ_lo][2];
econop_lo = DevList_logic_v[econ_lo][3];
ecoT10d_lo = DevList_logic_v[econ_lo][4];
ecoPrice_lo = DevList_logic_v[econ_lo][5];

ecoName_out = DevList_output_n[econ_out][0];
ecoCompany_out = DevList_output_n[econ_out][1];
ecoPL_out = DevList_output_n[econ_out][2];
ecoPFHd_out = DevList_output_v[econ_out][0];
ecoMTTFd_out = DevList_output_v[econ_out][1];
ecoTechnology_out = DevList_output_n[econ_out][3];
ecoB10d_out = DevList_output_v[econ_out][2];
econop_out = DevList_output_v[econ_out][3];
ecoT10d_out = DevList_output_v[econ_out][4];
ecoPrice_out = DevList_output_v[econ_out][5];

ecoPFHd_total = PFHd_eco;
ecoPrice_total = priceref;

showEconOpticDialog = true;
}
else {
    _view.Operations.append("> No economically optimized solution
was found!" + "\n");
    showEconOpticDialog = false;
}
}
}
}
}

```

B9. Activación de botones y ventanas.

Nombre	Valor inicial	Tipo	Dimensión
MachineButton	true	boolean	-
RiskButton	true	boolean	-
SecFunButton	true	boolean	-
SecLevButton	true	boolean	-
SecDevListButton	true	boolean	-
HelpButton	true	boolean	-

DevListButton	true	boolean	-
EconomicButton	true	boolean	-
AddMachineButton	true	boolean	-
AddRiskButton	true	boolean	-
AddSecFunButton	true	boolean	-
AddSecLevButton	true	boolean	-

Tabla B9.1. Parámetros relativos a la activación de botones.

Nombre	Valor inicial	Tipo	Dimensión
showMachineDialog	false	boolean	-
showRiskDialog	false	boolean	-
showSecFunDialog	false	boolean	-
showSecDevDialog	false	boolean	-
showHelpDialog	false	boolean	-
showDevListDialog	false	boolean	-
showEconOpticDialog	false	boolean	-

Tabla B9.2. Parámetros relativos a la activación de ventanas.

```
// Machine Dialog (OPEN/CLOSE).
public void showMachineDialog () {
    if (MachineButton == true) {
        showMachineDialog = true;
    }
}
// Risk Dialog (OPEN/CLOSE).
public void showRiskDialog () {
    if (RiskButton == true) {
        showRiskDialog = true;
    }
}
// Safety Function Dialog (OPEN/CLOSE).
public void showSecFunDialog () {
    if (SecFunButton == true) {
        showSecFunDialog = true;
    }
}
// Safety Level Dialog (OPEN/CLOSE).
public void showSecLevDialog () {
    if (SecLevButton == true) {
        showSecLevDialog = true;
    }
}
// Help Dialog (OPEN).
public void showHelpDialog () {
```

```
if (HelpButton == true) {  
    showHelpDialog = true;  
}  
}
```

B10. Gráfico general.

Nombre	Valor inicial	Tipo	Dimensión
PLr_x	2	double	-
PLr_y	4	double	-
showPLrpoint	false	boolean	-

Tabla 10.1. Parámetros del gráfico general.

B11. Gráfico $B_{10d} = f(n_{op})$.

B11.1. Dispositivo de entrada.

```
if (nop_input >= 15000) {  
    nop_input = 15000;  
}  
else if (nop_input <= 0) {  
    nop_input = 0;  
}  
  
if (category.equals("B")) {  
    B10d_input = 0.1*1.14*0.0001*Math.pow(PFHd_input,-1.0002)*nop_input;  
}  
else if (category.equals("1")) {  
    B10d_input = 0.1*1.16*0.0001*Math.pow(PFHd_input,-0.9985)*nop_input;  
}  
else if (category.equals("2")) {  
    B10d_input = 0.1*0.0002*Math.pow(PFHd_input,-0.8975)*nop_input;  
}  
else if (category.equals("3")) {  
    B10d_input = 0.1*0.001*Math.pow(PFHd_input,-0.7191)*nop_input;  
}  
else if (category.equals("4")) {  
    B10d_input = 0.1*0.85*0.00002*Math.pow(PFHd_input,-0.89)*nop_input;  
}  
  
T10d_input = B10d_input/nop_input;
```

B11.2. Unidad lógica.

```
if (nop_logic >= 15000) {
    nop_logic = 15000;
}
else if (nop_logic <= 0) {
    nop_logic = 0;
}

if (category.equals("B")) {
    B10d_logic = 0.1*1.14*0.0001*Math.pow(PFHd_logic,-1.0002)*nop_logic;
}
else if (category.equals("1")) {
    B10d_logic = 0.1*1.16*0.0001*Math.pow(PFHd_logic,-0.9985)*nop_logic;
}
else if (category.equals("2")) {
    B10d_logic = 0.1*0.0002*Math.pow(PFHd_logic,-0.8975)*nop_logic;
}
else if (category.equals("3")) {
    B10d_logic = 0.1*0.001*Math.pow(PFHd_logic,-0.7191)*nop_logic;
}
else if (category.equals("4")) {
    B10d_logic = 0.1*0.85*0.00002*Math.pow(PFHd_logic,-0.89)*nop_logic;
}

T10d_logic = B10d_logic/nop_logic;
```

B11.3. Dispositivo de salida.

```
if (nop_output >= 15000){
    nop_output = 15000;
}
else if (nop_output <= 0) {
    nop_output = 0;
}

if (category.equals("B")) {
    B10d_output = 0.1*1.14*0.0001*Math.pow(PFHd_output,-
1.0002)*nop_output;
}
else if (category.equals("1")) {
    B10d_output = 0.1*1.16*0.0001*Math.pow(PFHd_output,-
0.9985)*nop_output;
}
else if (category.equals("2")) {
    B10d_output = 0.1*0.0002*Math.pow(PFHd_output,-0.8975)*nop_output;
}
else if (category.equals("3")) {
    B10d_output = 0.1*0.001*Math.pow(PFHd_output,-0.7191)*nop_output;
}
}
```

```
else if (category.equals("4")) {
    B10d_output = 0.1*0.85*0.00002*Math.pow(PFHd_output,-
0.89)*nop_output;
}
T10d_output = B10d_output/nop_output;
```

B12. Gráfico $MTTF_d = f(PFH_d)$.

B12.1. Dispositivo de entrada.

```
// INPUT
// Determination of PFHd.
if (fix_input == false && fix_logic == false && fix_output == false) {
    if (category.equals("B")) {
        if (PFHd_input >= 2.954*Math.pow(10,-5)){
            PFHd_input = 2.954*Math.pow(10,-5);
            if (PFHd_logic <= 4.23*Math.pow(10,-6) && PFHd_output >
4.23*Math.pow(10,-6)) {
                PFHd_logic = 4.23*Math.pow(10,-6);
                PFHd_output = PFHr - PFHd_input - PFHd_logic;
                if (PFHd_output <= 4.23*Math.pow(10,-6)) {
                    PFHd_output = 4.23*Math.pow(10,-6);
                }
            } // lo min
            else if (PFHd_output <= 4.23*Math.pow(10,-6) && PFHd_logic >
4.23*Math.pow(10,-6)) {
                PFHd_output = 4.23*Math.pow(10,-6);
                PFHd_logic = PFHr - PFHd_input - PFHd_output;
                if (PFHd_logic <= 4.23*Math.pow(10,-6)) {
                    PFHd_logic = 4.23*Math.pow(10,-6);
                }
            } // out min
            else if (PFHd_logic <= 4.23*Math.pow(10,-6) && PFHd_output <=
4.23*Math.pow(10,-6)) {
                PFHd_logic = 4.23*Math.pow(10,-6);
                PFHd_output = 4.23*Math.pow(10,-6);
            } // lo min out min
        } // in max
        else if (PFHd_input <= 4.23*Math.pow(10,-6)) {
            PFHd_input = 4.23*Math.pow(10,-6);
            if (PFHd_logic >= 2.954*Math.pow(10,-5) && PFHd_output <
2.954*Math.pow(10,-5)){
                PFHd_logic = 2.954*Math.pow(10,-5);
                PFHd_output = PFHr - PFHd_input - PFHd_logic;
                if (PFHd_output >= 2.954*Math.pow(10,-5)) {
                    PFHd_output = 2.954*Math.pow(10,-5);
                }
            } // lo max
            else if (PFHd_output >= 2.954*Math.pow(10,-5) && PFHd_logic <
```

```

2.954*Math.pow(10,-5)){
    PFHd_output = 2.954*Math.pow(10,-5);
    PFHd_logic = PFHr - PFHd_input - PFHd_output;
    if (PFHd_logic >= 2.954*Math.pow(10,-5)) {
        PFHd_logic = 2.954*Math.pow(10,-5);
    }
    } // out max
    else if (PFHd_logic >= 2.954*Math.pow(10,-5) && PFHd_output >=
2.954*Math.pow(10,-5)){
        PFHd_logic = 2.954*Math.pow(10,-5);
        PFHd_output = 2.954*Math.pow(10,-5);
    } // lo max out max
    } // in min
    else {
        Dinput = PFHd0_input - PFHd_input;
        PFHd_logic = PFHd_logic + Dinput/2;
        PFHd_output = PFHd_output + Dinput/2;
        if (PFHd_logic >= 2.954*Math.pow(10,-5) && PFHd_output <
2.954*Math.pow(10,-5)) {
            PFHd_logic = 2.954*Math.pow(10,-5);
            PFHd_output = PFHr - PFHd_input - PFHd_logic;
            if (PFHd_output >= 2.954*Math.pow(10,-5)) {
                PFHd_output = 2.954*Math.pow(10,-5);
                PFHd_input = PFHr - PFHd_logic - PFHd_output;
            }
        } // lo max
        else if (PFHd_output >= 2.954*Math.pow(10,-5) && PFHd_logic <
2.954*Math.pow(10,-5)) {
            PFHd_output = 2.954*Math.pow(10,-5);
            PFHd_logic = PFHr - PFHd_input - PFHd_output;
            if (PFHd_logic >= 2.954*Math.pow(10,-5)) {
                PFHd_output = 2.954*Math.pow(10,-5);
                PFHd_input = PFHr - PFHd_logic - PFHd_output;
            }
        } // out max
        else if (PFHd_logic >= 2.954*Math.pow(10,-5) && PFHd_output >=
2.954*Math.pow(10,-5)) {
            PFHd_logic = 2.954*Math.pow(10,-5);
            PFHd_output = 2.954*Math.pow(10,-5);
            PFHd_input = PFHr - PFHd_logic - PFHd_output;
        } // lo max out max
        else if (PFHd_logic <= 4.23*Math.pow(10,-6) && PFHd_output >
4.23*Math.pow(10,-6)) {
            PFHd_logic = 4.23*Math.pow(10,-6);
            PFHd_output = PFHr - PFHd_input - PFHd_logic;
            if (PFHd_output <= 4.23*Math.pow(10,-6)) {
                PFHd_output = 4.23*Math.pow(10,-6);
                PFHd_input = PFHr - PFHd_logic - PFHd_output;
            }
        } // lo min
        else if (PFHd_output <= 4.23*Math.pow(10,-6) && PFHd_logic >
4.23*Math.pow(10,-6)) {
            PFHd_output = 4.23*Math.pow(10,-6);

```

```
    PFHd_logic = PFHr - PFHd_input - PFHd_output;
    if (PFHd_logic <= 4.23*Math.pow(10,-6)) {
        PFHd_logic = 4.23*Math.pow(10,-6);
        PFHd_input = PFHr - PFHd_logic - PFHd_output;
    }
    } // out min
    else if (PFHd_logic <= 4.23*Math.pow(10,-6) && PFHd_output <=
4.23*Math.pow(10,-6)) {
        PFHd_logic = 4.23*Math.pow(10,-6);
        PFHd_output = 4.23*Math.pow(10,-6);
        PFHd_input = PFHr - PFHd_logic - PFHd_output;
    } // lo min out min
    } // else
} // cat B
if (category.equals("1")) {
    if (PFHd_input >= 1.52*Math.pow(10,-6)){
        PFHd_input = 1.52*Math.pow(10,-6);
        if (PFHd_logic <= 1.14*Math.pow(10,-6) && PFHd_output >
1.14*Math.pow(10,-6)) {
            PFHd_logic = 1.14*Math.pow(10,-6);
            PFHd_output = PFHr - PFHd_input - PFHd_logic;
            if (PFHd_output <= 1.14*Math.pow(10,-6)) {
                PFHd_output = 1.14*Math.pow(10,-6);
            }
        } // lo min
        else if (PFHd_output <= 1.14*Math.pow(10,-6) && PFHd_logic >
1.14*Math.pow(10,-6)) {
            PFHd_output = 1.14*Math.pow(10,-6);
            PFHd_logic = PFHr - PFHd_input - PFHd_output;
            if (PFHd_logic <= 1.14*Math.pow(10,-6)) {
                PFHd_logic = 1.14*Math.pow(10,-6);
            }
        } // out min
        else if (PFHd_logic <= 1.14*Math.pow(10,-6) && PFHd_output <=
1.14*Math.pow(10,-6)) {
            PFHd_logic = 1.14*Math.pow(10,-6);
            PFHd_output = 1.14*Math.pow(10,-6);
        } // lo min out min
    } // in max
    else if (PFHd_input <= 1.14*Math.pow(10,-6)) {
        PFHd_input = 1.14*Math.pow(10,-6);
        if (PFHd_logic >= 1.52*Math.pow(10,-6) && PFHd_output <
1.52*Math.pow(10,-6)){
            PFHd_logic = 1.52*Math.pow(10,-6);
            PFHd_output = PFHr - PFHd_input - PFHd_logic;
            if (PFHd_output >= 1.52*Math.pow(10,-6)) {
                PFHd_output = 1.52*Math.pow(10,-6);
            }
        } // lo max
        else if (PFHd_output >= 1.52*Math.pow(10,-6) && PFHd_logic <
1.52*Math.pow(10,-6)){
            PFHd_output = 1.52*Math.pow(10,-6);
            PFHd_logic = PFHr - PFHd_input - PFHd_output;
```

```

        if (PFHd_logic >= 1.52*Math.pow(10,-6)) {
            PFHd_logic = 1.52*Math.pow(10,-6);
        }
    } // out max
    else if (PFHd_logic >= 1.52*Math.pow(10,-6) && PFHd_output >=
1.52*Math.pow(10,-6)) {
        PFHd_logic = 1.52*Math.pow(10,-6);
        PFHd_output = 1.52*Math.pow(10,-6);
    } // lo max out max
} // in min
else {
    Dinput = PFHd0_input - PFHd_input;
    PFHd_logic = PFHd_logic + Dinput/2;
    PFHd_output = PFHd_output + Dinput/2;
    if (PFHd_logic >= 1.52*Math.pow(10,-6) && PFHd_output <
1.52*Math.pow(10,-6)) {
        PFHd_logic = 1.52*Math.pow(10,-6);
        PFHd_output = PFHr - PFHd_input - PFHd_logic;
        if (PFHd_output >= 1.52*Math.pow(10,-6)) {
            PFHd_output = 1.52*Math.pow(10,-6);
            PFHd_input = PFHr - PFHd_logic - PFHd_output;
        }
    } // lo max
    else if (PFHd_output >= 1.52*Math.pow(10,-6) && PFHd_logic <
1.52*Math.pow(10,-6)) {
        PFHd_output = 1.52*Math.pow(10,-6);
        PFHd_logic = PFHr - PFHd_input - PFHd_output;
        if (PFHd_logic >= 1.52*Math.pow(10,-6)) {
            PFHd_logic = 1.52*Math.pow(10,-6);
            PFHd_input = PFHr - PFHd_logic - PFHd_output;
        }
    } // out max
    else if (PFHd_logic >= 1.52*Math.pow(10,-6) && PFHd_output >=
1.52*Math.pow(10,-6)) {
        PFHd_logic = 1.52*Math.pow(10,-6);
        PFHd_output = 1.52*Math.pow(10,-6);
        PFHd_input = PFHr - PFHd_logic - PFHd_output;
    } // lo max out max
    else if (PFHd_logic <= 1.14*Math.pow(10,-6) && PFHd_output >
1.14*Math.pow(10,-6)) {
        PFHd_logic = 1.14*Math.pow(10,-6);
        PFHd_output = PFHr - PFHd_input - PFHd_logic;
        if (PFHd_output <= 1.14*Math.pow(10,-6)) {
            PFHd_output = 1.14*Math.pow(10,-6);
            PFHd_input = PFHr - PFHd_logic - PFHd_output;
        }
    } // lo min
    else if (PFHd_output <= 1.14*Math.pow(10,-6) && PFHd_logic >
1.14*Math.pow(10,-6)) {
        PFHd_output = 1.14*Math.pow(10,-6);
        PFHd_logic = PFHr - PFHd_input - PFHd_output;
        if (PFHd_logic <= 1.14*Math.pow(10,-6)) {
            PFHd_logic = 1.14*Math.pow(10,-6);

```



```
        PFHd_input = PFHr - PFHd_logic - PFHd_output;
    }
    } // out min
    else if (PFHd_logic <= 1.14*Math.pow(10,-6) && PFHd_output <=
1.14*Math.pow(10,-6)) {
        PFHd_logic = 1.14*Math.pow(10,-6);
        PFHd_output = 1.14*Math.pow(10,-6);
        PFHd_input = PFHr - PFHd_logic - PFHd_output;
    } // lo min out min
    } // else
} // cat 1
if (category.equals("2")) {
    if (PFHd_input >= 2.4744*Math.pow(10,-5)){
        PFHd_input = 2.4744*Math.pow(10,-5);
        if (PFHd_logic <= 5.28*Math.pow(10,-7) && PFHd_output >
5.28*Math.pow(10,-7)) {
            PFHd_logic = 5.28*Math.pow(10,-7);
            PFHd_output = PFHr - PFHd_input - PFHd_logic;
            if (PFHd_output <= 5.28*Math.pow(10,-7)) {
                PFHd_output = 5.28*Math.pow(10,-7);
            }
        } // lo min
        else if (PFHd_output <= 5.28*Math.pow(10,-7) && PFHd_logic >
5.28*Math.pow(10,-7)) {
            PFHd_output = 5.28*Math.pow(10,-7);
            PFHd_logic = PFHr - PFHd_input - PFHd_output;
            if (PFHd_logic <= 5.28*Math.pow(10,-7)) {
                PFHd_logic = 5.28*Math.pow(10,-7);
            }
        } // out min
        else if (PFHd_logic <= 5.28*Math.pow(10,-7) && PFHd_output <=
5.28*Math.pow(10,-7)) {
            PFHd_logic = 5.28*Math.pow(10,-7);
            PFHd_output = 5.28*Math.pow(10,-7);
        } // lo min out min
    } // in max
    else if (PFHd_input <= 5.28*Math.pow(10,-7)) {
        PFHd_input = 5.28*Math.pow(10,-7);
        if (PFHd_logic >= 2.4744*Math.pow(10,-5) && PFHd_output <
2.4744*Math.pow(10,-5)){
            PFHd_logic = 2.4744*Math.pow(10,-5);
            PFHd_output = PFHr - PFHd_input - PFHd_logic;
            if (PFHd_output >= 2.4744*Math.pow(10,-5)) {
                PFHd_output = 2.4744*Math.pow(10,-5);
            }
        } // lo max
        else if (PFHd_output >= 2.4744*Math.pow(10,-5) && PFHd_logic <
2.4744*Math.pow(10,-5)){
            PFHd_output = 2.4744*Math.pow(10,-5);
            PFHd_logic = PFHr - PFHd_input - PFHd_output;
            if (PFHd_logic >= 2.4744*Math.pow(10,-5)) {
                PFHd_logic = 2.4744*Math.pow(10,-5);
            }
        }
    }
}
```

```

    } // out max
    else if (PFHd_logic >= 2.4744*Math.pow(10,-5) && PFHd_output >=
2.4744*Math.pow(10,-5)){
        PFHd_logic = 2.4744*Math.pow(10,-5);
        PFHd_output = 2.4744*Math.pow(10,-5);
    } // lo max out max
} // in min
else {
    Dinput = PFHd0_input - PFHd_input;
    PFHd_logic = PFHd_logic + Dinput/2;
    PFHd_output = PFHd_output + Dinput/2;
    if (PFHd_logic >= 2.4744*Math.pow(10,-5) && PFHd_output <
2.4744*Math.pow(10,-5)) {
        PFHd_logic = 2.4744*Math.pow(10,-5);
        PFHd_output = PFHr - PFHd_input - PFHd_logic;
        if (PFHd_output >= 2.4744*Math.pow(10,-5)) {
            PFHd_output = 2.4744*Math.pow(10,-5);
            PFHd_input = PFHr - PFHd_logic - PFHd_output;
        }
    } // lo max
    else if (PFHd_output >= 2.4744*Math.pow(10,-5) && PFHd_logic <
2.4744*Math.pow(10,-5)) {
        PFHd_output = 2.4744*Math.pow(10,-5);
        PFHd_logic = PFHr - PFHd_input - PFHd_output;
        if (PFHd_logic >= 2.4744*Math.pow(10,-5)) {
            PFHd_logic = 2.4744*Math.pow(10,-5);
            PFHd_input = PFHr - PFHd_logic - PFHd_output;
        }
    } // out max
    else if (PFHd_logic >= 2.4744*Math.pow(10,-5) && PFHd_output >=
2.4744*Math.pow(10,-5)) {
        PFHd_logic = 2.4744*Math.pow(10,-5);
        PFHd_output = 2.4744*Math.pow(10,-5);
        PFHd_input = PFHr - PFHd_logic - PFHd_output;
    } // lo max out max
    else if (PFHd_logic <= 5.28*Math.pow(10,-7) && PFHd_output >
5.28*Math.pow(10,-7)) {
        PFHd_logic = 5.28*Math.pow(10,-7);
        PFHd_output = PFHr - PFHd_input - PFHd_logic;
        if (PFHd_output <= 5.28*Math.pow(10,-7)) {
            PFHd_output = 5.28*Math.pow(10,-7);
            PFHd_input = PFHr - PFHd_logic - PFHd_output;
        }
    } // lo min
    else if (PFHd_output <= 5.28*Math.pow(10,-7) && PFHd_logic >
5.28*Math.pow(10,-7)) {
        PFHd_output = 5.28*Math.pow(10,-7);
        PFHd_logic = PFHr - PFHd_input - PFHd_output;
        if (PFHd_logic <= 5.28*Math.pow(10,-7)) {
            PFHd_logic = 5.28*Math.pow(10,-7);
            PFHd_input = PFHr - PFHd_logic - PFHd_output;
        }
    } // out min
}

```

```

        else if (PFHd_logic <= 5.28*Math.pow(10,-7) && PFHd_output <=
5.28*Math.pow(10,-7)) {
            PFHd_logic = 5.28*Math.pow(10,-7);
            PFHd_output = 5.28*Math.pow(10,-7);
            PFHd_input = PFHr - PFHd_logic - PFHd_output;
        } // lo min out min
    } // else
} // cat 2
if (category.equals("3")) {
    if (PFHd_input >= 1.2398*Math.pow(10,-5)){
        PFHd_input = 1.2398*Math.pow(10,-5);
        if (PFHd_logic <= 1.01*Math.pow(10,-7) && PFHd_output >
1.01*Math.pow(10,-7)) {
            PFHd_logic = 1.01*Math.pow(10,-7);
            PFHd_output = PFHr - PFHd_input - PFHd_logic;
            if (PFHd_output <= 1.01*Math.pow(10,-7)) {
                PFHd_output = 1.01*Math.pow(10,-7);
            }
        } // lo min
        else if (PFHd_output <= 1.01*Math.pow(10,-7) && PFHd_logic >
1.01*Math.pow(10,-7)) {
            PFHd_output = 1.01*Math.pow(10,-7);
            PFHd_logic = PFHr - PFHd_input - PFHd_output;
            if (PFHd_logic <= 1.01*Math.pow(10,-7)) {
                PFHd_logic = 1.01*Math.pow(10,-7);
            }
        } // out min
        else if (PFHd_logic <= 1.01*Math.pow(10,-7) && PFHd_output <=
1.01*Math.pow(10,-7)) {
            PFHd_logic = 1.01*Math.pow(10,-7);
            PFHd_output = 1.01*Math.pow(10,-7);
        } // lo min out min
    } // in max
    else if (PFHd_input <= 1.01*Math.pow(10,-7)) {
        PFHd_input = 1.01*Math.pow(10,-7);
        if (PFHd_logic >= 1.2398*Math.pow(10,-5) && PFHd_output <
1.2398*Math.pow(10,-5)){
            PFHd_logic = 1.2398*Math.pow(10,-5);
            PFHd_output = PFHr - PFHd_input - PFHd_logic;
            if (PFHd_output >= 1.2398*Math.pow(10,-5)) {
                PFHd_output = 1.2398*Math.pow(10,-5);
            }
        } // lo max
        else if (PFHd_output >= 1.2398*Math.pow(10,-5) && PFHd_logic <
1.2398*Math.pow(10,-5)){
            PFHd_output = 1.2398*Math.pow(10,-5);
            PFHd_logic = PFHr - PFHd_input - PFHd_output;
            if (PFHd_logic >= 1.2398*Math.pow(10,-5)) {
                PFHd_logic = 1.2398*Math.pow(10,-5);
            }
        } // out max
        else if (PFHd_logic >= 1.2398*Math.pow(10,-5) && PFHd_output >=
1.2398*Math.pow(10,-5)){

```

```

    PFHd_logic = 1.2398*Math.pow(10,-5);
    PFHd_output = 1.2398*Math.pow(10,-5);
} // lo max out max
} // in min
else {
    Dinput = PFHd0_input - PFHd_input;
    PFHd_logic = PFHd_logic + Dinput/2;
    PFHd_output = PFHd_output + Dinput/2;
    if (PFHd_logic >= 1.2398*Math.pow(10,-5) && PFHd_output <
1.2398*Math.pow(10,-5)) {
        PFHd_logic = 1.2398*Math.pow(10,-5);
        PFHd_output = PFHr - PFHd_input - PFHd_logic;
        if (PFHd_output >= 1.2398*Math.pow(10,-5)) {
            PFHd_output = 1.2398*Math.pow(10,-5);
            PFHd_input = PFHr - PFHd_logic - PFHd_output;
        }
    } // lo max
    else if (PFHd_output >= 1.2398*Math.pow(10,-5) && PFHd_logic <
1.2398*Math.pow(10,-5)) {
        PFHd_output = 1.2398*Math.pow(10,-5);
        PFHd_logic = PFHr - PFHd_input - PFHd_output;
        if (PFHd_logic >= 1.2398*Math.pow(10,-5)) {
            PFHd_logic = 1.2398*Math.pow(10,-5);
            PFHd_input = PFHr - PFHd_logic - PFHd_output;
        }
    } // out max
    else if (PFHd_logic >= 1.2398*Math.pow(10,-5) && PFHd_output >=
1.2398*Math.pow(10,-5)) {
        PFHd_logic = 1.2398*Math.pow(10,-5);
        PFHd_output = 1.2398*Math.pow(10,-5);
        PFHd_input = PFHr - PFHd_logic - PFHd_output;
    } // lo max out max
    else if (PFHd_logic <= 1.01*Math.pow(10,-7) && PFHd_output >
1.01*Math.pow(10,-7)) {
        PFHd_logic = 1.01*Math.pow(10,-7);
        PFHd_output = PFHr - PFHd_input - PFHd_logic;
        if (PFHd_output <= 1.01*Math.pow(10,-7)) {
            PFHd_output = 1.01*Math.pow(10,-7);
            PFHd_input = PFHr - PFHd_logic - PFHd_output;
        }
    } // lo min
    else if (PFHd_output <= 1.01*Math.pow(10,-7) && PFHd_logic >
1.01*Math.pow(10,-7)) {
        PFHd_output = 1.01*Math.pow(10,-7);
        PFHd_logic = PFHr - PFHd_input - PFHd_output;
        if (PFHd_logic <= 1.01*Math.pow(10,-7)) {
            PFHd_logic = 1.01*Math.pow(10,-7);
            PFHd_input = PFHr - PFHd_logic - PFHd_output;
        }
    } // out min
    else if (PFHd_logic <= 1.01*Math.pow(10,-7) && PFHd_output <=
1.01*Math.pow(10,-7)) {
        PFHd_logic = 1.01*Math.pow(10,-7);

```

```

        PFHd_output = 1.01*Math.pow(10,-7);
        PFHd_input = PFHr - PFHd_logic - PFHd_output;
    } // lo min out min
} // else
} // cat 3
if (category.equals("4")) {
    if (PFHd_input >= 4.6*Math.pow(10,-8)){
        PFHd_input = 4.6*Math.pow(10,-8);
        if (PFHd_logic <= 2.47*Math.pow(10,-8) && PFHd_output >
2.47*Math.pow(10,-8)) {
            PFHd_logic = 2.47*Math.pow(10,-8);
            PFHd_output = PFHr - PFHd_input - PFHd_logic;
            if (PFHd_output <= 2.47*Math.pow(10,-8)) {
                PFHd_output = 2.47*Math.pow(10,-8);
            }
        } // lo min
        else if (PFHd_output <= 2.47*Math.pow(10,-8) && PFHd_logic >
2.47*Math.pow(10,-8)) {
            PFHd_output = 2.47*Math.pow(10,-8);
            PFHd_logic = PFHr - PFHd_input - PFHd_output;
            if (PFHd_logic <= 2.47*Math.pow(10,-8)) {
                PFHd_logic = 2.47*Math.pow(10,-8);
            }
        } // out min
        else if (PFHd_logic <= 2.47*Math.pow(10,-8) && PFHd_output <=
2.47*Math.pow(10,-8)) {
            PFHd_logic = 2.47*Math.pow(10,-8);
            PFHd_output = 2.47*Math.pow(10,-8);
        } // lo min out min
    } // in max
    else if (PFHd_input <= 2.47*Math.pow(10,-8)) {
        PFHd_input = 2.47*Math.pow(10,-8);
        Dinput = PFHd0_input - PFHd_input;
        PFHd_logic = PFHd_logic + Dinput/2;
        PFHd_output = PFHd_output + Dinput/2;
        if (PFHd_logic >= 4.6*Math.pow(10,-8) && PFHd_output <
4.6*Math.pow(10,-8)){
            PFHd_logic = 4.6*Math.pow(10,-8);
            PFHd_output = PFHr - PFHd_input - PFHd_logic;
            if (PFHd_output >= 4.6*Math.pow(10,-8)) {
                PFHd_output = 4.6*Math.pow(10,-8);
            }
        } // lo max
        else if (PFHd_output >= 4.6*Math.pow(10,-8) && PFHd_logic <
4.6*Math.pow(10,-8)){
            PFHd_output = 4.6*Math.pow(10,-8);
            PFHd_logic = PFHr - PFHd_input - PFHd_output;
            if (PFHd_logic >= 4.6*Math.pow(10,-8)) {
                PFHd_logic = 4.6*Math.pow(10,-8);
            }
        } // out max
        else if (PFHd_logic >= 4.6*Math.pow(10,-8) && PFHd_output >=
4.6*Math.pow(10,-8)){

```

```

    PFHd_logic = 4.6*Math.pow(10,-8);
    PFHd_output = 4.6*Math.pow(10,-8);
} // lo max out max
} // in min
else {
    Dinput = PFHd0_input - PFHd_input;
    PFHd_logic = PFHd_logic + Dinput/2;
    PFHd_output = PFHd_output + Dinput/2;
    if (PFHd_logic >= 4.6*Math.pow(10,-8) && PFHd_output <
4.6*Math.pow(10,-8)) {
        PFHd_logic = 4.6*Math.pow(10,-8);
        PFHd_output = PFHr - PFHd_input - PFHd_logic;
        if (PFHd_output >= 4.6*Math.pow(10,-8)) {
            PFHd_output = 4.6*Math.pow(10,-8);
            PFHd_input = PFHr - PFHd_logic - PFHd_output;
        }
    } // lo max
    else if (PFHd_output >= 4.6*Math.pow(10,-8) && PFHd_logic <
4.6*Math.pow(10,-8)) {
        PFHd_output = 4.6*Math.pow(10,-8);
        PFHd_logic = PFHr - PFHd_input - PFHd_output;
        if (PFHd_logic >= 4.6*Math.pow(10,-8)) {
            PFHd_logic = 4.6*Math.pow(10,-8);
            PFHd_input = PFHr - PFHd_logic - PFHd_output;
        }
    } // out max
    else if (PFHd_logic >= 4.6*Math.pow(10,-8) && PFHd_output >=
4.6*Math.pow(10,-8)) {
        PFHd_logic = 4.6*Math.pow(10,-8);
        PFHd_output = 4.6*Math.pow(10,-8);
        PFHd_input = PFHr - PFHd_logic - PFHd_output;
    } // lo max out max
    else if (PFHd_logic <= 2.47*Math.pow(10,-8) && PFHd_output >
2.47*Math.pow(10,-8)) {
        PFHd_logic = 2.47*Math.pow(10,-8);
        PFHd_output = PFHr - PFHd_input - PFHd_logic;
        if (PFHd_output <= 2.47*Math.pow(10,-8)) {
            PFHd_output = 2.47*Math.pow(10,-8);
            PFHd_input = PFHr - PFHd_logic - PFHd_output;
        }
    } // lo min
    else if (PFHd_output <= 2.47*Math.pow(10,-8) && PFHd_logic >
2.47*Math.pow(10,-8)) {
        PFHd_output = 2.47*Math.pow(10,-8);
        PFHd_logic = PFHr - PFHd_input - PFHd_output;
        if (PFHd_logic <= 2.47*Math.pow(10,-8)) {
            PFHd_logic = 2.47*Math.pow(10,-8);
            PFHd_input = PFHr - PFHd_logic - PFHd_output;
        }
    } // out min
    else if (PFHd_logic <= 2.47*Math.pow(10,-8) && PFHd_output <=
2.47*Math.pow(10,-8)) {
        PFHd_logic = 2.47*Math.pow(10,-8);

```

```
        PFHd_output = 2.47*Math.pow(10,-8);
        PFHd_input = PFHr - PFHd_logic - PFHd_output;
    } // lo min out min
    } // else
} // cat 4
fix_in = PFHd_input;
fix_lo = PFHd_logic;
fix_out = PFHd_output;
} // in = F, lo = F, out = F
else if (fix_input == true) {
    PFHd_input = fixvalue_in;
    PFHd_logic = PFHd_logic;
    PFHd_output = PFHd_output;
} // in = T
else if (fix_input == false && fix_logic == true && fix_output ==
true) {
    PFHd_input = fix_in;
    PFHd_logic = fixvalue_lo;
    PFHd_output = fixvalue_out;
} // in = F, lo = T, out = T
else if (fix_input == false && fix_logic == false && fix_output ==
true) {
    if (category.equals("B")) {
        if (PFHd_input >= 2.954*Math.pow(10,-5)){
            PFHd_input = 2.954*Math.pow(10,-5);
            PFHd_logic = PFHr - PFHd_input - fixvalue_out;
            if (PFHd_logic >= 2.954*Math.pow(10,-5)) {
                PFHd_logic = 2.954*Math.pow(10,-5);
            }
            else if (PFHd_logic <=4.23*Math.pow(10,-6)) {
                PFHd_logic = 4.23*Math.pow(10,-6);
            }
        }
        else if (PFHd_input <= 4.23*Math.pow(10,-6)){
            PFHd_input = 4.23*Math.pow(10,-6);
            PFHd_logic = PFHr - PFHd_input - fixvalue_out;
            if (PFHd_logic >= 2.954*Math.pow(10,-5)) {
                PFHd_logic = 2.954*Math.pow(10,-5);
            }
            else if (PFHd_logic <=4.23*Math.pow(10,-6)) {
                PFHd_logic = 4.23*Math.pow(10,-6);
            }
        }
    }
    else {
        Dinput = PFHd0_input - PFHd_input;
        PFHd_logic = PFHd_logic + Dinput;
        if (PFHd_logic >= 2.954*Math.pow(10,-5)){
            PFHd_logic = 2.954*Math.pow(10,-5);
            PFHd_input = PFHr - PFHd_logic - fixvalue_out;
        }
        else if (PFHd_logic <= 4.23*Math.pow(10,-6)){
            PFHd_logic = 4.23*Math.pow(10,-6);
            PFHd_input = PFHr - PFHd_logic - fixvalue_out;
        }
    }
}
```

```

    }
  }
  else if (category.equals("1")) {
    if (PFHd_input >= 1.52*Math.pow(10,-6)){
      PFHd_input = 1.52*Math.pow(10,-6);
      PFHd_logic = PFHr - PFHd_input - fixvalue_out;
      if (PFHd_logic >= 1.52*Math.pow(10,-6)){
        PFHd_logic = 1.52*Math.pow(10,-6);
      }
      else if (PFHd_logic <= 1.14*Math.pow(10,-6)){
        PFHd_logic = 1.14*Math.pow(10,-6);
      }
    }
    else if (PFHd_input <= 1.14*Math.pow(10,-6)){
      PFHd_input = 1.14*Math.pow(10,-6);
      PFHd_logic = PFHr - PFHd_input - fixvalue_out;
      if (PFHd_logic >= 1.52*Math.pow(10,-6)){
        PFHd_logic = 1.52*Math.pow(10,-6);
      }
      else if (PFHd_logic <= 1.14*Math.pow(10,-6)){
        PFHd_logic = 1.14*Math.pow(10,-6);
      }
    }
  }
  else {
    Dinput = PFHd0_input - PFHd_input;
    PFHd_logic = PFHd_logic + Dinput;
    if (PFHd_logic >= 1.52*Math.pow(10,-6)){
      PFHd_logic = 1.52*Math.pow(10,-6);
      PFHd_input = PFHr - PFHd_logic - fixvalue_out;
    }
    else if (PFHd_logic <= 1.14*Math.pow(10,-6)){
      PFHd_logic = 1.14*Math.pow(10,-6);
      PFHd_input = PFHr - PFHd_logic - fixvalue_out;
    }
  }
}
else if (category.equals("2")) {
  if (PFHd_input >= 2.4744*Math.pow(10,-5)){
    PFHd_input = 2.4744*Math.pow(10,-5);
    PFHd_logic = PFHr - PFHd_input - fixvalue_out;
    if (PFHd_logic >= 2.4744*Math.pow(10,-5)){
      PFHd_logic = 2.4744*Math.pow(10,-5);
    }
    else if (PFHd_logic <= 5.28*Math.pow(10,-7)){
      PFHd_logic = 5.28*Math.pow(10,-7);
    }
  }
  else if (PFHd_input <= 5.28*Math.pow(10,-7)){
    PFHd_input = 5.28*Math.pow(10,-7);
    PFHd_logic = PFHr - PFHd_input - fixvalue_out;
    if (PFHd_logic >= 2.4744*Math.pow(10,-5)){
      PFHd_logic = 2.4744*Math.pow(10,-5);
    }
  }
}

```



```
    else if (PFHd_logic <= 5.28*Math.pow(10,-7)){
        PFHd_logic = 5.28*Math.pow(10,-7);
    }
}
else {
    Dinput = PFHd0_input - PFHd_input;
    PFHd_logic = PFHd_logic + Dinput;
    if (PFHd_logic >= 2.4744*Math.pow(10,-5)){
        PFHd_logic = 2.4744*Math.pow(10,-5);
        PFHd_input = PFHr - PFHd_logic - fixvalue_out;
    }
    else if (PFHd_logic <= 5.28*Math.pow(10,-7)){
        PFHd_logic = 5.28*Math.pow(10,-7);
        PFHd_input = PFHr - PFHd_logic - fixvalue_out;
    }
}
}
else if (category.equals("3")) {
    if (PFHd_input >= 1.2398*Math.pow(10,-5)){
        PFHd_input = 1.2398*Math.pow(10,-5);
        PFHd_logic = PFHr - PFHd_input - fixvalue_out;
        if (PFHd_logic >= 1.2398*Math.pow(10,-5)){
            PFHd_logic = 1.2398*Math.pow(10,-5);
        }
        else if (PFHd_logic <= 1.01*Math.pow(10,-7)){
            PFHd_logic = 1.01*Math.pow(10,-7);
        }
    }
    else if (PFHd_input <= 1.01*Math.pow(10,-7)){
        PFHd_input = 1.01*Math.pow(10,-7);
        PFHd_logic = PFHr - PFHd_input - fixvalue_out;
        if (PFHd_logic >= 1.2398*Math.pow(10,-5)){
            PFHd_logic = 1.2398*Math.pow(10,-5);
        }
        else if (PFHd_logic <= 1.01*Math.pow(10,-7)){
            PFHd_logic = 1.01*Math.pow(10,-7);
        }
    }
}
else {
    Dinput = PFHd0_input - PFHd_input;
    PFHd_logic = PFHd_logic + Dinput;
    if (PFHd_logic >= 1.2398*Math.pow(10,-5)){
        PFHd_logic = 1.2398*Math.pow(10,-5);
        PFHd_input = PFHr - PFHd_logic - fixvalue_out;
    }
    else if (PFHd_logic <= 1.01*Math.pow(10,-7)){
        PFHd_logic = 1.01*Math.pow(10,-7);
        PFHd_input = PFHr - PFHd_logic - fixvalue_out;
    }
}
}
else if (category.equals("4")) {
    if (PFHd_input >= 4.6*Math.pow(10,-8)){
```

```

    PFHd_input = 4.6*Math.pow(10,-8);
    PFHd_logic = PFHr - PFHd_input - fixvalue_out;
    if (PFHd_logic >= 4.6*Math.pow(10,-8)){
        PFHd_logic = 4.6*Math.pow(10,-8);
    }
    else if (PFHd_logic <= 2.47*Math.pow(10,-8)){
        PFHd_logic = 2.47*Math.pow(10,-8);
    }
}
else if (PFHd_input <= 2.47*Math.pow(10,-8)){
    PFHd_input = 2.47*Math.pow(10,-8);
    PFHd_logic = PFHr - PFHd_input - fixvalue_out;
    if (PFHd_logic >= 4.6*Math.pow(10,-8)){
        PFHd_logic = 4.6*Math.pow(10,-8);
    }
    else if (PFHd_logic <= 2.47*Math.pow(10,-8)){
        PFHd_logic = 2.47*Math.pow(10,-8);
    }
}
}
else {
    Dinput = PFHd0_input - PFHd_input;
    PFHd_logic = PFHd_logic + Dinput;
    if (PFHd_logic >= 4.6*Math.pow(10,-8)){
        PFHd_logic = 4.6*Math.pow(10,-8);
        PFHd_input = PFHr - PFHd_logic - fixvalue_out;
    }
    else if (PFHd_logic <= 2.47*Math.pow(10,-8)){
        PFHd_logic = 2.47*Math.pow(10,-8);
        PFHd_input = PFHr - PFHd_logic - fixvalue_out;
    }
}
}
fix_in = PFHd_input;
fix_lo = PFHd_logic;
fix_out = PFHd_output;
} // in = F, lo = F, out = T
else if (fix_input == false && fix_logic == true && fix_output ==
false) {
    if (category.equals("B")) {
        if (PFHd_input >= 2.954*Math.pow(10,-5)){
            PFHd_input = 2.954*Math.pow(10,-5);
            PFHd_output = PFHr - PFHd_input - fixvalue_lo;
            if (PFHd_output >= 2.954*Math.pow(10,-5)) {
                PFHd_output = 2.954*Math.pow(10,-5);
            }
            else if (PFHd_output <=4.23*Math.pow(10,-6)) {
                PFHd_output = 4.23*Math.pow(10,-6);
            }
        }
    }
    else if (PFHd_input <= 4.23*Math.pow(10,-6)){
        PFHd_input = 4.23*Math.pow(10,-6);
        PFHd_output = PFHr - PFHd_input - fixvalue_lo;
        if (PFHd_output >= 2.954*Math.pow(10,-5)) {

```

```
        PFHd_output = 2.954*Math.pow(10,-5);
    }
    else if (PFHd_output <=4.23*Math.pow(10,-6)) {
        PFHd_output = 4.23*Math.pow(10,-6);
    }
}
else {
    Dinput = PFHd0_input - PFHd_input;
    PFHd_output = PFHd_output + Dinput;
    if (PFHd_output >= 2.954*Math.pow(10,-5)){
        PFHd_output = 2.954*Math.pow(10,-5);
        PFHd_input = PFHr - PFHd_output - fixvalue_lo;
    }
    else if (PFHd_output <= 4.23*Math.pow(10,-6)){
        PFHd_output = 4.23*Math.pow(10,-6);
        PFHd_input = PFHr - PFHd_output - fixvalue_lo;
    }
}
}
else if (category.equals("1")) {
    if (PFHd_input >= 1.52*Math.pow(10,-6)){
        PFHd_input = 1.52*Math.pow(10,-6);
        PFHd_output = PFHr - PFHd_input - fixvalue_lo;
        if (PFHd_output >= 1.52*Math.pow(10,-6)){
            PFHd_output = 1.52*Math.pow(10,-6);
        }
    }
    else if (PFHd_output <= 1.14*Math.pow(10,-6)){
        PFHd_output = 1.14*Math.pow(10,-6);
    }
}
else if (PFHd_input <= 1.14*Math.pow(10,-6)){
    PFHd_input = 1.14*Math.pow(10,-6);
    PFHd_output = PFHr - PFHd_input - fixvalue_lo;
    if (PFHd_output >= 1.52*Math.pow(10,-6)){
        PFHd_output = 1.52*Math.pow(10,-6);
    }
    else if (PFHd_output <= 1.14*Math.pow(10,-6)){
        PFHd_output = 1.14*Math.pow(10,-6);
    }
}
else {
    Dinput = PFHd0_input - PFHd_input;
    PFHd_output = PFHd_output + Dinput;
    if (PFHd_output >= 1.52*Math.pow(10,-6)){
        PFHd_output = 1.52*Math.pow(10,-6);
        PFHd_input = PFHr - PFHd_output - fixvalue_lo;
    }
    else if (PFHd_output <= 1.14*Math.pow(10,-6)){
        PFHd_output = 1.14*Math.pow(10,-6);
        PFHd_input = PFHr - PFHd_output - fixvalue_lo;
    }
}
}
```

```

else if (category.equals("2")) {
    if (PFHd_input >= 2.4744*Math.pow(10,-5)){
        PFHd_input = 2.4744*Math.pow(10,-5);
        PFHd_output = PFHr - PFHd_input - fixvalue_lo;
        if (PFHd_output >= 2.4744*Math.pow(10,-5)){
            PFHd_output = 2.4744*Math.pow(10,-5);
        }
        else if (PFHd_output <= 5.28*Math.pow(10,-7)){
            PFHd_output = 5.28*Math.pow(10,-7);
        }
    }
    else if (PFHd_input <= 5.28*Math.pow(10,-7)){
        PFHd_input = 5.28*Math.pow(10,-7);
        PFHd_output = PFHr - PFHd_input - fixvalue_lo;
        if (PFHd_output >= 2.4744*Math.pow(10,-5)){
            PFHd_output = 2.4744*Math.pow(10,-5);
        }
        else if (PFHd_output <= 5.28*Math.pow(10,-7)){
            PFHd_output = 5.28*Math.pow(10,-7);
        }
    }
    else {
        Dinput = PFHd0_input - PFHd_input;
        PFHd_output = PFHd_output + Dinput;
        if (PFHd_output >= 2.4744*Math.pow(10,-5)){
            PFHd_output = 2.4744*Math.pow(10,-5);
            PFHd_input = PFHr - PFHd_output - fixvalue_lo;
        }
        else if (PFHd_output <= 5.28*Math.pow(10,-7)){
            PFHd_output = 5.28*Math.pow(10,-7);
            PFHd_input = PFHr - PFHd_output - fixvalue_lo;
        }
    }
}
else if (category.equals("3")) {
    if (PFHd_input >= 1.2398*Math.pow(10,-5)){
        PFHd_input = 1.2398*Math.pow(10,-5);
        PFHd_output = PFHr - PFHd_input - fixvalue_lo;
        if (PFHd_output >= 1.2398*Math.pow(10,-5)){
            PFHd_output = 1.2398*Math.pow(10,-5);
        }
        else if (PFHd_output <= 1.01*Math.pow(10,-7)){
            PFHd_output = 1.01*Math.pow(10,-7);
        }
    }
    else if (PFHd_input <= 1.01*Math.pow(10,-7)){
        PFHd_input = 1.01*Math.pow(10,-7);
        PFHd_output = PFHr - PFHd_input - fixvalue_lo;
        if (PFHd_output >= 1.2398*Math.pow(10,-5)){
            PFHd_output = 1.2398*Math.pow(10,-5);
        }
        else if (PFHd_output <= 1.01*Math.pow(10,-7)){
            PFHd_output = 1.01*Math.pow(10,-7);
        }
    }
}

```

```

    }
  }
  else {
    Dinput = PFHd0_input - PFHd_input;
    PFHd_output = PFHd_output + Dinput;
    if (PFHd_output >= 1.2398*Math.pow(10,-5)){
      PFHd_output = 1.2398*Math.pow(10,-5);
      PFHd_input = PFHr - PFHd_output - fixvalue_lo;
    }
    else if (PFHd_output <= 1.01*Math.pow(10,-7)){
      PFHd_output = 1.01*Math.pow(10,-7);
      PFHd_input = PFHr - PFHd_output - fixvalue_lo;
    }
  }
}
else if (category.equals("4")) {
  if (PFHd_input >= 4.6*Math.pow(10,-8)){
    PFHd_input = 4.6*Math.pow(10,-8);
    PFHd_output = PFHr - PFHd_input - fixvalue_lo;
    if (PFHd_output >= 4.6*Math.pow(10,-8)){
      PFHd_output = 4.6*Math.pow(10,-8);
    }
    else if (PFHd_output <= 2.47*Math.pow(10,-8)){
      PFHd_output = 2.47*Math.pow(10,-8);
    }
  }
  else if (PFHd_input <= 2.47*Math.pow(10,-8)){
    PFHd_input = 2.47*Math.pow(10,-8);
    PFHd_output = PFHr - PFHd_input - fixvalue_lo;
    if (PFHd_output >= 4.6*Math.pow(10,-8)){
      PFHd_output = 4.6*Math.pow(10,-8);
    }
    else if (PFHd_output <= 2.47*Math.pow(10,-8)){
      PFHd_output = 2.47*Math.pow(10,-8);
    }
  }
}
else {
  Dinput = PFHd0_input - PFHd_input;
  PFHd_output = PFHd_output + Dinput;
  if (PFHd_output >= 4.6*Math.pow(10,-8)){
    PFHd_output = 4.6*Math.pow(10,-8);
    PFHd_input = PFHr - PFHd_output - fixvalue_lo;
  }
  else if (PFHd_output <= 2.47*Math.pow(10,-8)){
    PFHd_output = 2.47*Math.pow(10,-8);
    PFHd_input = PFHr - PFHd_output - fixvalue_lo;
  }
}
}
fix_in = PFHd_input;
fix_lo = PFHd_logic;
fix_out = PFHd_output;
} // in = F, lo = T, out = F

```

```

// Initial values.
PFHd0_input = PFHd_input;
PFHd0_logic = PFHd_logic;
PFHd0_output = PFHd_output;

// Determination of MTTFd.
if (category.equals("B")) {
    MTTFd_input = 1.14*0.0001*Math.pow(PFHd_input,-1.0002);
    MTTFd_logic = 1.14*0.0001*Math.pow(PFHd_logic,-1.0002);
    MTTFd_output = 1.14*0.0001*Math.pow(PFHd_output,-1.0002);
}
else if (category.equals("1")) {
    MTTFd_input = 1.16*0.0001*Math.pow(PFHd_input,-0.9985);
    MTTFd_logic = 1.16*0.0001*Math.pow(PFHd_logic,-0.9985);
    MTTFd_output = 1.16*0.0001*Math.pow(PFHd_output,-0.9985);
}
else if (category.equals("2")) {
    MTTFd_input = 0.0002*Math.pow(PFHd_input,-0.8975);
    MTTFd_logic = 0.0002*Math.pow(PFHd_logic,-0.8975);
    MTTFd_output = 0.0002*Math.pow(PFHd_output,-0.8975);
}
else if (category.equals("3")) {
    MTTFd_input = 0.001*Math.pow(PFHd_input,-0.7191);
    MTTFd_logic = 0.001*Math.pow(PFHd_logic,-0.7191);
    MTTFd_output = 0.001*Math.pow(PFHd_output,-0.7191);
}
else if (category.equals("4")) {
    MTTFd_input = 0.85*0.00002*Math.pow(PFHd_input,-0.89);
    MTTFd_logic = 0.85*0.00002*Math.pow(PFHd_logic,-0.89);
    MTTFd_output = 0.85*0.00002*Math.pow(PFHd_output,-0.89);
}

// Determination of B10d.
B10d_input = 0.1*MTTFd_input*nop_input;
B10d_logic = 0.1*MTTFd_logic*nop_logic;
B10d_output = 0.1*MTTFd_output*nop_output;

// Determination of PL_input.
if (PFHd_input > 1*Math.pow(10,-5)){
    PL_input = "a";
}
else if (PFHd_input <= 1*Math.pow(10,-5) && PFHd_input >
3*Math.pow(10,-6)){
    PL_input = "b";
}
else if (PFHd_input <= 3*Math.pow(10,-6) && PFHd_input >
1*Math.pow(10,-6)){
    PL_input = "c";
}
else if (PFHd_input <= 1*Math.pow(10,-6) && PFHd_input >
1*Math.pow(10,-7)){
    PL_input = "d";
}

```

```
}
else if (PFHd_input <= 1*Math.pow(10,-7)){
    PL_logic = "e";
}

// Determiantion of PL_logic.
if (PFHd_logic > 1*Math.pow(10,-5)){
    PL_logic = "a";
}
else if (PFHd_logic <= 1*Math.pow(10,-5) && PFHd_logic >
3*Math.pow(10,-6)){
    PL_logic = "b";
}
else if (PFHd_logic <= 3*Math.pow(10,-6) && PFHd_logic >
1*Math.pow(10,-6)){
    PL_logic = "c";
}
else if (PFHd_logic <= 1*Math.pow(10,-6) && PFHd_logic >
1*Math.pow(10,-7)){
    PL_logic = "d";
}
else if (PFHd_logic <= 1*Math.pow(10,-7)){
    PL_logic = "e";
}

// Determination of PL_output.
if (PFHd_output > 1*Math.pow(10,-5)){
    PL_output = "a";
}
else if (PFHd_output <= 1*Math.pow(10,-5) && PFHd_output >
3*Math.pow(10,-6)){
    PL_output = "b";
}
else if (PFHd_output <= 3*Math.pow(10,-6) && PFHd_output >
1*Math.pow(10,-6)){
    PL_output = "c";
}
else if (PFHd_output <= 1*Math.pow(10,-6) && PFHd_output >
1*Math.pow(10,-7)){
    PL_output = "d";
}
else if (PFHd_output <= 1*Math.pow(10,-7)){
    PL_output = "e";
}
}
```

B12.2. Unidad lógica.

```
// LOGIC
// Determination of PFHd.
if (fix_input == false && fix_logic == false && fix_output == false) {
    if (category.equals("B")) {
```

```

if (PFHd_logic >= 2.954*Math.pow(10,-5)){
    PFHd_logic = 2.954*Math.pow(10,-5);
    if (PFHd_input <= 4.23*Math.pow(10,-6) && PFHd_output >
4.23*Math.pow(10,-6)) {
        PFHd_input = 4.23*Math.pow(10,-6);
        PFHd_output = PFHr - PFHd_input - PFHd_logic;
        if (PFHd_output <= 4.23*Math.pow(10,-6)) {
            PFHd_output = 4.23*Math.pow(10,-6);
        }
    } // in min
    else if (PFHd_output <= 4.23*Math.pow(10,-6) && PFHd_input >
4.23*Math.pow(10,-6)) {
        PFHd_output = 4.23*Math.pow(10,-6);
        PFHd_input = PFHr - PFHd_logic - PFHd_output;
        if (PFHd_input <= 4.23*Math.pow(10,-6)) {
            PFHd_input = 4.23*Math.pow(10,-6);
        }
    } // out min
    else if (PFHd_input <= 4.23*Math.pow(10,-6) && PFHd_output <=
4.23*Math.pow(10,-6)) {
        PFHd_input = 4.23*Math.pow(10,-6);
        PFHd_output = 4.23*Math.pow(10,-6);
    } // in min out min
} // lo max
else if (PFHd_logic <= 4.23*Math.pow(10,-6)) {
    PFHd_logic = 4.23*Math.pow(10,-6);
    if (PFHd_input >= 2.954*Math.pow(10,-5) && PFHd_output <
2.954*Math.pow(10,-5)){
        PFHd_input = 2.954*Math.pow(10,-5);
        PFHd_output = PFHr - PFHd_input - PFHd_logic;
        if (PFHd_output >= 2.954*Math.pow(10,-5)) {
            PFHd_output = 2.954*Math.pow(10,-5);
        }
    } // in max
    else if (PFHd_output >= 2.954*Math.pow(10,-5) && PFHd_input <
2.954*Math.pow(10,-5)){
        PFHd_output = 2.954*Math.pow(10,-5);
        PFHd_input = PFHr - PFHd_logic - PFHd_output;
        if (PFHd_input >= 2.954*Math.pow(10,-5)) {
            PFHd_input = 2.954*Math.pow(10,-5);
        }
    } // Out max
    else if (PFHd_input >= 2.954*Math.pow(10,-5) && PFHd_output >=
2.954*Math.pow(10,-5)){
        PFHd_input = 2.954*Math.pow(10,-5);
        PFHd_output = 2.954*Math.pow(10,-5);
    } // in max out max
} // ilo min
else {
    Dlogic = PFHd0_logic - PFHd_logic;
    PFHd_input = PFHd_input + Dlogic/2;
    PFHd_output = PFHd_output + Dlogic/2;
    if (PFHd_input >= 2.954*Math.pow(10,-5) && PFHd_output <

```



```

2.954*Math.pow(10,-5)) {
    PFHd_input = 2.954*Math.pow(10,-5);
    PFHd_output = PFHr - PFHd_input - PFHd_logic;
    if (PFHd_output >= 2.954*Math.pow(10,-5)) {
        PFHd_output = 2.954*Math.pow(10,-5);
        PFHd_logic = PFHr - PFHd_input - PFHd_output;
    }
} // in max
else if (PFHd_output >= 2.954*Math.pow(10,-5) && PFHd_input <
2.954*Math.pow(10,-5)) {
    PFHd_output = 2.954*Math.pow(10,-5);
    PFHd_input = PFHr - PFHd_logic - PFHd_output;
    if (PFHd_input >= 2.954*Math.pow(10,-5)) {
        PFHd_input = 2.954*Math.pow(10,-5);
        PFHd_logic = PFHr - PFHd_input - PFHd_output;
    }
} // out max
else if (PFHd_input >= 2.954*Math.pow(10,-5) && PFHd_output >=
2.954*Math.pow(10,-5)) {
    PFHd_input = 2.954*Math.pow(10,-5);
    PFHd_output = 2.954*Math.pow(10,-5);
    PFHd_logic = PFHr - PFHd_input - PFHd_output;
} // in max out max
else if (PFHd_input <= 4.23*Math.pow(10,-6) && PFHd_output >
4.23*Math.pow(10,-6)) {
    PFHd_input = 4.23*Math.pow(10,-6);
    PFHd_output = PFHr - PFHd_input - PFHd_logic;
    if (PFHd_output <= 4.23*Math.pow(10,-6)) {
        PFHd_output = 4.23*Math.pow(10,-6);
        PFHd_logic = PFHr - PFHd_input - PFHd_output;
    }
} // in min
else if (PFHd_output <= 4.23*Math.pow(10,-6) && PFHd_input >
4.23*Math.pow(10,-6)) {
    PFHd_output = 4.23*Math.pow(10,-6);
    PFHd_input = PFHr - PFHd_logic - PFHd_output;
    if (PFHd_input <= 4.23*Math.pow(10,-6)) {
        PFHd_input = 4.23*Math.pow(10,-6);
        PFHd_logic = PFHr - PFHd_input - PFHd_output;
    }
} // out min
else if (PFHd_input <= 4.23*Math.pow(10,-6) && PFHd_output <=
4.23*Math.pow(10,-6)) {
    PFHd_input = 4.23*Math.pow(10,-6);
    PFHd_output = 4.23*Math.pow(10,-6);
    PFHd_logic = PFHr - PFHd_input - PFHd_output;
} // in min out min
} // else
} // cat B
if (category.equals("1")) {
    if (PFHd_logic >= 1.52*Math.pow(10,-6)) {
        PFHd_logic = 1.52*Math.pow(10,-6);
        if (PFHd_input <= 1.14*Math.pow(10,-6) && PFHd_output >

```

```

1.14*Math.pow(10,-6)) {
    PFHd_input = 1.14*Math.pow(10,-6);
    PFHd_output = PFHr - PFHd_input - PFHd_logic;
    if (PFHd_output <= 1.14*Math.pow(10,-6)) {
        PFHd_output = 1.14*Math.pow(10,-6);
    }
    } // in min
    else if (PFHd_output <= 1.14*Math.pow(10,-6) && PFHd_input >
1.14*Math.pow(10,-6)) {
        PFHd_output = 1.14*Math.pow(10,-6);
        PFHd_input = PFHr - PFHd_logic - PFHd_output;
        if (PFHd_input <= 1.14*Math.pow(10,-6)) {
            PFHd_input = 1.14*Math.pow(10,-6);
        }
    } // out min
    else if (PFHd_input <= 1.14*Math.pow(10,-6) && PFHd_output <=
1.14*Math.pow(10,-6)) {
        PFHd_input = 1.14*Math.pow(10,-6);
        PFHd_output = 1.14*Math.pow(10,-6);
    } // in min out min
    } // lo max
    else if (PFHd_logic <= 1.14*Math.pow(10,-6)) {
        PFHd_logic = 1.14*Math.pow(10,-6);
        if (PFHd_input >= 1.52*Math.pow(10,-6) && PFHd_output <
1.52*Math.pow(10,-6)) {
            PFHd_input = 1.52*Math.pow(10,-6);
            PFHd_output = PFHr - PFHd_input - PFHd_logic;
            if (PFHd_output >= 1.52*Math.pow(10,-6)) {
                PFHd_output = 1.52*Math.pow(10,-6);
            }
        } // in max
        else if (PFHd_output >= 1.52*Math.pow(10,-6) && PFHd_input <
1.52*Math.pow(10,-6)) {
            PFHd_output = 1.52*Math.pow(10,-6);
            PFHd_input = PFHr - PFHd_logic - PFHd_output;
            if (PFHd_input >= 1.52*Math.pow(10,-6)) {
                PFHd_input = 1.52*Math.pow(10,-6);
            }
        } // Out max
        else if (PFHd_input >= 1.52*Math.pow(10,-6) && PFHd_output >=
1.52*Math.pow(10,-6)) {
            PFHd_input = 1.52*Math.pow(10,-6);
            PFHd_output = 1.52*Math.pow(10,-6);
        } // in max out max
    } // lo min
    else {
        Dlogic = PFHd0_logic - PFHd_logic;
        PFHd_input = PFHd_input + Dlogic/2;
        PFHd_output = PFHd_output + Dlogic/2;
        if (PFHd_input >= 1.52*Math.pow(10,-6) && PFHd_output <
1.52*Math.pow(10,-6)) {
            PFHd_input = 1.52*Math.pow(10,-6);
            PFHd_output = PFHr - PFHd_input - PFHd_logic;

```

```
        if (PFHd_output >= 1.52*Math.pow(10,-6)) {
            PFHd_output = 1.52*Math.pow(10,-6);
            PFHd_logic = PFHr - PFHd_input - PFHd_output;
        }
    } // in max
    else if (PFHd_output >= 1.52*Math.pow(10,-6) && PFHd_input <
1.52*Math.pow(10,-6)) {
        PFHd_output = 1.52*Math.pow(10,-6);
        PFHd_input = PFHr - PFHd_logic - PFHd_output;
        if (PFHd_input >= 1.52*Math.pow(10,-6)) {
            PFHd_input = 1.52*Math.pow(10,-6);
            PFHd_logic = PFHr - PFHd_input - PFHd_output;
        }
    } // out max
    else if (PFHd_input >= 1.52*Math.pow(10,-6) && PFHd_output >=
1.52*Math.pow(10,-6)) {
        PFHd_input = 1.52*Math.pow(10,-6);
        PFHd_output = 1.52*Math.pow(10,-6);
        PFHd_logic = PFHr - PFHd_input - PFHd_output;
    } // in max out max
    else if (PFHd_input <= 1.14*Math.pow(10,-6) && PFHd_output >
1.14*Math.pow(10,-6)) {
        PFHd_input = 1.14*Math.pow(10,-6);
        PFHd_output = PFHr - PFHd_input - PFHd_logic;
        if (PFHd_output <= 1.14*Math.pow(10,-6)) {
            PFHd_output = 1.14*Math.pow(10,-6);
            PFHd_logic = PFHr - PFHd_input - PFHd_output;
        }
    } // in min
    else if (PFHd_output <= 1.14*Math.pow(10,-6) && PFHd_input >
1.14*Math.pow(10,-6)) {
        PFHd_output = 1.14*Math.pow(10,-6);
        PFHd_input = PFHr - PFHd_logic - PFHd_output;
        if (PFHd_input <= 1.14*Math.pow(10,-6)) {
            PFHd_input = 1.14*Math.pow(10,-6);
            PFHd_logic = PFHr - PFHd_input - PFHd_output;
        }
    } // out min
    else if (PFHd_input <= 1.14*Math.pow(10,-6) && PFHd_output <=
1.14*Math.pow(10,-6)) {
        PFHd_input = 1.14*Math.pow(10,-6);
        PFHd_output = 1.14*Math.pow(10,-6);
        PFHd_logic = PFHr - PFHd_input - PFHd_output;
    } // in min out min
    } // else
} // cat 1
if (category.equals("2")) {
    if (PFHd_logic >= 2.4744*Math.pow(10,-5)) {
        PFHd_logic = 2.4744*Math.pow(10,-5);
        if (PFHd_input <= 5.28*Math.pow(10,-7) && PFHd_output >
5.28*Math.pow(10,-7)) {
            PFHd_input = 5.28*Math.pow(10,-7);
            PFHd_output = PFHr - PFHd_input - PFHd_logic;
        }
    }
}
```

```

        if (PFHd_output <= 5.28*Math.pow(10,-7)) {
            PFHd_output = 5.28*Math.pow(10,-7);
        }
    } // in min
    else if (PFHd_output <= 5.28*Math.pow(10,-7) && PFHd_input >
5.28*Math.pow(10,-7)) {
        PFHd_output = 5.28*Math.pow(10,-7);
        PFHd_input = PFHr - PFHd_logic - PFHd_output;
        if (PFHd_input <= 5.28*Math.pow(10,-7)) {
            PFHd_input = 5.28*Math.pow(10,-7);
        }
    } // out min
    else if (PFHd_input <= 5.28*Math.pow(10,-7) && PFHd_output <=
5.28*Math.pow(10,-7)) {
        PFHd_input = 5.28*Math.pow(10,-7);
        PFHd_output = 5.28*Math.pow(10,-7);
    } // in min out min
} // lo max
else if (PFHd_logic <= 5.28*Math.pow(10,-7)) {
    PFHd_logic = 5.28*Math.pow(10,-7);
    if (PFHd_input >= 2.4744*Math.pow(10,-5) && PFHd_output <
2.4744*Math.pow(10,-5)) {
        PFHd_input = 2.4744*Math.pow(10,-5);
        PFHd_output = PFHr - PFHd_input - PFHd_logic;
        if (PFHd_output >= 2.4744*Math.pow(10,-5)) {
            PFHd_output = 2.4744*Math.pow(10,-5);
        }
    } // in max
    else if (PFHd_output >= 2.4744*Math.pow(10,-5) && PFHd_input <
2.4744*Math.pow(10,-5)) {
        PFHd_output = 2.4744*Math.pow(10,-5);
        PFHd_input = PFHr - PFHd_logic - PFHd_output;
        if (PFHd_input >= 2.4744*Math.pow(10,-5)) {
            PFHd_input = 2.4744*Math.pow(10,-5);
        }
    } // Out max
    else if (PFHd_input >= 2.4744*Math.pow(10,-5) && PFHd_output >=
2.4744*Math.pow(10,-5)) {
        PFHd_input = 2.4744*Math.pow(10,-5);
        PFHd_output = 2.4744*Math.pow(10,-5);
    } // in max out max
} // lo min
else {
    Dlogic = PFHd0_logic - PFHd_logic;
    PFHd_input = PFHd_input + Dlogic/2;
    PFHd_output = PFHd_output + Dlogic/2;
    if (PFHd_input >= 2.4744*Math.pow(10,-5) && PFHd_output <
2.4744*Math.pow(10,-5)) {
        PFHd_input = 2.4744*Math.pow(10,-5);
        PFHd_output = PFHr - PFHd_input - PFHd_logic;
        if (PFHd_output >= 2.4744*Math.pow(10,-5)) {
            PFHd_output = 2.4744*Math.pow(10,-5);
            PFHd_logic = PFHr - PFHd_input - PFHd_output;
        }
    }
}

```

```

    }
    } // in max
    else if (PFHd_output >= 2.4744*Math.pow(10,-5) && PFHd_input <
2.4744*Math.pow(10,-5)) {
        PFHd_output = 2.4744*Math.pow(10,-5);
        PFHd_input = PFHr - PFHd_logic - PFHd_output;
        if (PFHd_input >= 2.4744*Math.pow(10,-5)) {
            PFHd_input = 2.4744*Math.pow(10,-5);
            PFHd_logic = PFHr - PFHd_input - PFHd_output;
        }
    } // out max
    else if (PFHd_input >= 2.4744*Math.pow(10,-5) && PFHd_output >=
2.4744*Math.pow(10,-5)) {
        PFHd_input = 2.4744*Math.pow(10,-5);
        PFHd_output = 2.4744*Math.pow(10,-5);
        PFHd_logic = PFHr - PFHd_input - PFHd_output;
    } // in max out max
    else if (PFHd_input <= 5.28*Math.pow(10,-7) && PFHd_output >
5.28*Math.pow(10,-7)) {
        PFHd_input = 5.28*Math.pow(10,-7);
        PFHd_output = PFHr - PFHd_input - PFHd_logic;
        if (PFHd_output <= 5.28*Math.pow(10,-7)) {
            PFHd_output = 5.28*Math.pow(10,-7);
            PFHd_logic = PFHr - PFHd_input - PFHd_output;
        }
    } // in min
    else if (PFHd_output <= 5.28*Math.pow(10,-7) && PFHd_input >
5.28*Math.pow(10,-7)) {
        PFHd_output = 5.28*Math.pow(10,-7);
        PFHd_input = PFHr - PFHd_logic - PFHd_output;
        if (PFHd_input <= 5.28*Math.pow(10,-7)) {
            PFHd_input = 5.28*Math.pow(10,-7);
            PFHd_logic = PFHr - PFHd_input - PFHd_output;
        }
    } // out min
    else if (PFHd_input <= 5.28*Math.pow(10,-7) && PFHd_output <=
5.28*Math.pow(10,-7)) {
        PFHd_input = 5.28*Math.pow(10,-7);
        PFHd_output = 5.28*Math.pow(10,-7);
        PFHd_logic = PFHr - PFHd_input - PFHd_output;
    } // in min out min
    } // else
} // cat 2
if (category.equals("3")) {
    if (PFHd_logic >= 1.2398*Math.pow(10,-5)) {
        PFHd_logic = 1.2398*Math.pow(10,-5);
        if (PFHd_input <= 1.01*Math.pow(10,-7) && PFHd_output >
1.01*Math.pow(10,-7)) {
            PFHd_input = 1.01*Math.pow(10,-7);
            PFHd_output = PFHr - PFHd_input - PFHd_logic;
            if (PFHd_output <= 1.01*Math.pow(10,-7)) {
                PFHd_output = 1.01*Math.pow(10,-7);
            }
        }
    }
}

```

```

    } // in min
    else if (PFHd_output <= 1.01*Math.pow(10,-7) && PFHd_input >
1.01*Math.pow(10,-7)) {
        PFHd_output = 1.01*Math.pow(10,-7);
        PFHd_input = PFHr - PFHd_logic - PFHd_output;
        if (PFHd_input <= 1.01*Math.pow(10,-7)) {
            PFHd_input = 1.01*Math.pow(10,-7);
        }
    } // out min
    else if (PFHd_input <= 1.01*Math.pow(10,-7) && PFHd_output <=
1.01*Math.pow(10,-7)) {
        PFHd_input = 1.01*Math.pow(10,-7);
        PFHd_output = 1.01*Math.pow(10,-7);
    } // in min out min
} // lo max
else if (PFHd_logic <= 1.01*Math.pow(10,-7)) {
    PFHd_logic = 1.01*Math.pow(10,-7);
    if (PFHd_input >= 1.2398*Math.pow(10,-5) && PFHd_output <
1.2398*Math.pow(10,-5)) {
        PFHd_input = 1.2398*Math.pow(10,-5);
        PFHd_output = PFHr - PFHd_input - PFHd_logic;
        if (PFHd_output >= 1.2398*Math.pow(10,-5)) {
            PFHd_output = 1.2398*Math.pow(10,-5);
        }
    } // in max
    else if (PFHd_output >= 1.2398*Math.pow(10,-5) && PFHd_input <
1.2398*Math.pow(10,-5)) {
        PFHd_output = 1.2398*Math.pow(10,-5);
        PFHd_input = PFHr - PFHd_logic - PFHd_output;
        if (PFHd_input >= 1.2398*Math.pow(10,-5)) {
            PFHd_input = 1.2398*Math.pow(10,-5);
        }
    } // Out max
    else if (PFHd_input >= 1.2398*Math.pow(10,-5) && PFHd_output >=
1.2398*Math.pow(10,-5)) {
        PFHd_input = 1.2398*Math.pow(10,-5);
        PFHd_output = 1.2398*Math.pow(10,-5);
    } // in max out max
} // lo min
else {
    Dlogic = PFHd0_logic - PFHd_logic;
    PFHd_input = PFHd_input + Dlogic/2;
    PFHd_output = PFHd_output + Dlogic/2;
    if (PFHd_input >= 1.2398*Math.pow(10,-5) && PFHd_output <
1.2398*Math.pow(10,-5)) {
        PFHd_input = 1.2398*Math.pow(10,-5);
        PFHd_output = PFHr - PFHd_input - PFHd_logic;
        if (PFHd_output >= 1.2398*Math.pow(10,-5)) {
            PFHd_output = 1.2398*Math.pow(10,-5);
            PFHd_logic = PFHr - PFHd_input - PFHd_output;
        }
    } // in max
    else if (PFHd_output >= 1.2398*Math.pow(10,-5) && PFHd_input <

```

```

1.2398*Math.pow(10,-5)) {
    PFHd_output = 1.2398*Math.pow(10,-5);
    PFHd_input = PFHr - PFHd_logic - PFHd_output;
    if (PFHd_input >= 1.2398*Math.pow(10,-5)) {
        PFHd_input = 1.2398*Math.pow(10,-5);
        PFHd_logic = PFHr - PFHd_input - PFHd_output;
    }
} // out max
else if (PFHd_input >= 1.2398*Math.pow(10,-5) && PFHd_output >=
1.2398*Math.pow(10,-5)) {
    PFHd_input = 1.2398*Math.pow(10,-5);
    PFHd_output = 1.2398*Math.pow(10,-5);
    PFHd_logic = PFHr - PFHd_input - PFHd_output;
} // in max out max
else if (PFHd_input <= 1.01*Math.pow(10,-7) && PFHd_output >
1.01*Math.pow(10,-7)) {
    PFHd_input = 1.01*Math.pow(10,-7);
    PFHd_output = PFHr - PFHd_input - PFHd_logic;
    if (PFHd_output <= 1.01*Math.pow(10,-7)) {
        PFHd_output = 1.01*Math.pow(10,-7);
        PFHd_logic = PFHr - PFHd_input - PFHd_output;
    }
} // in min
else if (PFHd_output <= 1.01*Math.pow(10,-7) && PFHd_input >
1.01*Math.pow(10,-7)) {
    PFHd_output = 1.01*Math.pow(10,-7);
    PFHd_input = PFHr - PFHd_logic - PFHd_output;
    if (PFHd_input <= 1.01*Math.pow(10,-7)) {
        PFHd_input = 1.01*Math.pow(10,-7);
        PFHd_logic = PFHr - PFHd_input - PFHd_output;
    }
} // out min
else if (PFHd_input <= 1.01*Math.pow(10,-7) && PFHd_output <=
1.01*Math.pow(10,-7)) {
    PFHd_input = 1.01*Math.pow(10,-7);
    PFHd_output = 1.01*Math.pow(10,-7);
    PFHd_logic = PFHr - PFHd_input - PFHd_output;
} // in min out min
} // else
} // cat 3
if (category.equals("4")) {
    if (PFHd_logic >= 4.6*Math.pow(10,-8)){
        PFHd_logic = 4.6*Math.pow(10,-8);
        if (PFHd_input <= 2.47*Math.pow(10,-8) && PFHd_output >
2.47*Math.pow(10,-8)) {
            PFHd_input = 2.47*Math.pow(10,-8);
            PFHd_output = PFHr - PFHd_input - PFHd_logic;
            if (PFHd_output <= 2.47*Math.pow(10,-8)) {
                PFHd_output = 2.47*Math.pow(10,-8);
            }
        }
    } // in min
    else if (PFHd_output <= 2.47*Math.pow(10,-8) && PFHd_input >
2.47*Math.pow(10,-8)) {

```

```

    PFHd_output = 2.47*Math.pow(10,-8);
    PFHd_input = PFHr - PFHd_logic - PFHd_output;
    if (PFHd_input <= 2.47*Math.pow(10,-8)) {
        PFHd_input = 2.47*Math.pow(10,-8);
    }
    } // out min
    else if (PFHd_input <= 2.47*Math.pow(10,-8) && PFHd_output <=
2.47*Math.pow(10,-8)) {
        PFHd_input = 2.47*Math.pow(10,-8);
        PFHd_output = 2.47*Math.pow(10,-8);
    } // in min out min
    } // lo max
    else if (PFHd_logic <= 2.47*Math.pow(10,-8)) {
        PFHd_logic = 2.47*Math.pow(10,-8);
        if (PFHd_input >= 4.6*Math.pow(10,-8) && PFHd_output <
4.6*Math.pow(10,-8)){
            PFHd_input = 4.6*Math.pow(10,-8);
            PFHd_output = PFHr - PFHd_input - PFHd_logic;
            if (PFHd_output >= 4.6*Math.pow(10,-8)) {
                PFHd_output = 4.6*Math.pow(10,-8);
            }
        } // in max
        else if (PFHd_output >= 4.6*Math.pow(10,-8) && PFHd_input <
4.6*Math.pow(10,-8)){
            PFHd_output = 4.6*Math.pow(10,-8);
            PFHd_input = PFHr - PFHd_logic - PFHd_output;
            if (PFHd_input >= 4.6*Math.pow(10,-8)) {
                PFHd_input = 4.6*Math.pow(10,-8);
            }
        } // out max
        else if (PFHd_input >= 4.6*Math.pow(10,-8) && PFHd_output >=
4.6*Math.pow(10,-8)){
            PFHd_input = 4.6*Math.pow(10,-8);
            PFHd_output = 4.6*Math.pow(10,-8);
        } // in max out max
    } // lo min
    else {
        Dlogic = PFHd0_logic - PFHd_logic;
        PFHd_input = PFHd_input + Dlogic/2;
        PFHd_output = PFHd_output + Dlogic/2;
        if (PFHd_input >= 4.6*Math.pow(10,-8) && PFHd_output <
4.6*Math.pow(10,-8)) {
            PFHd_input = 4.6*Math.pow(10,-8);
            PFHd_output = PFHr - PFHd_input - PFHd_logic;
            if (PFHd_output >= 4.6*Math.pow(10,-8)) {
                PFHd_output = 4.6*Math.pow(10,-8);
                PFHd_logic = PFHr - PFHd_input - PFHd_output;
            }
        } // in max
        else if (PFHd_output >= 4.6*Math.pow(10,-8) && PFHd_input <
4.6*Math.pow(10,-8)) {
            PFHd_output = 4.6*Math.pow(10,-8);
            PFHd_input = PFHr - PFHd_logic - PFHd_output;

```



```
        if (PFHd_input >= 4.6*Math.pow(10,-8)) {
            PFHd_input = 4.6*Math.pow(10,-8);
            PFHd_logic = PFHr - PFHd_input - PFHd_output;
        }
    } // out max
    else if (PFHd_input >= 4.6*Math.pow(10,-8) && PFHd_output >=
4.6*Math.pow(10,-8)) {
        PFHd_input = 4.6*Math.pow(10,-8);
        PFHd_output = 4.6*Math.pow(10,-8);
        PFHd_logic = PFHr - PFHd_input - PFHd_output;
    } // in max out max
    else if (PFHd_input <= 2.47*Math.pow(10,-8) && PFHd_output >
2.47*Math.pow(10,-8)) {
        PFHd_input = 2.47*Math.pow(10,-8);
        PFHd_output = PFHr - PFHd_input - PFHd_logic;
        if (PFHd_output <= 2.47*Math.pow(10,-8)) {
            PFHd_output = 2.47*Math.pow(10,-8);
            PFHd_logic = PFHr - PFHd_input - PFHd_output;
        }
    } // in min
    else if (PFHd_output <= 2.47*Math.pow(10,-8) && PFHd_input >
2.47*Math.pow(10,-8)) {
        PFHd_output = 2.47*Math.pow(10,-8);
        PFHd_input = PFHr - PFHd_logic - PFHd_output;
        if (PFHd_input <= 2.47*Math.pow(10,-8)) {
            PFHd_input = 2.47*Math.pow(10,-8);
            PFHd_logic = PFHr - PFHd_input - PFHd_output;
        }
    } // out min
    else if (PFHd_input <= 2.47*Math.pow(10,-8) && PFHd_output <=
2.47*Math.pow(10,-8)) {
        PFHd_input = 2.47*Math.pow(10,-8);
        PFHd_output = 2.47*Math.pow(10,-8);
        PFHd_logic = PFHr - PFHd_input - PFHd_output;
    } // in min out min
    } // else
} // cat 4
fix_in = PFHd_input;
fix_lo = PFHd_logic;
fix_out = PFHd_output;
} // in = F, lo = F, out = F
else if (fix_logic == true){
    PFHd_input = PFHd_input;
    PFHd_logic = fixvalue_lo;
    PFHd_output = PFHd_output;
}
else if (fix_input == true && fix_logic == false && fix_output ==
true) {
    PFHd_input = fixvalue_in;
    PFHd_logic = fix_lo;
    PFHd_output = fixvalue_out;
}
else if (fix_input == true && fix_logic == false && fix_output ==
```

```

false) {
  if (category.equals("B")) {
    if (PFHd_logic >= 2.954*Math.pow(10,-5)){
      PFHd_logic = 2.954*Math.pow(10,-5);
      PFHd_output = PFHr - PFHd_logic - fixvalue_in;
      if (PFHd_output >= 2.954*Math.pow(10,-5)){
        PFHd_output = 2.954*Math.pow(10,-5);
      }
    }
    else if (PFHd_output <= 4.23*Math.pow(10,-6)){
      PFHd_output = 4.23*Math.pow(10,-6);
    }
  }
  else if (PFHd_logic <= 4.23*Math.pow(10,-6)){
    PFHd_logic = 4.23*Math.pow(10,-6);
    PFHd_output = PFHr - PFHd_logic - fixvalue_in;
    if (PFHd_output >= 2.954*Math.pow(10,-5)){
      PFHd_output = 2.954*Math.pow(10,-5);
    }
    else if (PFHd_output <= 4.23*Math.pow(10,-6)){
      PFHd_output = 4.23*Math.pow(10,-6);
    }
  }
  else {
    Dlogic = PFHd0_logic - PFHd_logic;
    PFHd_output = PFHd_output + Dlogic;
    if (PFHd_output >= 2.954*Math.pow(10,-5)){
      PFHd_output = 2.954*Math.pow(10,-5);
      PFHd_logic = PFHr - PFHd_output - fixvalue_in;
    }
    else if (PFHd_output <= 4.23*Math.pow(10,-6)){
      PFHd_output = 4.23*Math.pow(10,-6);
      PFHd_logic = PFHr - PFHd_output - fixvalue_in;
    }
  }
}
else if (category.equals("1")) {
  if (PFHd_logic >= 1.52*Math.pow(10,-6)){
    PFHd_logic = 1.52*Math.pow(10,-6);
    PFHd_output = PFHr - PFHd_logic - fixvalue_in;
    if (PFHd_output >= 1.52*Math.pow(10,-6)){
      PFHd_output = 1.52*Math.pow(10,-6);
    }
  }
  else if (PFHd_output <= 1.14*Math.pow(10,-6)){
    PFHd_output = 1.14*Math.pow(10,-6);
  }
}
else if (PFHd_logic <= 1.14*Math.pow(10,-6)){
  PFHd_logic = 1.14*Math.pow(10,-6);
  PFHd_output = PFHr - PFHd_logic - fixvalue_in;
  if (PFHd_output >= 1.52*Math.pow(10,-6)){
    PFHd_output = 1.52*Math.pow(10,-6);
  }
  else if (PFHd_output <= 1.14*Math.pow(10,-6)){

```

```
        PFHd_output = 1.14*Math.pow(10,-6);
    }
}
else {
    Dlogic = PFHd0_logic - PFHd_logic;
    PFHd_output = PFHd_output + Dlogic;
    if (PFHd_output >= 1.52*Math.pow(10,-6)){
        PFHd_output = 1.52*Math.pow(10,-6);
        PFHd_logic = PFHr - PFHd_output - fixvalue_in;
    }
    else if (PFHd_output <= 1.14*Math.pow(10,-6)){
        PFHd_output = 1.14*Math.pow(10,-6);
        PFHd_logic = PFHr - PFHd_output - fixvalue_in;
    }
}
}
else if (category.equals("2")) {
    if (PFHd_logic >= 2.4744*Math.pow(10,-5)){
        PFHd_logic = 2.4744*Math.pow(10,-5);
        PFHd_output = PFHr - PFHd_logic - fixvalue_in;
        if (PFHd_output >= 2.4744*Math.pow(10,-5)){
            PFHd_output = 2.4744*Math.pow(10,-5);
        }
        else if (PFHd_output <= 5.28*Math.pow(10,-7)){
            PFHd_output = 5.28*Math.pow(10,-7);
        }
    }
    else if (PFHd_logic <= 5.28*Math.pow(10,-7)){
        PFHd_logic = 5.28*Math.pow(10,-7);
        PFHd_output = PFHr - PFHd_logic - fixvalue_in;
        if (PFHd_output >= 2.4744*Math.pow(10,-5)){
            PFHd_output = 2.4744*Math.pow(10,-5);
        }
        else if (PFHd_output <= 5.28*Math.pow(10,-7)){
            PFHd_output = 5.28*Math.pow(10,-7);
        }
    }
}
else {
    Dlogic = PFHd0_logic - PFHd_logic;
    PFHd_output = PFHd_output + Dlogic;
    if (PFHd_output >= 2.4744*Math.pow(10,-5)){
        PFHd_output = 2.4744*Math.pow(10,-5);
        PFHd_logic = PFHr - PFHd_output - fixvalue_in;
    }
    else if (PFHd_output <= 5.28*Math.pow(10,-7)){
        PFHd_output = 5.28*Math.pow(10,-7);
        PFHd_logic = PFHr - PFHd_output - fixvalue_in;
    }
}
}
else if (category.equals("3")) {
    if (PFHd_logic >= 1.2398*Math.pow(10,-5)){
        PFHd_logic = 1.2398*Math.pow(10,-5);
    }
}
```

```

    PFHd_output = PFHr - PFHd_logic - fixvalue_in;
    if (PFHd_output >= 1.2398*Math.pow(10,-5)){
        PFHd_output = 1.2398*Math.pow(10,-5);
    }
    else if (PFHd_output <= 1.01*Math.pow(10,-7)){
        PFHd_output = 1.01*Math.pow(10,-7);
    }
}
else if (PFHd_logic <= 1.01*Math.pow(10,-7)){
    PFHd_logic = 1.01*Math.pow(10,-7);
    PFHd_output = PFHr - PFHd_logic - fixvalue_in;
    if (PFHd_output >= 1.2398*Math.pow(10,-5)){
        PFHd_output = 1.2398*Math.pow(10,-5);
    }
    else if (PFHd_output <= 1.01*Math.pow(10,-7)){
        PFHd_output = 1.01*Math.pow(10,-7);
    }
}
else {
    Dlogic = PFHd0_logic - PFHd_logic;
    PFHd_output = PFHd_output + Dlogic;
    if (PFHd_output >= 1.2398*Math.pow(10,-5)){
        PFHd_output = 1.2398*Math.pow(10,-5);
        PFHd_logic = PFHr - PFHd_output - fixvalue_in;
    }
    else if (PFHd_output <= 1.01*Math.pow(10,-7)){
        PFHd_output = 1.01*Math.pow(10,-7);
        PFHd_logic = PFHr - PFHd_output - fixvalue_in;
    }
}
}
else if (category.equals("4")) {
    if (PFHd_logic >= 4.6*Math.pow(10,-8)){
        PFHd_logic = 4.6*Math.pow(10,-8);
        PFHd_output = PFHr - PFHd_logic - fixvalue_in;
        if (PFHd_output >= 4.6*Math.pow(10,-8)){
            PFHd_output = 4.6*Math.pow(10,-8);
        }
    }
    else if (PFHd_output <= 2.47*Math.pow(10,-8)){
        PFHd_output = 2.47*Math.pow(10,-8);
    }
}
else if (PFHd_logic <= 2.47*Math.pow(10,-8)){
    PFHd_logic = 2.47*Math.pow(10,-8);
    PFHd_output = PFHr - PFHd_logic - fixvalue_in;
    if (PFHd_output >= 4.6*Math.pow(10,-8)){
        PFHd_output = 4.6*Math.pow(10,-8);
    }
    else if (PFHd_output <= 2.47*Math.pow(10,-8)){
        PFHd_output = 2.47*Math.pow(10,-8);
    }
}
}
else {

```

```
Dlogic = PFHd0_logic - PFHd_logic;
PFHd_output = PFHd_output + Dlogic;
if (PFHd_output >= 4.6*Math.pow(10,-8)){
    PFHd_output = 4.6*Math.pow(10,-8);
    PFHd_logic = PFHr - PFHd_output - fixvalue_in;
}
else if (PFHd_output <= 2.47*Math.pow(10,-8)){
    PFHd_output = 2.47*Math.pow(10,-8);
    PFHd_logic = PFHr - PFHd_output - fixvalue_in;
}
}
}
fix_in = PFHd_input;
fix_lo = PFHd_logic;
fix_out = PFHd_output;
}
else if (fix_input == false && fix_logic == false && fix_output ==
true) {
    if (category.equals("B")) {
        if (PFHd_logic >= 2.954*Math.pow(10,-5)){
            PFHd_logic = 2.954*Math.pow(10,-5);
            PFHd_input = PFHr - PFHd_logic - fixvalue_out;
            if (PFHd_input >= 2.954*Math.pow(10,-5)){
                PFHd_input = 2.954*Math.pow(10,-5);
            }
            else if (PFHd_input <= 4.23*Math.pow(10,-6)){
                PFHd_input = 4.23*Math.pow(10,-6);
            }
        }
        else if (PFHd_logic <= 4.23*Math.pow(10,-6)){
            PFHd_logic = 4.23*Math.pow(10,-6);
            PFHd_input = PFHr - PFHd_logic - fixvalue_out;
            if (PFHd_input >= 2.954*Math.pow(10,-5)){
                PFHd_input = 2.954*Math.pow(10,-5);
            }
            else if (PFHd_input <= 4.23*Math.pow(10,-6)){
                PFHd_input = 4.23*Math.pow(10,-6);
            }
        }
    }
    else {
        Dlogic = PFHd0_logic - PFHd_logic;
        PFHd_input = PFHd_input + Dlogic;
        if (PFHd_input >= 2.954*Math.pow(10,-5)){
            PFHd_input = 2.954*Math.pow(10,-5);
            PFHd_logic = PFHr - PFHd_input - fixvalue_out;
        }
        else if (PFHd_input <= 4.23*Math.pow(10,-6)){
            PFHd_input = 4.23*Math.pow(10,-6);
            PFHd_logic = PFHr - PFHd_input - fixvalue_out;
        }
    }
}
else if (category.equals("1")) {
```

```

if (PFHd_logic >= 1.52*Math.pow(10,-6)) {
    PFHd_logic = 1.52*Math.pow(10,-6);
    PFHd_input = PFHr - PFHd_logic - fixvalue_out;
    if (PFHd_input >= 1.52*Math.pow(10,-6)) {
        PFHd_input = 1.52*Math.pow(10,-6);
    }
    else if (PFHd_input <= 1.14*Math.pow(10,-6)) {
        PFHd_input = 1.14*Math.pow(10,-6);
    }
}
else if (PFHd_logic <= 1.14*Math.pow(10,-6)) {
    PFHd_logic = 1.14*Math.pow(10,-6);
    PFHd_input = PFHr - PFHd_logic - fixvalue_out;
    if (PFHd_input >= 1.52*Math.pow(10,-6)) {
        PFHd_input = 1.52*Math.pow(10,-6);
    }
    else if (PFHd_output <= 1.14*Math.pow(10,-6)) {
        PFHd_input = 1.14*Math.pow(10,-6);
    }
}
else {
    Dlogic = PFHd0_logic - PFHd_logic;
    PFHd_input = PFHd_input + Dlogic;
    if (PFHd_output >= 1.52*Math.pow(10,-6)) {
        PFHd_input = 1.52*Math.pow(10,-6);
        PFHd_logic = PFHr - PFHd_input - fixvalue_out;
    }
    else if (PFHd_input <= 1.14*Math.pow(10,-6)) {
        PFHd_input = 1.14*Math.pow(10,-6);
        PFHd_logic = PFHr - PFHd_input - fixvalue_out;
    }
}
}
else if (category.equals("2")) {
    if (PFHd_logic >= 2.4744*Math.pow(10,-5)) {
        PFHd_logic = 2.4744*Math.pow(10,-5);
        PFHd_input = PFHr - PFHd_logic - fixvalue_out;
        if (PFHd_input >= 2.4744*Math.pow(10,-5)) {
            PFHd_input = 2.4744*Math.pow(10,-5);
        }
        else if (PFHd_input <= 5.28*Math.pow(10,-7)) {
            PFHd_input = 5.28*Math.pow(10,-7);
        }
    }
    else if (PFHd_logic <= 5.28*Math.pow(10,-7)) {
        PFHd_logic = 5.28*Math.pow(10,-7);
        PFHd_input = PFHr - PFHd_logic - fixvalue_out;
        if (PFHd_input >= 2.4744*Math.pow(10,-5)) {
            PFHd_input = 2.4744*Math.pow(10,-5);
        }
        else if (PFHd_input <= 5.28*Math.pow(10,-7)) {
            PFHd_input = 5.28*Math.pow(10,-7);
        }
    }
}
}

```

```
}
else {
  Dlogic = PFHd0_logic - PFHd_logic;
  PFHd_input = PFHd_input + Dlogic;
  if (PFHd_input >= 2.4744*Math.pow(10,-5)){
    PFHd_input = 2.4744*Math.pow(10,-5);
    PFHd_logic = PFHr - PFHd_input - fixvalue_out;
  }
  else if (PFHd_input <= 5.28*Math.pow(10,-7)){
    PFHd_input = 5.28*Math.pow(10,-7);
    PFHd_logic = PFHr - PFHd_input - fixvalue_out;
  }
}
}
else if (category.equals("3")) {
  if (PFHd_logic >= 1.2398*Math.pow(10,-5)){
    PFHd_logic = 1.2398*Math.pow(10,-5);
    PFHd_input = PFHr - PFHd_logic - fixvalue_out;
    if (PFHd_input >= 1.2398*Math.pow(10,-5)){
      PFHd_input = 1.2398*Math.pow(10,-5);
    }
  }
  else if (PFHd_input <= 1.01*Math.pow(10,-7)){
    PFHd_input = 1.01*Math.pow(10,-7);
  }
}
else if (PFHd_logic <= 1.01*Math.pow(10,-7)){
  PFHd_logic = 1.01*Math.pow(10,-7);
  PFHd_input = PFHr - PFHd_logic - fixvalue_out;
  if (PFHd_input >= 1.2398*Math.pow(10,-5)){
    PFHd_input = 1.2398*Math.pow(10,-5);
  }
  else if (PFHd_input <= 1.01*Math.pow(10,-7)){
    PFHd_input = 1.01*Math.pow(10,-7);
  }
}
else {
  Dlogic = PFHd0_logic - PFHd_logic;
  PFHd_input = PFHd_input + Dlogic;
  if (PFHd_input >= 1.2398*Math.pow(10,-5)){
    PFHd_input = 1.2398*Math.pow(10,-5);
    PFHd_logic = PFHr - PFHd_input - fixvalue_out;
  }
  else if (PFHd_input <= 1.01*Math.pow(10,-7)){
    PFHd_input = 1.01*Math.pow(10,-7);
    PFHd_logic = PFHr - PFHd_input - fixvalue_out;
  }
}
}
else if (category.equals("4")) {
  if (PFHd_logic >= 4.6*Math.pow(10,-8)){
    PFHd_logic = 4.6*Math.pow(10,-8);
    PFHd_input = PFHr - PFHd_logic - fixvalue_out;
    if (PFHd_input >= 4.6*Math.pow(10,-8)){
```

```

        PFHd_input = 4.6*Math.pow(10,-8);
    }
    else if (PFHd_input <= 2.47*Math.pow(10,-8)){
        PFHd_input = 2.47*Math.pow(10,-8);
    }
}
else if (PFHd_logic <= 2.47*Math.pow(10,-8)){
    PFHd_logic = 2.47*Math.pow(10,-8);
    PFHd_input = PFHr - PFHd_logic - fixvalue_out;
    if (PFHd_input >= 4.6*Math.pow(10,-8)){
        PFHd_input = 4.6*Math.pow(10,-8);
    }
    else if (PFHd_input <= 2.47*Math.pow(10,-8)){
        PFHd_input = 2.47*Math.pow(10,-8);
    }
}
else {
    Dlogic = PFHd0_logic - PFHd_logic;
    PFHd_input = PFHd_input + Dlogic;
    if (PFHd_input >= 4.6*Math.pow(10,-8)){
        PFHd_input = 4.6*Math.pow(10,-8);
        PFHd_logic = PFHr - PFHd_input - fixvalue_out;
    }
    else if (PFHd_input <= 2.47*Math.pow(10,-8)){
        PFHd_input = 2.47*Math.pow(10,-8);
        PFHd_logic = PFHr - PFHd_input - fixvalue_out;
    }
}
}
fix_in = PFHd_input;
fix_lo = PFHd_logic;
fix_out = PFHd_output;
}

// Initial values.
PFHd0_input = PFHd_input;
PFHd0_logic = PFHd_logic;
PFHd0_output = PFHd_output;

// Determination of MTTFd.
if (category.equals("B")) {
    MTTFd_input = 1.14*0.0001*Math.pow(PFHd_input,-1.0002);
    MTTFd_logic = 1.14*0.0001*Math.pow(PFHd_logic,-1.0002);
    MTTFd_output = 1.14*0.0001*Math.pow(PFHd_output,-1.0002);
}
else if (category.equals("1")) {
    MTTFd_input = 1.16*0.0001*Math.pow(PFHd_input,-0.9985);
    MTTFd_logic = 1.16*0.0001*Math.pow(PFHd_logic,-0.9985);
    MTTFd_output = 1.16*0.0001*Math.pow(PFHd_output,-0.9985);
}
else if (category.equals("2")) {
    MTTFd_input = 0.0002*Math.pow(PFHd_input,-0.8975);
    MTTFd_logic = 0.0002*Math.pow(PFHd_logic,-0.8975);
}

```



```
    MTTFd_output = 0.0002*Math.pow(PFHd_output,-0.8975);
}
else if (category.equals("3")) {
    MTTFd_input = 0.001*Math.pow(PFHd_input,-0.7191);
    MTTFd_logic = 0.001*Math.pow(PFHd_logic,-0.7191);
    MTTFd_output = 0.001*Math.pow(PFHd_output,-0.7191);
}
else if (category.equals("4")) {
    MTTFd_input = 0.85*0.00002*Math.pow(PFHd_input,-0.89);
    MTTFd_logic = 0.85*0.00002*Math.pow(PFHd_logic,-0.89);
    MTTFd_output = 0.85*0.00002*Math.pow(PFHd_output,-0.89);
}

// Determination of B10d.
B10d_input = 0.1*MTTFd_input*nop_input;
B10d_logic = 0.1*MTTFd_logic*nop_logic;
B10d_output = 0.1*MTTFd_output*nop_output;

// Determination of PL_input.
if (PFHd_input > 1*Math.pow(10,-5)){
    PL_input = "a";
}
else if (PFHd_input <= 1*Math.pow(10,-5) && PFHd_input >
3*Math.pow(10,-6)){
    PL_input = "b";
}
else if (PFHd_input <= 3*Math.pow(10,-6) && PFHd_input >
1*Math.pow(10,-6)){
    PL_input = "c";
}
else if (PFHd_input <= 1*Math.pow(10,-6) && PFHd_input >
1*Math.pow(10,-7)){
    PL_input = "d";
}
else if (PFHd_input <= 1*Math.pow(10,-7)){
    PL_logic = "e";
}

// Determination of PL_logic.
if (PFHd_logic > 1*Math.pow(10,-5)){
    PL_logic = "a";
}
else if (PFHd_logic <= 1*Math.pow(10,-5) && PFHd_logic >
3*Math.pow(10,-6)){
    PL_logic = "b";
}
else if (PFHd_logic <= 3*Math.pow(10,-6) && PFHd_logic >
1*Math.pow(10,-6)){
    PL_logic = "c";
}
else if (PFHd_logic <= 1*Math.pow(10,-6) && PFHd_logic >
1*Math.pow(10,-7)){
    PL_logic = "d";
}
```

```

}
else if (PFHd_logic <= 1*Math.pow(10,-7)){
    PL_logic = "e";
}

// Determination of PL_output.
if (PFHd_output > 1*Math.pow(10,-5)){
    PL_output = "a";
}
else if (PFHd_output <= 1*Math.pow(10,-5) && PFHd_output >
3*Math.pow(10,-6)){
    PL_output = "b";
}
else if (PFHd_output <= 3*Math.pow(10,-6) && PFHd_output >
1*Math.pow(10,-6)){
    PL_output = "c";
}
else if (PFHd_output <= 1*Math.pow(10,-6) && PFHd_output >
1*Math.pow(10,-7)){
    PL_output = "d";
}
else if (PFHd_output <= 1*Math.pow(10,-7)){
    PL_output = "e";
}
}

```

B12.3. Dispositivo de salida.

```

// OUTPUT
// Determination of PFHd.
if (fix_input == false && fix_logic == false && fix_output == false) {
    if (category.equals("B")) {
        if (PFHd_output >= 2.954*Math.pow(10,-5)){
            PFHd_output = 2.954*Math.pow(10,-5);
            if (PFHd_logic <= 4.23*Math.pow(10,-6) && PFHd_input >
4.23*Math.pow(10,-6)) {
                PFHd_logic = 4.23*Math.pow(10,-6);
                PFHd_input = PFHr - PFHd_output - PFHd_logic;
                if (PFHd_input <= 4.23*Math.pow(10,-6)) {
                    PFHd_input = 4.23*Math.pow(10,-6);
                }
            } // lo min
            else if (PFHd_input <= 4.23*Math.pow(10,-6) && PFHd_logic >
4.23*Math.pow(10,-6)) {
                PFHd_input = 4.23*Math.pow(10,-6);
                PFHd_logic = PFHr - PFHd_input - PFHd_output;
                if (PFHd_logic <= 4.23*Math.pow(10,-6)) {
                    PFHd_logic = 4.23*Math.pow(10,-6);
                }
            } // in min
            else if (PFHd_logic <= 4.23*Math.pow(10,-6) && PFHd_input <=
4.23*Math.pow(10,-6)) {

```

```

        PFHd_logic = 4.23*Math.pow(10,-6);
        PFHd_input = 4.23*Math.pow(10,-6);
    } // lo min in min
} // out max
else if (PFHd_output <= 4.23*Math.pow(10,-6)) {
    PFHd_output = 4.23*Math.pow(10,-6);
    if (PFHd_logic >= 2.954*Math.pow(10,-5) && PFHd_input <
2.954*Math.pow(10,-5)){
        PFHd_logic = 2.954*Math.pow(10,-5);
        PFHd_input = PFHr - PFHd_output - PFHd_logic;
        if (PFHd_input >= 2.954*Math.pow(10,-5)) {
            PFHd_input = 2.954*Math.pow(10,-5);
        }
    } // lo max
    else if (PFHd_input >= 2.954*Math.pow(10,-5) && PFHd_logic <
2.954*Math.pow(10,-5)){
        PFHd_output = 2.954*Math.pow(10,-5);
        PFHd_logic = PFHr - PFHd_input - PFHd_output;
        if (PFHd_logic >= 2.954*Math.pow(10,-5)) {
            PFHd_logic = 2.954*Math.pow(10,-5);
        }
    } // in max
    else if (PFHd_logic >= 2.954*Math.pow(10,-5) && PFHd_input >=
2.954*Math.pow(10,-5)){
        PFHd_logic = 2.954*Math.pow(10,-5);
        PFHd_input = 2.954*Math.pow(10,-5);
    } // lo max in max
} // out min
else {
    Doutput = PFHd0_output - PFHd_output;
    PFHd_logic = PFHd_logic + Doutput/2;
    PFHd_input = PFHd_input + Doutput/2;
    if (PFHd_logic >= 2.954*Math.pow(10,-5) && PFHd_input <
2.954*Math.pow(10,-5)) {
        PFHd_logic = 2.954*Math.pow(10,-5);
        PFHd_input = PFHr - PFHd_output - PFHd_logic;
        if (PFHd_input >= 2.954*Math.pow(10,-5)) {
            PFHd_input = 2.954*Math.pow(10,-5);
            PFHd_output = PFHr - PFHd_input - PFHd_logic;
        }
    } // lo max
    else if (PFHd_input >= 2.954*Math.pow(10,-5) && PFHd_logic <
2.954*Math.pow(10,-5)) {
        PFHd_input = 2.954*Math.pow(10,-5);
        PFHd_logic = PFHr - PFHd_input - PFHd_output;
        if (PFHd_logic >= 2.954*Math.pow(10,-5)) {
            PFHd_logic = 2.954*Math.pow(10,-5);
            PFHd_output = PFHr - PFHd_input - PFHd_logic;
        }
    } // in max
    else if (PFHd_logic >= 2.954*Math.pow(10,-5) && PFHd_input >=
2.954*Math.pow(10,-5)) {
        PFHd_logic = 2.954*Math.pow(10,-5);
    }
}

```

```

    PFHd_input = 2.954*Math.pow(10,-5);
    PFHd_output = PFHr - PFHd_input - PFHd_logic;
  } // lo max in max
  else if (PFHd_logic <= 4.23*Math.pow(10,-6) && PFHd_input >
4.23*Math.pow(10,-6)) {
    PFHd_logic = 4.23*Math.pow(10,-6);
    PFHd_input = PFHr - PFHd_output - PFHd_logic;
    if (PFHd_input <= 4.23*Math.pow(10,-6)) {
      PFHd_input = 4.23*Math.pow(10,-6);
      PFHd_output = PFHr - PFHd_input - PFHd_logic;
    }
  } // lo min
  else if (PFHd_input <= 4.23*Math.pow(10,-6) && PFHd_logic >
4.23*Math.pow(10,-6)) {
    PFHd_input = 4.23*Math.pow(10,-6);
    PFHd_logic = PFHr - PFHd_input - PFHd_output;
    if (PFHd_logic <= 4.23*Math.pow(10,-6)) {
      PFHd_logic = 4.23*Math.pow(10,-6);
      PFHd_output = PFHr - PFHd_input - PFHd_logic;
    }
  } // in min
  else if (PFHd_logic <= 4.23*Math.pow(10,-6) && PFHd_input <=
4.23*Math.pow(10,-6)) {
    PFHd_logic = 4.23*Math.pow(10,-6);
    PFHd_input = 4.23*Math.pow(10,-6);
    PFHd_output = PFHr - PFHd_input - PFHd_logic;
  } // lo min in min
  } // else
} // cat B
if (category.equals("1")) {
  if (PFHd_output >= 1.52*Math.pow(10,-6)){
    PFHd_output = 1.52*Math.pow(10,-6);
    if (PFHd_logic <= 1.14*Math.pow(10,-6) && PFHd_input >
1.14*Math.pow(10,-6)) {
      PFHd_logic = 1.14*Math.pow(10,-6);
      PFHd_input = PFHr - PFHd_output - PFHd_logic;
      if (PFHd_input <= 1.14*Math.pow(10,-6)) {
        PFHd_input = 1.14*Math.pow(10,-6);
      }
    } // lo min
    else if (PFHd_input <= 1.14*Math.pow(10,-6) && PFHd_logic >
1.14*Math.pow(10,-6)) {
      PFHd_input = 1.14*Math.pow(10,-6);
      PFHd_logic = PFHr - PFHd_input - PFHd_output;
      if (PFHd_logic <= 1.14*Math.pow(10,-6)) {
        PFHd_logic = 1.14*Math.pow(10,-6);
      }
    } // in min
    else if (PFHd_logic <= 1.14*Math.pow(10,-6) && PFHd_input <=
1.14*Math.pow(10,-6)) {
      PFHd_logic = 1.14*Math.pow(10,-6);
      PFHd_input = 1.14*Math.pow(10,-6);
    } // lo min in min
  }
}

```

```

} // out max
else if (PFHd_output <= 1.14*Math.pow(10,-6)) {
    PFHd_output = 1.14*Math.pow(10,-6);
    if (PFHd_logic >= 1.52*Math.pow(10,-6) && PFHd_input <
1.52*Math.pow(10,-6)){
        PFHd_logic = 1.52*Math.pow(10,-6);
        PFHd_input = PFHr - PFHd_output - PFHd_logic;
        if (PFHd_input >= 1.52*Math.pow(10,-6)) {
            PFHd_input = 1.52*Math.pow(10,-6);
        }
    } // lo max
    else if (PFHd_input >= 1.52*Math.pow(10,-6) && PFHd_logic <
1.52*Math.pow(10,-6)){
        PFHd_input = 1.52*Math.pow(10,-6);
        PFHd_logic = PFHr - PFHd_input - PFHd_output;
        if (PFHd_logic >= 1.52*Math.pow(10,-6)) {
            PFHd_logic = 1.52*Math.pow(10,-6);
        }
    } // Out max
    else if (PFHd_logic >= 1.52*Math.pow(10,-6) && PFHd_input >=
1.52*Math.pow(10,-6)){
        PFHd_logic = 1.52*Math.pow(10,-6);
        PFHd_input = 1.52*Math.pow(10,-6);
    } // lo max in max
} // out min
else {
    Doutput = PFHd0_output - PFHd_output;
    PFHd_logic = PFHd_logic + Doutput/2;
    PFHd_input = PFHd_input + Doutput/2;
    if (PFHd_logic >= 1.52*Math.pow(10,-6) && PFHd_input <
1.52*Math.pow(10,-6)) {
        PFHd_logic = 1.52*Math.pow(10,-6);
        PFHd_input = PFHr - PFHd_output - PFHd_logic;
        if (PFHd_input >= 1.52*Math.pow(10,-6)) {
            PFHd_input = 1.52*Math.pow(10,-6);
            PFHd_output = PFHr - PFHd_input - PFHd_logic;
        }
    } // lo max
    else if (PFHd_input >= 1.52*Math.pow(10,-6) && PFHd_logic <
1.52*Math.pow(10,-6)) {
        PFHd_input = 1.52*Math.pow(10,-6);
        PFHd_logic = PFHr - PFHd_input - PFHd_output;
        if (PFHd_logic >= 1.52*Math.pow(10,-6)) {
            PFHd_logic = 1.52*Math.pow(10,-6);
            PFHd_output = PFHr - PFHd_input - PFHd_logic;
        }
    } // in max
    else if (PFHd_logic >= 1.52*Math.pow(10,-6) && PFHd_input >=
1.52*Math.pow(10,-6)) {
        PFHd_logic = 1.52*Math.pow(10,-6);
        PFHd_input = 1.52*Math.pow(10,-6);
        PFHd_output = PFHr - PFHd_input - PFHd_logic;
    } // lo max in max
}

```

```

else if (PFHd_logic <= 1.14*Math.pow(10,-6) && PFHd_input >
1.14*Math.pow(10,-6)) {
    PFHd_logic = 1.14*Math.pow(10,-6);
    PFHd_input = PFHr - PFHd_output - PFHd_logic;
    if (PFHd_input <= 1.14*Math.pow(10,-6)) {
        PFHd_input = 1.14*Math.pow(10,-6);
        PFHd_output = PFHr - PFHd_input - PFHd_logic;
    }
} // lo min
else if (PFHd_input <= 1.14*Math.pow(10,-6) && PFHd_logic >
1.14*Math.pow(10,-6)) {
    PFHd_input = 1.14*Math.pow(10,-6);
    PFHd_logic = PFHr - PFHd_input - PFHd_output;
    if (PFHd_logic <= 1.14*Math.pow(10,-6)) {
        PFHd_logic = 1.14*Math.pow(10,-6);
        PFHd_output = PFHr - PFHd_input - PFHd_logic;
    }
} // in min
else if (PFHd_logic <= 1.14*Math.pow(10,-6) && PFHd_input <=
1.14*Math.pow(10,-6)) {
    PFHd_logic = 1.14*Math.pow(10,-6);
    PFHd_input = 1.14*Math.pow(10,-6);
    PFHd_output = PFHr - PFHd_input - PFHd_logic;
} // lo min in min
} // else
} // cat 1
if (category.equals("2")) {
    if (PFHd_output >= 2.4744*Math.pow(10,-5)){
        PFHd_output = 2.4744*Math.pow(10,-5);
        if (PFHd_logic <= 5.28*Math.pow(10,-7) && PFHd_input >
5.28*Math.pow(10,-7)) {
            PFHd_logic = 5.28*Math.pow(10,-7);
            PFHd_input = PFHr - PFHd_output - PFHd_logic;
            if (PFHd_input <= 5.28*Math.pow(10,-7)) {
                PFHd_input = 5.28*Math.pow(10,-7);
            }
        } // lo min
    } else if (PFHd_input <= 5.28*Math.pow(10,-7) && PFHd_logic >
5.28*Math.pow(10,-7)) {
        PFHd_input = 5.28*Math.pow(10,-7);
        PFHd_logic = PFHr - PFHd_input - PFHd_output;
        if (PFHd_logic <= 5.28*Math.pow(10,-7)) {
            PFHd_logic = 5.28*Math.pow(10,-7);
        }
    } // in min
    else if (PFHd_logic <= 5.28*Math.pow(10,-7) && PFHd_input <=
5.28*Math.pow(10,-7)) {
        PFHd_logic = 5.28*Math.pow(10,-7);
        PFHd_input = 5.28*Math.pow(10,-7);
    } // lo min in min
} // out max
else if (PFHd_output <= 5.28*Math.pow(10,-7)) {
    PFHd_output = 5.28*Math.pow(10,-7);

```

```
    if (PFHd_logic >= 2.4744*Math.pow(10,-5) && PFHd_input <
2.4744*Math.pow(10,-5)){
        PFHd_logic = 2.4744*Math.pow(10,-5);
        PFHd_input = PFHr - PFHd_output - PFHd_logic;
        if (PFHd_input >= 2.4744*Math.pow(10,-5)) {
            PFHd_input = 2.4744*Math.pow(10,-5);
        }
    } // lo max
    else if (PFHd_input >= 2.4744*Math.pow(10,-5) && PFHd_logic <
2.4744*Math.pow(10,-5)){
        PFHd_input = 2.4744*Math.pow(10,-5);
        PFHd_logic = PFHr - PFHd_input - PFHd_output;
        if (PFHd_logic >= 2.4744*Math.pow(10,-5)) {
            PFHd_logic = 2.4744*Math.pow(10,-5);
        }
    } // int max
    else if (PFHd_logic >= 2.4744*Math.pow(10,-5) && PFHd_input >=
2.4744*Math.pow(10,-5)){
        PFHd_logic = 2.4744*Math.pow(10,-5);
        PFHd_input = 2.4744*Math.pow(10,-5);
    } // lo max in max
} // out min
else {
    Doutput = PFHd0_output - PFHd_output;
    PFHd_logic = PFHd_logic + Doutput/2;
    PFHd_input = PFHd_input + Doutput/2;
    if (PFHd_logic >= 2.4744*Math.pow(10,-5) && PFHd_input <
2.4744*Math.pow(10,-5)) {
        PFHd_logic = 2.4744*Math.pow(10,-5);
        PFHd_input = PFHr - PFHd_output - PFHd_logic;
        if (PFHd_input >= 2.4744*Math.pow(10,-5)) {
            PFHd_input = 2.4744*Math.pow(10,-5);
            PFHd_output = PFHr - PFHd_input - PFHd_logic;
        }
    } // lo max
    else if (PFHd_input >= 2.4744*Math.pow(10,-5) && PFHd_logic <
2.4744*Math.pow(10,-5)) {
        PFHd_input = 2.4744*Math.pow(10,-5);
        PFHd_logic = PFHr - PFHd_input - PFHd_output;
        if (PFHd_logic >= 2.4744*Math.pow(10,-5)) {
            PFHd_logic = 2.4744*Math.pow(10,-5);
            PFHd_output = PFHr - PFHd_input - PFHd_logic;
        }
    } // in max
    else if (PFHd_logic >= 2.4744*Math.pow(10,-5) && PFHd_input >=
2.4744*Math.pow(10,-5)) {
        PFHd_logic = 2.4744*Math.pow(10,-5);
        PFHd_input = 2.4744*Math.pow(10,-5);
        PFHd_output = PFHr - PFHd_input - PFHd_logic;
    } // lo max in max
    else if (PFHd_logic <= 5.28*Math.pow(10,-7) && PFHd_input >
5.28*Math.pow(10,-7)) {
        PFHd_logic = 5.28*Math.pow(10,-7);
    }
```

```

    PFHd_input = PFHr - PFHd_output - PFHd_logic;
    if (PFHd_input <= 5.28*Math.pow(10,-7)) {
        PFHd_input = 5.28*Math.pow(10,-7);
        PFHd_output = PFHr - PFHd_input - PFHd_logic;
    }
    // lo min
    else if (PFHd_input <= 5.28*Math.pow(10,-7) && PFHd_logic >
5.28*Math.pow(10,-7)) {
        PFHd_input = 5.28*Math.pow(10,-7);
        PFHd_logic = PFHr - PFHd_input - PFHd_output;
        if (PFHd_logic <= 5.28*Math.pow(10,-7)) {
            PFHd_logic = 5.28*Math.pow(10,-7);
            PFHd_output = PFHr - PFHd_input - PFHd_logic;
        }
    } // in min
    else if (PFHd_logic <= 5.28*Math.pow(10,-7) && PFHd_input <=
5.28*Math.pow(10,-7)) {
        PFHd_logic = 5.28*Math.pow(10,-7);
        PFHd_input = 5.28*Math.pow(10,-7);
        PFHd_output = PFHr - PFHd_input - PFHd_logic;
    } // lo min in min
    } // else
} // cat 2
if (category.equals("3")) {
    if (PFHd_output >= 1.2398*Math.pow(10,-5)){
        PFHd_output = 1.2398*Math.pow(10,-5);
        if (PFHd_logic <= 1.01*Math.pow(10,-7) && PFHd_input >
1.01*Math.pow(10,-7)) {
            PFHd_logic = 1.01*Math.pow(10,-7);
            PFHd_input = PFHr - PFHd_output - PFHd_logic;
            if (PFHd_input <= 1.01*Math.pow(10,-7)) {
                PFHd_input = 1.01*Math.pow(10,-7);
            }
        } // lo min
        else if (PFHd_input <= 1.01*Math.pow(10,-7) && PFHd_logic >
1.01*Math.pow(10,-7)) {
            PFHd_input = 1.01*Math.pow(10,-7);
            PFHd_logic = PFHr - PFHd_input - PFHd_output;
            if (PFHd_logic <= 1.01*Math.pow(10,-7)) {
                PFHd_logic = 1.01*Math.pow(10,-7);
            }
        } // in min
        else if (PFHd_logic <= 1.01*Math.pow(10,-7) && PFHd_input <=
1.01*Math.pow(10,-7)) {
            PFHd_logic = 1.01*Math.pow(10,-7);
            PFHd_input = 1.01*Math.pow(10,-7);
        } // lo min in min
    } // out max
    else if (PFHd_output <= 1.01*Math.pow(10,-7)) {
        PFHd_output = 1.01*Math.pow(10,-7);
        if (PFHd_logic >= 1.2398*Math.pow(10,-5) && PFHd_input <
1.2398*Math.pow(10,-5)){
            PFHd_logic = 1.2398*Math.pow(10,-5);

```



```
    PFHd_input = PFHr - PFHd_output - PFHd_logic;
    if (PFHd_input >= 1.2398*Math.pow(10,-5)) {
        PFHd_input = 1.2398*Math.pow(10,-5);
    }
    } // lo max
    else if (PFHd_input >= 1.2398*Math.pow(10,-5) && PFHd_logic <
1.2398*Math.pow(10,-5)) {
        PFHd_input = 1.2398*Math.pow(10,-5);
        PFHd_logic = PFHr - PFHd_input - PFHd_output;
        if (PFHd_logic >= 1.2398*Math.pow(10,-5)) {
            PFHd_logic = 1.2398*Math.pow(10,-5);
        }
    } // in max
    else if (PFHd_logic >= 1.2398*Math.pow(10,-5) && PFHd_input >=
1.2398*Math.pow(10,-5)) {
        PFHd_logic = 1.2398*Math.pow(10,-5);
        PFHd_input = 1.2398*Math.pow(10,-5);
    } // lo max in max
    } // out min
    else {
        Doutput = PFHd0_output - PFHd_output;
        PFHd_logic = PFHd_logic + Doutput/2;
        PFHd_input = PFHd_input + Doutput/2;
        if (PFHd_logic >= 1.2398*Math.pow(10,-5) && PFHd_input <
1.2398*Math.pow(10,-5)) {
            PFHd_logic = 1.2398*Math.pow(10,-5);
            PFHd_input = PFHr - PFHd_output - PFHd_logic;
            if (PFHd_input >= 1.2398*Math.pow(10,-5)) {
                PFHd_input = 1.2398*Math.pow(10,-5);
                PFHd_output = PFHr - PFHd_input - PFHd_logic;
            }
        } // lo max
        else if (PFHd_input >= 1.2398*Math.pow(10,-5) && PFHd_logic <
1.2398*Math.pow(10,-5)) {
            PFHd_input = 1.2398*Math.pow(10,-5);
            PFHd_logic = PFHr - PFHd_input - PFHd_output;
            if (PFHd_logic >= 1.2398*Math.pow(10,-5)) {
                PFHd_logic = 1.2398*Math.pow(10,-5);
                PFHd_output = PFHr - PFHd_input - PFHd_logic;
            }
        } // in max
        else if (PFHd_logic >= 1.2398*Math.pow(10,-5) && PFHd_input >=
1.2398*Math.pow(10,-5)) {
            PFHd_logic = 1.2398*Math.pow(10,-5);
            PFHd_input = 1.2398*Math.pow(10,-5);
            PFHd_output = PFHr - PFHd_input - PFHd_logic;
        } // lo max in max
        else if (PFHd_logic <= 1.01*Math.pow(10,-7) && PFHd_input >
1.01*Math.pow(10,-7)) {
            PFHd_logic = 1.01*Math.pow(10,-7);
            PFHd_input = PFHr - PFHd_output - PFHd_logic;
            if (PFHd_input <= 1.01*Math.pow(10,-7)) {
                PFHd_input = 1.01*Math.pow(10,-7);
            }
        }
    }
}
```

```

        PFHd_output = PFHr - PFHd_input - PFHd_logic;
    }
    } // lo min
    else if (PFHd_input <= 1.01*Math.pow(10,-7) && PFHd_logic >
1.01*Math.pow(10,-7)) {
        PFHd_input = 1.01*Math.pow(10,-7);
        PFHd_logic = PFHr - PFHd_input - PFHd_output;
        if (PFHd_logic <= 1.01*Math.pow(10,-7)) {
            PFHd_logic = 1.01*Math.pow(10,-7);
            PFHd_output = PFHr - PFHd_input - PFHd_logic;
        }
    } // in min
    else if (PFHd_logic <= 1.01*Math.pow(10,-7) && PFHd_input <=
1.01*Math.pow(10,-7)) {
        PFHd_logic = 1.01*Math.pow(10,-7);
        PFHd_input = 1.01*Math.pow(10,-7);
        PFHd_output = PFHr - PFHd_input - PFHd_logic;
    } // lo min in min
    } // else
    } // cat 3
    if (category.equals("4")) {
        if (PFHd_output >= 4.6*Math.pow(10,-8)) {
            PFHd_output = 4.6*Math.pow(10,-8);
            if (PFHd_logic <= 2.47*Math.pow(10,-8) && PFHd_input >
2.47*Math.pow(10,-8)) {
                PFHd_logic = 2.47*Math.pow(10,-8);
                PFHd_input = PFHr - PFHd_input - PFHd_logic;
                if (PFHd_input <= 2.47*Math.pow(10,-8)) {
                    PFHd_input = 2.47*Math.pow(10,-8);
                }
            } // lo min
            else if (PFHd_input <= 2.47*Math.pow(10,-8) && PFHd_logic >
2.47*Math.pow(10,-8)) {
                PFHd_input = 2.47*Math.pow(10,-8);
                PFHd_logic = PFHr - PFHd_input - PFHd_output;
                if (PFHd_logic <= 2.47*Math.pow(10,-8)) {
                    PFHd_logic = 2.47*Math.pow(10,-8);
                }
            } // in min
            else if (PFHd_logic <= 2.47*Math.pow(10,-8) && PFHd_input <=
2.47*Math.pow(10,-8)) {
                PFHd_logic = 2.47*Math.pow(10,-8);
                PFHd_input = 2.47*Math.pow(10,-8);
            } // lo min in min
        } // out max
        else if (PFHd_output <= 2.47*Math.pow(10,-8)) {
            PFHd_output = 2.47*Math.pow(10,-8);
            if (PFHd_logic >= 4.6*Math.pow(10,-8) && PFHd_input <
4.6*Math.pow(10,-8)) {
                PFHd_logic = 4.6*Math.pow(10,-8);
                PFHd_input = PFHr - PFHd_output - PFHd_logic;
                if (PFHd_input >= 4.6*Math.pow(10,-8)) {
                    PFHd_input = 4.6*Math.pow(10,-8);
                }
            }
        }
    }
}

```

```

    }
    } // lo max
    else if (PFHd_input >= 4.6*Math.pow(10,-8) && PFHd_logic <
4.6*Math.pow(10,-8)){
        PFHd_input = 4.6*Math.pow(10,-8);
        PFHd_logic = PFHr - PFHd_input - PFHd_output;
        if (PFHd_logic >= 4.6*Math.pow(10,-8)) {
            PFHd_logic = 4.6*Math.pow(10,-8);
        }
    } // int max
    else if (PFHd_logic >= 4.6*Math.pow(10,-8) && PFHd_input >=
4.6*Math.pow(10,-8)){
        PFHd_logic = 4.6*Math.pow(10,-8);
        PFHd_input = 4.6*Math.pow(10,-8);
    } // lo max in max
} // out min
else {
    Doutput = PFHd0_output - PFHd_output;
    PFHd_logic = PFHd_logic + Doutput/2;
    PFHd_input = PFHd_input + Doutput/2;
    if (PFHd_logic >= 4.6*Math.pow(10,-8) && PFHd_input <
4.6*Math.pow(10,-8)) {
        PFHd_logic = 4.6*Math.pow(10,-8);
        PFHd_input = PFHr - PFHd_output - PFHd_logic;
        if (PFHd_input >= 4.6*Math.pow(10,-8)) {
            PFHd_input = 4.6*Math.pow(10,-8);
            PFHd_output = PFHr - PFHd_input - PFHd_logic;
        }
    } // lo max
    else if (PFHd_input >= 4.6*Math.pow(10,-8) && PFHd_logic <
4.6*Math.pow(10,-8)) {
        PFHd_input = 4.6*Math.pow(10,-8);
        PFHd_logic = PFHr - PFHd_input - PFHd_output;
        if (PFHd_logic >= 4.6*Math.pow(10,-8)) {
            PFHd_logic = 4.6*Math.pow(10,-8);
            PFHd_output = PFHr - PFHd_input - PFHd_logic;
        }
    } // in max
    else if (PFHd_logic >= 4.6*Math.pow(10,-8) && PFHd_input >=
4.6*Math.pow(10,-8)) {
        PFHd_logic = 4.6*Math.pow(10,-8);
        PFHd_input = 4.6*Math.pow(10,-8);
        PFHd_output = PFHr - PFHd_input - PFHd_logic;
    } // lo max in max
    else if (PFHd_logic <= 2.47*Math.pow(10,-8) && PFHd_input >
2.47*Math.pow(10,-8)) {
        PFHd_logic = 2.47*Math.pow(10,-8);
        PFHd_input = PFHr - PFHd_output - PFHd_logic;
        if (PFHd_input <= 2.47*Math.pow(10,-8)) {
            PFHd_input = 2.47*Math.pow(10,-8);
            PFHd_output = PFHr - PFHd_input - PFHd_logic;
        }
    } // lo min
}

```

```

else if (PFHd_input <= 2.47*Math.pow(10,-8) && PFHd_logic >
2.47*Math.pow(10,-8)) {
    PFHd_input = 2.47*Math.pow(10,-8);
    PFHd_logic = PFHr - PFHd_input - PFHd_output;
    if (PFHd_logic <= 2.47*Math.pow(10,-8)) {
        PFHd_logic = 2.47*Math.pow(10,-8);
        PFHd_output = PFHr - PFHd_input - PFHd_logic;
    }
} // out min
else if (PFHd_logic <= 2.47*Math.pow(10,-8) && PFHd_input <=
2.47*Math.pow(10,-8)) {
    PFHd_logic = 2.47*Math.pow(10,-8);
    PFHd_input = 2.47*Math.pow(10,-8);
    PFHd_output = PFHr - PFHd_input - PFHd_logic;
} // lo min in min
} // else
} // cat 4
fix_in = PFHd_input;
fix_lo = PFHd_logic;
fix_out = PFHd_output;
} // in = F, lo = F, out = F
else if (fix_output == true){
    PFHd_input = PFHd_input;
    PFHd_logic = PFHd_logic;
    PFHd_output = fixvalue_out;
} // out = F
else if (fix_input == true && fix_logic == true && fix_output ==
false){
    PFHd_input = fixvalue_in;
    PFHd_logic = fixvalue_lo;
    PFHd_output = fix_out;
} // in = T, L = T, out = F
else if (fix_input == true && fix_logic == false && fix_output ==
false){
    if (category.equals("B")) {
        if (PFHd_output >= 2.954*Math.pow(10,-5)){
            PFHd_output = 2.954*Math.pow(10,-5);
            PFHd_logic = PFHr - PFHd_output - fixvalue_in;
            if (PFHd_logic >= 2.954*Math.pow(10,-5)){
                PFHd_logic = 2.954*Math.pow(10,-5);
            }
        }
        else if (PFHd_logic <= 4.23*Math.pow(10,-6)){
            PFHd_logic = 4.23*Math.pow(10,-6);
        }
    }
}
else if (PFHd_output <= 4.23*Math.pow(10,-6)){
    PFHd_output = 4.23*Math.pow(10,-6);
    PFHd_logic = PFHr - PFHd_output - fixvalue_in;
    if (PFHd_logic >= 2.954*Math.pow(10,-5)){
        PFHd_logic = 2.954*Math.pow(10,-5);
    }
}
else if (PFHd_logic <= 4.23*Math.pow(10,-6)){
    PFHd_logic = 4.23*Math.pow(10,-6);
}

```

```
    }
  }
  else {
    Doutput = PFHd0_output - PFHd_output;
    PFHd_logic = PFHd_logic + Doutput;
    if (PFHd_logic >= 2.954*Math.pow(10,-5)){
      PFHd_logic = 2.954*Math.pow(10,-5);
      PFHd_output = PFHr - PFHd_logic - fixvalue_in;
    }
    else if (PFHd_logic <= 4.23*Math.pow(10,-6)){
      PFHd_logic = 4.23*Math.pow(10,-6);
      PFHd_output = PFHr - PFHd_logic - fixvalue_in;
    }
  }
}
else if (category.equals("1")) {
  if (PFHd_output >= 1.52*Math.pow(10,-6)){
    PFHd_output = 1.52*Math.pow(10,-6);
    PFHd_logic = PFHr - PFHd_output - fixvalue_in;
    if (PFHd_logic >= 1.52*Math.pow(10,-6)){
      PFHd_logic = 1.52*Math.pow(10,-6);
    }
    else if (PFHd_logic <= 1.14*Math.pow(10,-6)){
      PFHd_logic = 1.14*Math.pow(10,-6);
    }
  }
  else if (PFHd_output <= 1.14*Math.pow(10,-6)){
    PFHd_output = 1.14*Math.pow(10,-6);
    PFHd_logic = PFHr - PFHd_output - fixvalue_in;
    if (PFHd_logic >= 1.52*Math.pow(10,-6)){
      PFHd_logic = 1.52*Math.pow(10,-6);
    }
    else if (PFHd_logic <= 1.14*Math.pow(10,-6)){
      PFHd_logic = 1.14*Math.pow(10,-6);
    }
  }
}
else {
  Doutput = PFHd0_output - PFHd_output;
  PFHd_logic = PFHd_logic + Doutput;
  if (PFHd_logic >= 1.52*Math.pow(10,-6)){
    PFHd_logic = 1.52*Math.pow(10,-6);
    PFHd_output = PFHr - PFHd_logic - fixvalue_in;
  }
  else if (PFHd_logic <= 1.14*Math.pow(10,-6)){
    PFHd_logic = 1.14*Math.pow(10,-6);
    PFHd_output = PFHr - PFHd_logic - fixvalue_in;
  }
}
}
else if (category.equals("2")) {
  if (PFHd_output >= 2.4744*Math.pow(10,-5)){
    PFHd_output = 2.4744*Math.pow(10,-5);
    PFHd_logic = PFHr - PFHd_output - fixvalue_in;
```

```

    if (PFHd_logic >= 2.4744*Math.pow(10,-5)){
        PFHd_logic = 2.4744*Math.pow(10,-5);
    }
    else if (PFHd_logic <= 5.28*Math.pow(10,-7)){
        PFHd_logic = 5.28*Math.pow(10,-7);
    }
}
else if (PFHd_output <= 5.28*Math.pow(10,-7)){
    PFHd_output = 5.28*Math.pow(10,-7);
    PFHd_logic = PFHr - PFHd_output - fixvalue_in;
    if (PFHd_logic >= 2.4744*Math.pow(10,-5)){
        PFHd_logic = 2.4744*Math.pow(10,-5);
    }
    else if (PFHd_logic <= 5.28*Math.pow(10,-7)){
        PFHd_logic = 5.28*Math.pow(10,-7);
    }
}
else {
    Doutput = PFHd0_output - PFHd_output;
    PFHd_logic = PFHd_logic + Doutput;
    if (PFHd_logic >= 2.4744*Math.pow(10,-5)){
        PFHd_logic = 2.4744*Math.pow(10,-5);
        PFHd_output = PFHr - PFHd_logic - fixvalue_in;
    }
    else if (PFHd_logic <= 5.28*Math.pow(10,-7)){
        PFHd_logic = 5.28*Math.pow(10,-7);
        PFHd_output = PFHr - PFHd_logic - fixvalue_in;
    }
}
}
else if (category.equals("3")) {
    if (PFHd_output >= 1.2398*Math.pow(10,-5)){
        PFHd_output = 1.2398*Math.pow(10,-5);
        PFHd_logic = PFHr - PFHd_output - fixvalue_in;
        if (PFHd_logic >= 1.2398*Math.pow(10,-5)){
            PFHd_logic = 1.2398*Math.pow(10,-5);
        }
    }
    else if (PFHd_logic <= 1.01*Math.pow(10,-7)){
        PFHd_logic = 1.01*Math.pow(10,-7);
    }
}
else if (PFHd_output <= 1.01*Math.pow(10,-7)){
    PFHd_output = 1.01*Math.pow(10,-7);
    PFHd_logic = PFHr - PFHd_output - fixvalue_in;
    if (PFHd_logic >= 1.2398*Math.pow(10,-5)){
        PFHd_logic = 1.2398*Math.pow(10,-5);
    }
    else if (PFHd_logic <= 1.01*Math.pow(10,-7)){
        PFHd_logic = 1.01*Math.pow(10,-7);
    }
}
else {
    Doutput = PFHd0_output - PFHd_output;

```

```
PFHd_logic = PFHd_logic + Doutput;
if (PFHd_logic >= 1.2398*Math.pow(10,-5)){
    PFHd_logic = 1.2398*Math.pow(10,-5);
    PFHd_output = PFHr - PFHd_logic - fixvalue_in;
}
else if (PFHd_logic <= 1.01*Math.pow(10,-7)){
    PFHd_logic = 1.01*Math.pow(10,-7);
    PFHd_output = PFHr - PFHd_logic - fixvalue_in;
}
}
}
else if (category.equals("4")) {
    if (PFHd_output >= 4.6*Math.pow(10,-8)){
        PFHd_output = 4.6*Math.pow(10,-8);
        PFHd_logic = PFHr - PFHd_output - fixvalue_in;
        if (PFHd_logic >= 4.6*Math.pow(10,-8)){
            PFHd_logic = 4.6*Math.pow(10,-8);
        }
        else if (PFHd_logic <= 2.47*Math.pow(10,-8)){
            PFHd_logic = 2.47*Math.pow(10,-8);
        }
    }
    else if (PFHd_output <= 2.47*Math.pow(10,-8)){
        PFHd_output = 2.47*Math.pow(10,-8);
        PFHd_logic = PFHr - PFHd_output - fixvalue_in;
        if (PFHd_logic >= 4.6*Math.pow(10,-8)){
            PFHd_logic = 4.6*Math.pow(10,-8);
        }
        else if (PFHd_logic <= 2.47*Math.pow(10,-8)){
            PFHd_logic = 2.47*Math.pow(10,-8);
        }
    }
    else {
        Doutput = PFHd0_output - PFHd_output;
        PFHd_logic = PFHd_logic + Doutput;
        if (PFHd_logic >= 4.6*Math.pow(10,-8)){
            PFHd_logic = 4.6*Math.pow(10,-8);
            PFHd_output = PFHr - PFHd_logic - fixvalue_in;
        }
        else if (PFHd_logic <= 2.47*Math.pow(10,-8)){
            PFHd_logic = 2.47*Math.pow(10,-8);
            PFHd_output = PFHr - PFHd_logic - fixvalue_in;
        }
    }
}
fix_in = PFHd_input;
fix_lo = PFHd_logic;
fix_out = PFHd_output;
}
else if (fix_input == false && fix_logic == true && fix_output ==
false){
    if (category.equals("B")) {
        if (PFHd_output >= 2.954*Math.pow(10,-5)){
```

```

PFHd_output = 2.954*Math.pow(10,-5);
PFHd_input = PFHr - PFHd_output - fixvalue_lo;
if (PFHd_input >= 2.954*Math.pow(10,-5)){
    PFHd_input = 2.954*Math.pow(10,-5);
}
else if (PFHd_input <= 4.23*Math.pow(10,-6)){
    PFHd_input = 4.23*Math.pow(10,-6);
}
}
else if (PFHd_output <= 4.23*Math.pow(10,-6)){
    PFHd_output = 4.23*Math.pow(10,-6);
    PFHd_input = PFHr - PFHd_output - fixvalue_lo;
    if (PFHd_input >= 2.954*Math.pow(10,-5)){
        PFHd_input = 2.954*Math.pow(10,-5);
    }
    else if (PFHd_input <= 4.23*Math.pow(10,-6)){
        PFHd_input = 4.23*Math.pow(10,-6);
    }
}
}
else {
    Doutput = PFHd0_output - PFHd_output;
    PFHd_input = PFHd_input + Doutput;
    if (PFHd_input >= 2.954*Math.pow(10,-5)){
        PFHd_input = 2.954*Math.pow(10,-5);
        PFHd_output = PFHr - PFHd_input - fixvalue_lo;
    }
    else if (PFHd_input <= 4.23*Math.pow(10,-6)){
        PFHd_input = 4.23*Math.pow(10,-6);
        PFHd_output = PFHr - PFHd_input - fixvalue_lo;
    }
}
}
else if (category.equals("1")) {
    if (PFHd_output >= 1.52*Math.pow(10,-6)){
        PFHd_output = 1.52*Math.pow(10,-6);
        PFHd_input = PFHr - PFHd_output - fixvalue_lo;
        if (PFHd_input >= 1.52*Math.pow(10,-6)){
            PFHd_input = 1.52*Math.pow(10,-6);
        }
    }
    else if (PFHd_input <= 1.14*Math.pow(10,-6)){
        PFHd_input = 1.14*Math.pow(10,-6);
    }
}
}
else if (PFHd_output <= 1.14*Math.pow(10,-6)){
    PFHd_output = 1.14*Math.pow(10,-6);
    PFHd_input = PFHr - PFHd_output - fixvalue_lo;
    if (PFHd_input >= 1.52*Math.pow(10,-6)){
        PFHd_input = 1.52*Math.pow(10,-6);
    }
    else if (PFHd_input <= 1.14*Math.pow(10,-6)){
        PFHd_input = 1.14*Math.pow(10,-6);
    }
}
}
}

```



```
else {
    Doutput = PFHd0_output - PFHd_output;
    PFHd_input = PFHd_input + Doutput;
    if (PFHd_input >= 1.52*Math.pow(10,-6)){
        PFHd_input = 1.52*Math.pow(10,-6);
        PFHd_output = PFHr - PFHd_input - fixvalue_lo;
    }
    else if (PFHd_input <= 1.14*Math.pow(10,-6)){
        PFHd_input = 1.14*Math.pow(10,-6);
        PFHd_output = PFHr - PFHd_input - fixvalue_lo;
    }
}
}
else if (category.equals("2")) {
    if (PFHd_output >= 2.4744*Math.pow(10,-5)){
        PFHd_output = 2.4744*Math.pow(10,-5);
        PFHd_input = PFHr - PFHd_output - fixvalue_lo;
        if (PFHd_input >= 2.4744*Math.pow(10,-5)){
            PFHd_input = 2.4744*Math.pow(10,-5);
        }
        else if (PFHd_input <= 5.28*Math.pow(10,-7)){
            PFHd_input = 5.28*Math.pow(10,-7);
        }
    }
    else if (PFHd_output <= 5.28*Math.pow(10,-7)){
        PFHd_output = 5.28*Math.pow(10,-7);
        PFHd_input = PFHr - PFHd_output - fixvalue_lo;
        if (PFHd_input >= 2.4744*Math.pow(10,-5)){
            PFHd_input = 2.4744*Math.pow(10,-5);
        }
        else if (PFHd_input <= 5.28*Math.pow(10,-7)){
            PFHd_input = 5.28*Math.pow(10,-7);
        }
    }
}
else {
    Doutput = PFHd0_output - PFHd_output;
    PFHd_input = PFHd_input + Doutput;
    if (PFHd_input >= 2.4744*Math.pow(10,-5)){
        PFHd_input = 2.4744*Math.pow(10,-5);
        PFHd_output = PFHr - PFHd_input - fixvalue_lo;
    }
    else if (PFHd_input <= 5.28*Math.pow(10,-7)){
        PFHd_input = 5.28*Math.pow(10,-7);
        PFHd_output = PFHr - PFHd_input - fixvalue_lo;
    }
}
}
else if (category.equals("3")) {
    if (PFHd_output >= 1.2398*Math.pow(10,-5)){
        PFHd_output = 1.2398*Math.pow(10,-5);
        PFHd_input = PFHr - PFHd_output - fixvalue_lo;
        if (PFHd_input >= 1.2398*Math.pow(10,-5)){
            PFHd_input = 1.2398*Math.pow(10,-5);
        }
    }
}
```

```

    }
    else if (PFHd_input <= 1.01*Math.pow(10,-7)){
        PFHd_input = 1.01*Math.pow(10,-7);
    }
}
else if (PFHd_output <= 1.01*Math.pow(10,-7)){
    PFHd_output = 1.01*Math.pow(10,-7);
    PFHd_input = PFHr - PFHd_output - fixvalue_lo;
    if (PFHd_input >= 1.2398*Math.pow(10,-5)){
        PFHd_input = 1.2398*Math.pow(10,-5);
    }
    else if (PFHd_input <= 1.01*Math.pow(10,-7)){
        PFHd_input = 1.01*Math.pow(10,-7);
    }
}
else {
    Doutput = PFHd0_output - PFHd_output;
    PFHd_input = PFHd_input + Doutput;
    if (PFHd_input >= 1.2398*Math.pow(10,-5)){
        PFHd_input = 1.2398*Math.pow(10,-5);
        PFHd_output = PFHr - PFHd_input - fixvalue_lo;
    }
    else if (PFHd_input <= 1.01*Math.pow(10,-7)){
        PFHd_input = 1.01*Math.pow(10,-7);
        PFHd_output = PFHr - PFHd_input - fixvalue_lo;
    }
}
}
else if (category.equals("4")) {
    if (PFHd_output >= 4.6*Math.pow(10,-8)){
        PFHd_output = 4.6*Math.pow(10,-8);
        PFHd_input = PFHr - PFHd_output - fixvalue_lo;
        if (PFHd_input >= 4.6*Math.pow(10,-8)){
            PFHd_input = 4.6*Math.pow(10,-8);
        }
        else if (PFHd_input <= 2.47*Math.pow(10,-8)){
            PFHd_input = 2.47*Math.pow(10,-8);
        }
    }
    else if (PFHd_output <= 2.47*Math.pow(10,-8)){
        PFHd_output = 2.47*Math.pow(10,-8);
        PFHd_input = PFHr - PFHd_output - fixvalue_lo;
        if (PFHd_input >= 4.6*Math.pow(10,-8)){
            PFHd_input = 4.6*Math.pow(10,-8);
        }
        else if (PFHd_input <= 2.47*Math.pow(10,-8)){
            PFHd_input = 2.47*Math.pow(10,-8);
        }
    }
}
else {
    Doutput = PFHd0_output - PFHd_output;
    PFHd_input = PFHd_input + Doutput;
    if (PFHd_input >= 4.6*Math.pow(10,-8)){

```

```
        PFHd_input = 4.6*Math.pow(10,-8);
        PFHd_output = PFHr - PFHd_input - fixvalue_lo;
    }
    else if (PFHd_input <= 2.47*Math.pow(10,-8)){
        PFHd_input = 2.47*Math.pow(10,-8);
        PFHd_output = PFHr - PFHd_input - fixvalue_lo;
    }
}
}
fix_in = PFHd_input;
fix_lo = PFHd_logic;
fix_out = PFHd_output;
}

// Initial values.
PFHd0_input = PFHd_input;
PFHd0_logic = PFHd_logic;
PFHd0_output = PFHd_output;

// Determination of MTTFd.
if (category.equals("B")) {
    MTTFd_input = 1.14*0.0001*Math.pow(PFHd_input,-1.0002);
    MTTFd_logic = 1.14*0.0001*Math.pow(PFHd_logic,-1.0002);
    MTTFd_output = 1.14*0.0001*Math.pow(PFHd_output,-1.0002);
}
else if (category.equals("1")) {
    MTTFd_input = 1.16*0.0001*Math.pow(PFHd_input,-0.9985);
    MTTFd_logic = 1.16*0.0001*Math.pow(PFHd_logic,-0.9985);
    MTTFd_output = 1.16*0.0001*Math.pow(PFHd_output,-0.9985);
}
else if (category.equals("2")) {
    MTTFd_input = 0.0002*Math.pow(PFHd_input,-0.8975);
    MTTFd_logic = 0.0002*Math.pow(PFHd_logic,-0.8975);
    MTTFd_output = 0.0002*Math.pow(PFHd_output,-0.8975);
}
else if (category.equals("3")) {
    MTTFd_input = 0.001*Math.pow(PFHd_input,-0.7191);
    MTTFd_logic = 0.001*Math.pow(PFHd_logic,-0.7191);
    MTTFd_output = 0.001*Math.pow(PFHd_output,-0.7191);
}
else if (category.equals("4")) {
    MTTFd_input = 0.85*0.00002*Math.pow(PFHd_input,-0.89);
    MTTFd_logic = 0.85*0.00002*Math.pow(PFHd_logic,-0.89);
    MTTFd_output = 0.85*0.00002*Math.pow(PFHd_output,-0.89);
}

// Determination of B10d.
B10d_input = 0.1*MTTFd_input*nop_input;
B10d_logic = 0.1*MTTFd_logic*nop_logic;
B10d_output = 0.1*MTTFd_output*nop_output;

// Determination of PL_input.
if (PFHd_input > 1*Math.pow(10,-5)){
```

```
    PL_input = "a";
}
else if (PFHd_input <= 1*Math.pow(10,-5) && PFHd_input >
3*Math.pow(10,-6)){
    PL_input = "b";
}
else if (PFHd_input <= 3*Math.pow(10,-6) && PFHd_input >
1*Math.pow(10,-6)){
    PL_input = "c";
}
else if (PFHd_input <= 1*Math.pow(10,-6) && PFHd_input >
1*Math.pow(10,-7)){
    PL_input = "d";
}
else if (PFHd_input <= 1*Math.pow(10,-7)){
    PL_input = "e";
}

// Determiantion of PL_logic.
if (PFHd_logic > 1*Math.pow(10,-5)){
    PL_logic = "a";
}
else if (PFHd_logic <= 1*Math.pow(10,-5) && PFHd_logic >
3*Math.pow(10,-6)){
    PL_logic = "b";
}
else if (PFHd_logic <= 3*Math.pow(10,-6) && PFHd_logic >
1*Math.pow(10,-6)){
    PL_logic = "c";
}
else if (PFHd_logic <= 1*Math.pow(10,-6) && PFHd_logic >
1*Math.pow(10,-7)){
    PL_logic = "d";
}
else if (PFHd_logic <= 1*Math.pow(10,-7)){
    PL_logic = "e";
}

// Determination of PL_output.
if (PFHd_output > 1*Math.pow(10,-5)){
    PL_output = "a";
}
else if (PFHd_output <= 1*Math.pow(10,-5) && PFHd_output >
3*Math.pow(10,-6)){
    PL_output = "b";
}
else if (PFHd_output <= 3*Math.pow(10,-6) && PFHd_output >
1*Math.pow(10,-6)){
    PL_output = "c";
}
else if (PFHd_output <= 1*Math.pow(10,-6) && PFHd_output >
1*Math.pow(10,-7)){
    PL_output = "d";
}
```

```

}
else if (PFHd_output <= 1*Math.pow(10,-7)){
    PL_output = "e";
}

```

B13. Ventana “Dispositivos de seguridad”.

B13.1. Dispositivo de entrada.

Nombre	Valor inicial	Tipo	Dimensión
Deflist_in	true	boolean	-
PLlist_in	false	boolean	-
PFHdlist_in	false	boolean	-
MTTFdlist_in	false	boolean	-
Techlist_in	false	boolean	-
Pricelist_in	false	boolean	-
deflist_in	0	int	-
defcount_in	0	int	-
pplist_in	0	int	-
plcount_in	0	int	-
pfhdlist_in	0	int	-
pfhdlist2_in	0	int	-
pfhdlist3_in	0	int	-
pfhdlist4_in	0	int	-
pfhdcount_in	0	int	-
pfhdcount2_in	0	int	-
pfhdref_in	0	double	-
matrixn_in	“o”	String	[100] [4]
matrixv_in	0	double	[100] [6]
mttfddlist_in	0	int	-
mttfddlist2_in	0	int	-
mttfddlist3_in	0	int	-
mttfddlist4_in	0	int	-
mttfddcount_in	0	int	-
mttfddcount2_in	0	int	-
mttfddref_in	0	double	-
techlist_in	0	int	-
techcount_in	0	int	-
pricelist_in	0	int	-
pricelist2_in	0	int	-
pricelist3_in	0	int	-
pricelist4_in	0	int	-

pricecount_in	0	int	-
pricecount2_in	0	int	-
priceref_in	0	double	-

Tabla B13.1. Parámetros relativos a los dispositivos de entrada.

B13.1.1. Ordenación por defacto.

```

if (Deflist_in == true) {
    PLlist_in = false;
    PFHdlist_in = false;
    MTTFdlist_in = false;
    Techlist_in = false;
    Pricelist_in = false;
}

defcount_in = 0;
_view.Inputdev_list.setText("");
for (deflist_in = 0; deflist_in < 100; deflist_in++) {
    if (!inputprint_n[deflist_in][0].equals("o")) {
        defcount_in = defcount_in + 1;
        _view.Inputdev_list.append(String.valueOf(defcount_in) + ". " +
inputprint_n[deflist_in][0] + " | " + inputprint_n[deflist_in][1] + "
| PL " + inputprint_n[deflist_in][2] + " | PFHd " +
String.valueOf(inputprint_v[deflist_in][0]) + " | MTTFd " +
String.valueOf(inputprint_v[deflist_in][1]) + " | Technology " +
inputprint_n[deflist_in][3] + " | B10d " +
String.valueOf(inputprint_v[deflist_in][2]) + " | nop " +
String.valueOf(inputprint_v[deflist_in][3]) + " | T10d " +
String.valueOf(inputprint_v[deflist_in][4]) + " | Price " +
String.valueOf(inputprint_v[deflist_in][5]) + " €" + "\n");
        _view.Inputdev_list.append("-----
-----
-----" +
"\n");
    }
    else {
        break;
    }
}
}

```

B13.1.2. Ordenación según PL.

```

if (PLlist_in == true) {
    Deflist_in = false;
    PFHdlist_in = false;
    MTTFdlist_in = false;
    Techlist_in = false;
}

```

```

    Pricelist_in = false;
}

_view.Inputdev_list.setText("");
plcount_in = 0;

for (pplist_in = 0; pplist_in < 100; pplist_in++) {
    if (inputprint_n[pplist_in][2].equals("a")) {
        plcount_in = plcount_in + 1;
        _view.Inputdev_list.append(String.valueOf(plcount_in) + ". " +
inputprint_n[pplist_in][0] + " | " + inputprint_n[pplist_in][1] + " |
PL " + inputprint_n[pplist_in][2] + " | PFHd " +
String.valueOf(inputprint_v[pplist_in][0]) + " | MTTFd " +
String.valueOf(inputprint_v[pplist_in][1]) + " | Technology " +
inputprint_n[pplist_in][3] + " | B10d " +
String.valueOf(inputprint_v[pplist_in][2]) + " | nop " +
String.valueOf(inputprint_v[pplist_in][3]) + " | T10d " +
String.valueOf(inputprint_v[pplist_in][4]) + " | Price " +
String.valueOf(inputprint_v[pplist_in][5]) + " €" + "\n");
        _view.Inputdev_list.append("-----
-----" +
"\n");
    }
    else {
        plcount_in = plcount_in + 0;
    }
}
for (pplist_in = 0; pplist_in < 100; pplist_in++) {
    if (inputprint_n[pplist_in][2].equals("b")) {
        plcount_in = plcount_in + 1;
        _view.Inputdev_list.append(String.valueOf(plcount_in) + ". " +
inputprint_n[pplist_in][0] + " | " + inputprint_n[pplist_in][1] + " |
PL " + inputprint_n[pplist_in][2] + " | PFHd " +
String.valueOf(inputprint_v[pplist_in][0]) + " | MTTFd " +
String.valueOf(inputprint_v[pplist_in][1]) + " | Technology " +
inputprint_n[pplist_in][3] + " | B10d " +
String.valueOf(inputprint_v[pplist_in][2]) + " | nop " +
String.valueOf(inputprint_v[pplist_in][3]) + " | T10d " +
String.valueOf(inputprint_v[pplist_in][4]) + " | Price " +
String.valueOf(inputprint_v[pplist_in][5]) + " €" + "\n");
        _view.Inputdev_list.append("-----
-----" +
"\n");
    }
    else {
        plcount_in = plcount_in + 0;
    }
}
for (pplist_in = 0; pplist_in < 100; pplist_in++) {
    if (inputprint_n[pplist_in][2].equals("c")) {
        plcount_in = plcount_in + 1;

```

```

        _view.Inputdev_list.append(String.valueOf(plcount_in) + ". " +
inputprint_n[pplist_in][0] + " | " + inputprint_n[pplist_in][1] + " |
PL " + inputprint_n[pplist_in][2] + " | PFHd " +
String.valueOf(inputprint_v[pplist_in][0]) + " | MTTFd " +
String.valueOf(inputprint_v[pplist_in][1]) + " | Technology " +
inputprint_n[pplist_in][3] + " | B10d " +
String.valueOf(inputprint_v[pplist_in][2]) + " | nop " +
String.valueOf(inputprint_v[pplist_in][3]) + " | T10d " +
String.valueOf(inputprint_v[pplist_in][4]) + " | Price " +
String.valueOf(inputprint_v[pplist_in][5]) + " €" + "\n");
        _view.Inputdev_list.append("-----
-----" +
"\n");
    }
    else {
        plcount_in = plcount_in + 0;
    }
}
for (pplist_in = 0; pplist_in < 100; pplist_in++) {
    if (inputprint_n[pplist_in][2].equals("d")) {
        plcount_in = plcount_in + 1;
        _view.Inputdev_list.append(String.valueOf(plcount_in) + ". " +
inputprint_n[pplist_in][0] + " | " + inputprint_n[pplist_in][1] + " |
PL " + inputprint_n[pplist_in][2] + " | PFHd " +
String.valueOf(inputprint_v[pplist_in][0]) + " | MTTFd " +
String.valueOf(inputprint_v[pplist_in][1]) + " | Technology " +
inputprint_n[pplist_in][3] + " | B10d " +
String.valueOf(inputprint_v[pplist_in][2]) + " | nop " +
String.valueOf(inputprint_v[pplist_in][3]) + " | T10d " +
String.valueOf(inputprint_v[pplist_in][4]) + " | Price " +
String.valueOf(inputprint_v[pplist_in][5]) + " €" + "\n");
        _view.Inputdev_list.append("-----
-----" +
"\n");
    }
    else {
        plcount_in = plcount_in + 0;
    }
}
for (pplist_in = 0; pplist_in < 100; pplist_in++) {
    if (inputprint_n[pplist_in][2].equals("e")) {
        plcount_in = plcount_in + 1;
        _view.Inputdev_list.append(String.valueOf(plcount_in) + ". " +
inputprint_n[pplist_in][0] + " | " + inputprint_n[pplist_in][1] + " |
PL " + inputprint_n[pplist_in][2] + " | PFHd " +
String.valueOf(inputprint_v[pplist_in][0]) + " | MTTFd " +
String.valueOf(inputprint_v[pplist_in][1]) + " | Technology " +
inputprint_n[pplist_in][3] + " | B10d " +
String.valueOf(inputprint_v[pplist_in][2]) + " | nop " +
String.valueOf(inputprint_v[pplist_in][3]) + " | T10d " +
String.valueOf(inputprint_v[pplist_in][4]) + " | Price " +

```



```
String.valueOf(inputprint_v[pclist_in][5]) + " €" + "\n");
    _view.Inputdev_list.append("-----
-----
-----" +
"\n");
}
else {
    plcount_in = plcount_in + 0;
}
}
```

B13.1.3. Ordenación según PFH_d.

```
if (PFHdlist_in == true) {
    Deflist_in = false;
    PLlist_in = false;
    MTTFdlist_in = false;
    Techlist_in = false;
    Pricelist_in = false;
}

_view.Inputdev_list.setText("");
pfhdcount_in = 0;
pfhdcount2_in = 0;

// Copy of original input device matrix.
for (pfhdlist3_in = 0; pfhdlist3_in < 100; pfhdlist3_in ++) {
    if (!inputprint_n[pfhdlist3_in][0].equals("o")) {
        matrixn_in[pfhdlist3_in][0] = inputprint_n[pfhdlist3_in][0];
        matrixn_in[pfhdlist3_in][1] = inputprint_n[pfhdlist3_in][1];
        matrixn_in[pfhdlist3_in][2] = inputprint_n[pfhdlist3_in][2];
        matrixn_in[pfhdlist3_in][3] = inputprint_n[pfhdlist3_in][3];
        matrixv_in[pfhdlist3_in][0] = inputprint_v[pfhdlist3_in][0];
        matrixv_in[pfhdlist3_in][1] = inputprint_v[pfhdlist3_in][1];
        matrixv_in[pfhdlist3_in][2] = inputprint_v[pfhdlist3_in][2];
        matrixv_in[pfhdlist3_in][3] = inputprint_v[pfhdlist3_in][3];
        matrixv_in[pfhdlist3_in][4] = inputprint_v[pfhdlist3_in][4];
        matrixv_in[pfhdlist3_in][5] = inputprint_v[pfhdlist3_in][5];
        pfhdcount2_in = pfhdcount2_in + 1;
    }
    else {
        break;
    }
}

// Sort according to PFHd.
pfhdlist_in = 0;
for (pfhdlist4_in = 0; pfhdlist4_in < pfhdcount2_in; pfhdlist4_in++){
    pfhdref_in = -1;
    for (pfhdlist2_in = 0; pfhdlist2_in < 100; pfhdlist2_in++) {
        if (pfhdref_in < matrixv_in[pfhdlist2_in][0]) {
```

```

        pfhdref_in = matrixv_in[pfhdlst2_in][0];
        pfhdlst_in = pfhdlst2_in;
    }
    else {
        pfhdref_in = pfhdref_in;
    }
}
pfhdcount_in = pfhdcount_in + 1;
_view.Inputdev_list.append(String.valueOf(pfhdcount_in) + ". " +
matrixn_in[pfhdlst_in][0] + " | " + matrixn_in[pfhdlst_in][1] + " |
PL " + matrixn_in[pfhdlst_in][2] + " | PFHd " +
String.valueOf(matrixv_in[pfhdlst_in][0]) + " | MTTFd " +
String.valueOf(matrixv_in[pfhdlst_in][1]) + " | Technology " +
matrixn_in[pfhdlst_in][3] + " | B10d " +
String.valueOf(matrixv_in[pfhdlst_in][2]) + " | nop " +
String.valueOf(matrixv_in[pfhdlst_in][3]) + " | T10d " +
String.valueOf(matrixv_in[pfhdlst_in][4]) + " | Price " +
String.valueOf(matrixv_in[pfhdlst_in][5]) + " €" + "\n");
_view.Inputdev_list.append("-----
-----" +
"\n");
matrixn_in[pfhdlst_in][0] = "o";
matrixn_in[pfhdlst_in][1] = "o";
matrixn_in[pfhdlst_in][2] = "o";
matrixn_in[pfhdlst_in][3] = "o";
matrixv_in[pfhdlst_in][0] = 0;
matrixv_in[pfhdlst_in][1] = 0;
matrixv_in[pfhdlst_in][2] = 0;
matrixv_in[pfhdlst_in][3] = 0;
matrixv_in[pfhdlst_in][4] = 0;
matrixv_in[pfhdlst_in][5] = 0;
}

```

B13.1.4. Ordenación según MTTF_d.

```

if (MTTFdlist_in == true) {
    Deflist_in = false;
    PLlist_in = false;
    PFHdlist_in = false;
    Techlist_in = false;
    Pricelist_in = false;
}

_view.Inputdev_list.setText("");
mttfdcount_in = 0;
mttfdcount2_in = 0;

// Copy of original input device matrix.
for (mttfdlist3_in = 0; mttfdlist3_in < 100; mttfdlist3_in ++) {
    if (!inputprint_n[mttfdlist3_in][0].equals("o")) {

```

```

matrixn_in[mttfdlist3_in][0] = inputprint_n[mttfdlist3_in][0];
matrixn_in[mttfdlist3_in][1] = inputprint_n[mttfdlist3_in][1];
matrixn_in[mttfdlist3_in][2] = inputprint_n[mttfdlist3_in][2];
matrixn_in[mttfdlist3_in][3] = inputprint_n[mttfdlist3_in][3];
matrixv_in[mttfdlist3_in][0] = inputprint_v[mttfdlist3_in][0];
matrixv_in[mttfdlist3_in][1] = inputprint_v[mttfdlist3_in][1];
matrixv_in[mttfdlist3_in][2] = inputprint_v[mttfdlist3_in][2];
matrixv_in[mttfdlist3_in][3] = inputprint_v[mttfdlist3_in][3];
matrixv_in[mttfdlist3_in][4] = inputprint_v[mttfdlist3_in][4];
matrixv_in[mttfdlist3_in][5] = inputprint_v[mttfdlist3_in][5];
mttfdcount2_in = mttfdcount2_in + 1;
}
else {
    break;
}
}

// Sort according to MTTFd.
mttfdlist_in = 0;
for (mttfdlist4_in = 0; mttfdlist4_in < mttfdcount2_in;
mttfdlist4_in++){
    mttfdref_in = -1;
    for (mttfdlist2_in = 0; mttfdlist2_in < 100; mttfdlist2_in++) {
        if (mttfdref_in < matrixv_in[mttfdlist2_in][1]) {
            mttfdref_in = matrixv_in[mttfdlist2_in][1];
            mttfdlist_in = mttfdlist2_in;
        }
        else {
            mttfdref_in = mttfdref_in;
        }
    }
    mttfdcount_in = mttfdcount_in + 1;
    _view.Inputdev_list.append(String.valueOf(mttfdcount_in) + ". " +
matrixn_in[mttfdlist_in][0] + " | " + matrixn_in[mttfdlist_in][1] + "
| PL " + matrixn_in[mttfdlist_in][2] + " | PFHd " +
String.valueOf(matrixv_in[mttfdlist_in][0]) + " | MTTFd " +
String.valueOf(matrixv_in[mttfdlist_in][1]) + " | Technology " +
matrixn_in[mttfdlist_in][3] + " | B10d " +
String.valueOf(matrixv_in[mttfdlist_in][2]) + " | nop " +
String.valueOf(matrixv_in[mttfdlist_in][3]) + " | T10d " +
String.valueOf(matrixv_in[mttfdlist_in][4]) + " | Price " +
String.valueOf(matrixv_in[mttfdlist_in][5]) + " €" + "\n");
    _view.Inputdev_list.append("-----
-----" +
"\n");
    matrixn_in[mttfdlist_in][0] = "o";
    matrixn_in[mttfdlist_in][1] = "o";
    matrixn_in[mttfdlist_in][2] = "o";
    matrixn_in[mttfdlist_in][3] = "o";
    matrixv_in[mttfdlist_in][0] = 0;
    matrixv_in[mttfdlist_in][1] = 0;
    matrixv_in[mttfdlist_in][2] = 0;

```

```

matrixv_in[mttfdlist_in][3] = 0;
matrixv_in[mttfdlist_in][4] = 0;
matrixv_in[mttfdlist_in][5] = 0;
}

```

B13.1.5. Ordenación según tecnología.

```

if (Techlist_in == true) {
    Deflist_in = false;
    PLlist_in = false;
    PFHdlist_in = false;
    MTTFdlist_in = false;
    Pricelist_in = false;
}

_view.Inputdev_list.setText("");
techcount_in = 0;

for (techlist_in = 0; techlist_in < 100; techlist_in++) {
    if (inputprint_n[techlist_in][3].equals("Hydraulic")) {
        techcount_in = techcount_in + 1;
        _view.Inputdev_list.append(String.valueOf(techcount_in) + ". " +
inputprint_n[techlist_in][0] + " | " + inputprint_n[techlist_in][1] +
" | PL " + inputprint_n[techlist_in][2] + " | PFHd " +
String.valueOf(inputprint_v[techlist_in][0]) + " | MTTFd " +
String.valueOf(inputprint_v[techlist_in][1]) + " | Technology " +
inputprint_n[techlist_in][3] + " | B10d " +
String.valueOf(inputprint_v[techlist_in][2]) + " | nop " +
String.valueOf(inputprint_v[techlist_in][3]) + " | T10d " +
String.valueOf(inputprint_v[techlist_in][4]) + " | Price " +
String.valueOf(inputprint_v[techlist_in][5]) + " €" + "\n");
        _view.Inputdev_list.append("-----" +
-----" +
"\n");
    }
    else {
        techcount_in = techcount_in + 0;
    }
}

for (techlist_in = 0; techlist_in < 100; techlist_in++) {
    if (inputprint_n[techlist_in][3].equals("Pneumatic")) {
        techcount_in = techcount_in + 1;
        _view.Inputdev_list.append(String.valueOf(techcount_in) + ". " +
inputprint_n[techlist_in][0] + " | " + inputprint_n[techlist_in][1] +
" | PL " + inputprint_n[techlist_in][2] + " | PFHd " +
String.valueOf(inputprint_v[techlist_in][0]) + " | MTTFd " +
String.valueOf(inputprint_v[techlist_in][1]) + " | Technology " +
inputprint_n[techlist_in][3] + " | B10d " +
String.valueOf(inputprint_v[techlist_in][2]) + " | nop " +
String.valueOf(inputprint_v[techlist_in][3]) + " | T10d " +

```

```

String.valueOf(inputprint_v[techlist_in][4]) + " | Price " +
String.valueOf(inputprint_v[techlist_in][5]) + " €" + "\n");
    _view.Inputdev_list.append("-----
-----" +
"\n");
    }
    else {
        techcount_in = techcount_in + 0;
    }
}
for (techlist_in = 0; techlist_in < 100; techlist_in++) {
    if (inputprint_n[techlist_in][3].equals("Mechanical")) {
        techcount_in = techcount_in + 1;
        _view.Inputdev_list.append(String.valueOf(techcount_in) + ". " +
inputprint_n[techlist_in][0] + " | " + inputprint_n[techlist_in][1] +
" | PL " + inputprint_n[techlist_in][2] + " | PFHd " +
String.valueOf(inputprint_v[techlist_in][0]) + " | MTTFd " +
String.valueOf(inputprint_v[techlist_in][1]) + " | Technology " +
inputprint_n[techlist_in][3] + " | B10d " +
String.valueOf(inputprint_v[techlist_in][2]) + " | nop " +
String.valueOf(inputprint_v[techlist_in][3]) + " | T10d " +
String.valueOf(inputprint_v[techlist_in][4]) + " | Price " +
String.valueOf(inputprint_v[techlist_in][5]) + " €" + "\n");
        _view.Inputdev_list.append("-----
-----" +
"\n");
    }
    else {
        techcount_in = techcount_in + 0;
    }
}
for (techlist_in = 0; techlist_in < 100; techlist_in++) {
    if (inputprint_n[techlist_in][3].equals("Electrical")) {
        techcount_in = techcount_in + 1;
        _view.Inputdev_list.append(String.valueOf(techcount_in) + ". " +
inputprint_n[techlist_in][0] + " | " + inputprint_n[techlist_in][1] +
" | PL " + inputprint_n[techlist_in][2] + " | PFHd " +
String.valueOf(inputprint_v[techlist_in][0]) + " | MTTFd " +
String.valueOf(inputprint_v[techlist_in][1]) + " | Technology " +
inputprint_n[techlist_in][3] + " | B10d " +
String.valueOf(inputprint_v[techlist_in][2]) + " | nop " +
String.valueOf(inputprint_v[techlist_in][3]) + " | T10d " +
String.valueOf(inputprint_v[techlist_in][4]) + " | Price " +
String.valueOf(inputprint_v[techlist_in][5]) + " €" + "\n");
        _view.Inputdev_list.append("-----
-----" +
"\n");
    }
    else {
        techcount_in = techcount_in + 0;
    }
}

```

```

    }
}
for (techlist_in = 0; techlist_in < 100; techlist_in++) {
    if (inputprint_n[techlist_in][3].equals("Electronics")) {
        techcount_in = techcount_in + 1;
        _view.Inputdev_list.append(String.valueOf(techcount_in) + ". " +
inputprint_n[techlist_in][0] + " | " + inputprint_n[techlist_in][1] +
" | PL " + inputprint_n[techlist_in][2] + " | PFHd " +
String.valueOf(inputprint_v[techlist_in][0]) + " | MTTFd " +
String.valueOf(inputprint_v[techlist_in][1]) + " | Technology " +
inputprint_n[techlist_in][3] + " | B10d " +
String.valueOf(inputprint_v[techlist_in][2]) + " | nop " +
String.valueOf(inputprint_v[techlist_in][3]) + " | T10d " +
String.valueOf(inputprint_v[techlist_in][4]) + " | Price " +
String.valueOf(inputprint_v[techlist_in][5]) + " €" + "\n");
        _view.Inputdev_list.append("-----
-----
-----" +
"\n");
    }
    else {
        techcount_in = techcount_in + 0;
    }
}
for (techlist_in = 0; techlist_in < 100; techlist_in++) {
    if (inputprint_n[techlist_in][3].equals("Programmable electronics"))
{
        techcount_in = techcount_in + 1;
        _view.Inputdev_list.append(String.valueOf(techcount_in) + ". " +
inputprint_n[techlist_in][0] + " | " + inputprint_n[techlist_in][1] +
" | PL " + inputprint_n[techlist_in][2] + " | PFHd " +
String.valueOf(inputprint_v[techlist_in][0]) + " | MTTFd " +
String.valueOf(inputprint_v[techlist_in][1]) + " | Technology " +
inputprint_n[techlist_in][3] + " | B10d " +
String.valueOf(inputprint_v[techlist_in][2]) + " | nop " +
String.valueOf(inputprint_v[techlist_in][3]) + " | T10d " +
String.valueOf(inputprint_v[techlist_in][4]) + " | Price " +
String.valueOf(inputprint_v[techlist_in][5]) + " €" + "\n");
        _view.Inputdev_list.append("-----
-----
-----" +
"\n");
    }
    else {
        techcount_in = techcount_in + 0;
    }
}
}

```

B13.1.6. Ordenación según precio.

```
if (Pricelist_in == true) {
    MTTFdlist_in = false;
    Deflist_in = false;
    PLlist_in = false;
    PFHdlist_in = false;
    Techlist_in = false;
}

_view.Inputdev_list.setText("");
pricecount_in = 0;
pricecount2_in = 0;

// Copy of original input device matrix.
for (pricelist3_in = 0; pricelist3_in < 100; pricelist3_in++) {
    if (!inputprint_n[pricelist3_in][0].equals("o")) {
        matrixn_in[pricelist3_in][0] = inputprint_n[pricelist3_in][0];
        matrixn_in[pricelist3_in][1] = inputprint_n[pricelist3_in][1];
        matrixn_in[pricelist3_in][2] = inputprint_n[pricelist3_in][2];
        matrixn_in[pricelist3_in][3] = inputprint_n[pricelist3_in][3];
        matrixv_in[pricelist3_in][0] = inputprint_v[pricelist3_in][0];
        matrixv_in[pricelist3_in][1] = inputprint_v[pricelist3_in][1];
        matrixv_in[pricelist3_in][2] = inputprint_v[pricelist3_in][2];
        matrixv_in[pricelist3_in][3] = inputprint_v[pricelist3_in][3];
        matrixv_in[pricelist3_in][4] = inputprint_v[pricelist3_in][4];
        matrixv_in[pricelist3_in][5] = inputprint_v[pricelist3_in][5];
        pricecount2_in = pricecount2_in + 1;
    }
    else {
        break;
    }
}

// Sort according to price.
pricelist_in = 0;
for (pricelist4_in = 0; pricelist4_in < pricecount2_in;
pricelist4_in++){
    priceref_in = -1;
    for (pricelist2_in = 0; pricelist2_in < 100; pricelist2_in++) {
        if (priceref_in < matrixv_in[pricelist2_in][5]) {
            priceref_in = matrixv_in[pricelist2_in][5];
            pricelist_in = pricelist2_in;
        }
        else {
            priceref_in = priceref_in;
        }
    }
    pricecount_in = pricecount_in + 1;
    _view.Inputdev_list.append(String.valueOf(pricecount_in) + ". " +
matrixn_in[pricelist_in][0] + " | " + matrixn_in[pricelist_in][1] + "
| PL " + matrixn_in[pricelist_in][2] + " | PFHd " +
String.valueOf(matrixv_in[pricelist_in][0]) + " | MTTFd " +
```

```
String.valueOf(matrixv_in[pricelist_in][1]) + " | Technology " +
matrixn_in[pricelist_in][3] + " | B10d " +
String.valueOf(matrixv_in[pricelist_in][2]) + " | nop " +
String.valueOf(matrixv_in[pricelist_in][3]) + " | T10d " +
String.valueOf(matrixv_in[pricelist_in][4]) + " | Price " +
String.valueOf(matrixv_in[pricelist_in][5]) + " €" + "\n");
    _view.Inputdev_list.append("-----" +
-----" +
"\n");
    matrixn_in[pricelist_in][0] = "o";
    matrixn_in[pricelist_in][1] = "o";
    matrixn_in[pricelist_in][2] = "o";
    matrixn_in[pricelist_in][3] = "o";
    matrixv_in[pricelist_in][0] = -1;
    matrixv_in[pricelist_in][1] = -1;
    matrixv_in[pricelist_in][2] = -1;
    matrixv_in[pricelist_in][3] = -1;
    matrixv_in[pricelist_in][4] = -1;
    matrixv_in[pricelist_in][5] = -1;
}
```

B13.2. Unidad lógica.

Nombre	Valor inicial	Tipo	Dimensión
Deflist_lo	true	boolean	-
PLlist_lo	false	boolean	-
PFHdlist_lo	false	boolean	-
MTTFdlist_lo	false	boolean	-
Techlist_lo	false	boolean	-
Pricelist_lo	false	boolean	-
deflist_lo	0	int	-
defcount_lo	0	int	-
pplist_lo	0	int	-
plcount_lo	0	int	-
pfhdlist_lo	0	int	-
pfhdlist2_lo	0	int	-
pfhdlist3_lo	0	int	-
pfhdlist4_lo	0	int	-
pfhdcount_lo	0	int	-
pfhdcount2_lo	0	int	-
pfhdref_lo	0	double	-
matrixn_lo	"o"	String	[100] [4]
matrixv_lo	0	double	[100] [6]
mttfldlist_lo	0	int	-

mttfdlist2_lo	0	int	-
mttfdlist3_lo	0	int	-
mttfdlist4_lo	0	int	-
mttfdcount_lo	0	int	-
mttfdcount2_lo	0	int	-
mttfdref_lo	0	double	-
techlist_lo	0	int	-
techcount_lo	0	int	-
pricelist_lo	0	int	-
pricelist2_lo	0	int	-
pricelist3_lo	0	int	-
pricelist4_lo	0	int	-
pricecount_lo	0	int	-
pricecount2_lo	0	int	-
priceref_lo	0	double	-

Tabla B13.2. Parámetros relativos a las unidades lógicas.

B13.2.1. Ordenación por defacto.

```

if (Deflist_lo == true) {
    PLlist_lo = false;
    PFHdlist_lo = false;
    MTTFdlist_lo = false;
    Techlist_lo = false;
    Pricelist_lo = false;
}

defcount_lo = 0;
_view.Logicdev_list.setText("");
for (deflist_lo = 0; deflist_lo < 100; deflist_lo++) {
    if (!logicprint_n[deflist_lo][0].equals("o")) {
        defcount_lo = defcount_lo + 1;
        _view.Logicdev_list.append(String.valueOf(defcount_lo) + ". " +
logicprint_n[deflist_lo][0] + " | " + logicprint_n[deflist_lo][1] + "
| PL " + logicprint_n[deflist_lo][2] + " | PFHd " +
String.valueOf(logicprint_v[deflist_lo][0]) + " | MTTFd " +
String.valueOf(logicprint_v[deflist_lo][1]) + " | Technology " +
logicprint_n[deflist_lo][3] + " | B10d " +
String.valueOf(logicprint_v[deflist_lo][2]) + " | nop " +
String.valueOf(logicprint_v[deflist_lo][3]) + " | T10d " +
String.valueOf(logicprint_v[deflist_lo][4]) + " | Price " +
String.valueOf(logicprint_v[deflist_lo][5]) + " €" + "\n");
        _view.Logicdev_list.append("-----
-----
-----" +
"\n");
    }
}

```

```

}
else {
    break;
}
}

```

B13.2.2. Ordenación según PL.

```

if (PLlist_lo == true) {
    Deflist_lo = false;
    PFHdlist_lo = false;
    MTTFdlist_lo = false;
    Techlist_lo = false;
    Pricelist_lo = false;
}

_view.Logicdev_list.setText("");
plcount_lo = 0;

for (pplist_lo = 0; pplist_lo < 100; pplist_lo++) {
    if (logicprint_n[pplist_lo][2].equals("a")) {
        plcount_lo = plcount_lo + 1;
        _view.Logicdev_list.append(String.valueOf(plcount_lo) + ". " +
            logicprint_n[pplist_lo][0] + " | " + logicprint_n[pplist_lo][1] + " |
            PL " + logicprint_n[pplist_lo][2] + " | PFHd " +
            String.valueOf(logicprint_v[pplist_lo][0]) + " | MTTFd " +
            String.valueOf(logicprint_v[pplist_lo][1]) + " | Technology " +
            logicprint_n[pplist_lo][3] + " | B10d " +
            String.valueOf(logicprint_v[pplist_lo][2]) + " | nop " +
            String.valueOf(logicprint_v[pplist_lo][3]) + " | T10d " +
            String.valueOf(logicprint_v[pplist_lo][4]) + " | Price " +
            String.valueOf(logicprint_v[pplist_lo][5]) + " €" + "\n");
        _view.Logicdev_list.append("-----
        -----" +
        "\n");
    }
    else {
        plcount_lo = plcount_lo + 0;
    }
}

for (pplist_lo = 0; pplist_lo < 100; pplist_lo++) {
    if (logicprint_n[pplist_lo][2].equals("b")) {
        plcount_lo = plcount_lo + 1;
        _view.Logicdev_list.append(String.valueOf(plcount_lo) + ". " +
            logicprint_n[pplist_lo][0] + " | " + logicprint_n[pplist_lo][1] + " |
            PL " + logicprint_n[pplist_lo][2] + " | PFHd " +
            String.valueOf(logicprint_v[pplist_lo][0]) + " | MTTFd " +
            String.valueOf(logicprint_v[pplist_lo][1]) + " | Technology " +
            logicprint_n[pplist_lo][3] + " | B10d " +
            String.valueOf(logicprint_v[pplist_lo][2]) + " | nop " +

```

```

String.valueOf(logicprint_v[pllist_lo][3]) + " | T10d " +
String.valueOf(logicprint_v[pllist_lo][4]) + " | Price " +
String.valueOf(logicprint_v[pllist_lo][5]) + " €" + "\n");
    _view.Logicdev_list.append("-----
-----
-----" +
"\n");
    }
    else {
        plcount_lo = plcount_lo + 0;
    }
}
for (pllism_lo = 0; pllism_lo < 100; pllism_lo++) {
    if (logicprint_n[pllism_lo][2].equals("c")) {
        plcount_lo = plcount_lo + 1;
        _view.Logicdev_list.append(String.valueOf(plcount_lo) + ". " +
logicprint_n[pllism_lo][0] + " | " + logicprint_n[pllism_lo][1] + " |
PL " + logicprint_n[pllism_lo][2] + " | PFHd " +
String.valueOf(logicprint_v[pllism_lo][0]) + " | MTTFd " +
String.valueOf(logicprint_v[pllism_lo][1]) + " | Technology " +
logicprint_n[pllism_lo][3] + " | B10d " +
String.valueOf(logicprint_v[pllism_lo][2]) + " | nop " +
String.valueOf(logicprint_v[pllism_lo][3]) + " | T10d " +
String.valueOf(logicprint_v[pllism_lo][4]) + " | Price " +
String.valueOf(logicprint_v[pllism_lo][5]) + " €" + "\n");
        _view.Logicdev_list.append("-----
-----
-----" +
"\n");
    }
    else {
        plcount_lo = plcount_lo + 0;
    }
}
for (pllism_lo = 0; pllism_lo < 100; pllism_lo++) {
    if (logicprint_n[pllism_lo][2].equals("d")) {
        plcount_lo = plcount_lo + 1;
        _view.Logicdev_list.append(String.valueOf(plcount_lo) + ". " +
logicprint_n[pllism_lo][0] + " | " + logicprint_n[pllism_lo][1] + " |
PL " + logicprint_n[pllism_lo][2] + " | PFHd " +
String.valueOf(logicprint_v[pllism_lo][0]) + " | MTTFd " +
String.valueOf(logicprint_v[pllism_lo][1]) + " | Technology " +
logicprint_n[pllism_lo][3] + " | B10d " +
String.valueOf(logicprint_v[pllism_lo][2]) + " | nop " +
String.valueOf(logicprint_v[pllism_lo][3]) + " | T10d " +
String.valueOf(logicprint_v[pllism_lo][4]) + " | Price " +
String.valueOf(logicprint_v[pllism_lo][5]) + " €" + "\n");
        _view.Logicdev_list.append("-----
-----
-----" +
"\n");
    }
    else {

```

```

    plcount_lo = plcount_lo + 0;
  }
}
for (pplist_lo = 0; pplist_lo < 100; pplist_lo++) {
  if (logicprint_n[pplist_lo][2].equals("e")) {
    plcount_lo = plcount_lo + 1;
    _view.Logicdev_list.append(String.valueOf(plcount_lo) + ". " +
logicprint_n[pplist_lo][0] + " | " + logicprint_n[pplist_lo][1] + " |
PL " + logicprint_n[pplist_lo][2] + " | PFHd " +
String.valueOf(logicprint_v[pplist_lo][0]) + " | MTTFd " +
String.valueOf(logicprint_v[pplist_lo][1]) + " | Technology " +
logicprint_n[pplist_lo][3] + " | B10d " +
String.valueOf(logicprint_v[pplist_lo][2]) + " | nop " +
String.valueOf(logicprint_v[pplist_lo][3]) + " | T10d " +
String.valueOf(logicprint_v[pplist_lo][4]) + " | Price " +
String.valueOf(logicprint_v[pplist_lo][5]) + " €" + "\n");
    _view.Logicdev_list.append("-----
-----
-----" +
"\n");
  }
  else {
    plcount_lo = plcount_lo + 0;
  }
}
}

```

B13.2.3. Ordenación según PFH_d.

```

if (PFHdlist_lo == true) {
  Deflist_lo = false;
  PLlist_lo = false;
  MTTFdlist_lo = false;
  Techlist_lo = false;
  Pricelist_lo = false;
}

_view.Logicdev_list.setText("");
pfhdcount_lo = 0;
pfhdcount2_lo = 0;

// Copy of original logic device matrix.
for (pfhdlist3_lo = 0; pfhdlist3_lo < 100; pfhdlist3_lo ++) {
  if (!logicprint_n[pfhdlist3_lo][0].equals("o")) {
    matrixn_lo[pfhdlist3_lo][0] = logicprint_n[pfhdlist3_lo][0];
    matrixn_lo[pfhdlist3_lo][1] = logicprint_n[pfhdlist3_lo][1];
    matrixn_lo[pfhdlist3_lo][2] = logicprint_n[pfhdlist3_lo][2];
    matrixn_lo[pfhdlist3_lo][3] = logicprint_n[pfhdlist3_lo][3];
    matrixv_lo[pfhdlist3_lo][0] = logicprint_v[pfhdlist3_lo][0];
    matrixv_lo[pfhdlist3_lo][1] = logicprint_v[pfhdlist3_lo][1];
    matrixv_lo[pfhdlist3_lo][2] = logicprint_v[pfhdlist3_lo][2];
    matrixv_lo[pfhdlist3_lo][3] = logicprint_v[pfhdlist3_lo][3];
  }
}

```

```

    matrixv_lo[pfhdlist3_lo][4] = logicprint_v[pfhdlist3_lo][4];
    matrixv_lo[pfhdlist3_lo][5] = logicprint_v[pfhdlist3_lo][5];
    pfhdcount2_lo = pfhdcount2_lo + 1;
}
else {
    break;
}
}

// Sort according to PFHd.
pfhdlist_lo = 0;
for (pfhdlist4_lo = 0; pfhdlist4_lo < pfhdcount2_lo; pfhdlist4_lo++){
    pfhdref_lo = -1;
    for (pfhdlist2_lo = 0; pfhdlist2_lo < 100; pfhdlist2_lo++) {
        if (pfhdref_lo < matrixv_lo[pfhdlist2_lo][0]) {
            pfhdref_lo = matrixv_lo[pfhdlist2_lo][0];
            pfhdlist_lo = pfhdlist2_lo;
        }
        else {
            pfhdref_lo = pfhdref_lo;
        }
    }
    pfhdcount_lo = pfhdcount_lo + 1;
    _view.Logicdev_list.append(String.valueOf(pfhdcount_lo) + ". " +
matrixn_lo[pfhdlist_lo][0] + " | " + matrixn_lo[pfhdlist_lo][1] + " |
PL " + matrixn_lo[pfhdlist_lo][2] + " | PFHd " +
String.valueOf(matrixv_lo[pfhdlist_lo][0]) + " | MTTFd " +
String.valueOf(matrixv_lo[pfhdlist_lo][1]) + " | Technology " +
matrixn_lo[pfhdlist_lo][3] + " | B10d " +
String.valueOf(matrixv_lo[pfhdlist_lo][2]) + " | nop " +
String.valueOf(matrixv_lo[pfhdlist_lo][3]) + " | T10d " +
String.valueOf(matrixv_lo[pfhdlist_lo][4]) + " | Price " +
String.valueOf(matrixv_lo[pfhdlist_lo][5]) + " €" + "\n");
    _view.Logicdev_list.append("-----
-----" +
"\n");
    matrixn_lo[pfhdlist_lo][0] = "o";
    matrixn_lo[pfhdlist_lo][1] = "o";
    matrixn_lo[pfhdlist_lo][2] = "o";
    matrixn_lo[pfhdlist_lo][3] = "o";
    matrixv_lo[pfhdlist_lo][0] = 0;
    matrixv_lo[pfhdlist_lo][1] = 0;
    matrixv_lo[pfhdlist_lo][2] = 0;
    matrixv_lo[pfhdlist_lo][3] = 0;
    matrixv_lo[pfhdlist_lo][4] = 0;
    matrixv_lo[pfhdlist_lo][5] = 0;
}

```

B13.2.4. Ordenación según MTTF_d.

```

if (MTTFdlist_lo == true) {
    Deflist_lo = false;
    PLlist_lo = false;
    PFHdlist_lo = false;
    Techlist_lo = false;
    Pricelist_lo = false;
}

_view.Logicdev_list.setText("");
mttfdcount_lo = 0;
mttfdcount2_lo = 0;

// Copy of original logic device matrix.
for (mttfdlist3_lo = 0; mttfdlist3_lo < 100; mttfdlist3_lo ++ ) {
    if (!logicprint_n[mttfdlist3_lo][0].equals("o")) {
        matrixn_lo[mttfdlist3_lo][0] = logicprint_n[mttfdlist3_lo][0];
        matrixn_lo[mttfdlist3_lo][1] = logicprint_n[mttfdlist3_lo][1];
        matrixn_lo[mttfdlist3_lo][2] = logicprint_n[mttfdlist3_lo][2];
        matrixn_lo[mttfdlist3_lo][3] = logicprint_n[mttfdlist3_lo][3];
        matrixv_lo[mttfdlist3_lo][0] = logicprint_v[mttfdlist3_lo][0];
        matrixv_lo[mttfdlist3_lo][1] = logicprint_v[mttfdlist3_lo][1];
        matrixv_lo[mttfdlist3_lo][2] = logicprint_v[mttfdlist3_lo][2];
        matrixv_lo[mttfdlist3_lo][3] = logicprint_v[mttfdlist3_lo][3];
        matrixv_lo[mttfdlist3_lo][4] = logicprint_v[mttfdlist3_lo][4];
        matrixv_lo[mttfdlist3_lo][5] = logicprint_v[mttfdlist3_lo][5];
        mttfdcount2_lo = mttfdcount2_lo + 1;
    }
    else {
        break;
    }
}

// Sort according to MTTFd.
mttfdlist_lo = 0;
for (mttfdlist4_lo = 0; mttfdlist4_lo < mttfdcount2_lo;
mttfdlist4_lo++){
    mttfdref_lo = -1;
    for (mttfdlist2_lo = 0; mttfdlist2_lo < 100; mttfdlist2_lo++) {
        if (mttfdref_lo < matrixv_lo[mttfdlist2_lo][1]) {
            mttfdref_lo = matrixv_lo[mttfdlist2_lo][1];
            mttfdlist_lo = mttfdlist2_lo;
        }
        else {
            mttfdref_lo = mttfdref_lo;
        }
    }
    mttfdcount_lo = mttfdcount_lo + 1;
    _view.Logicdev_list.append(String.valueOf(mttfdcount_lo) + ". " +
matrixn_lo[mttfdlist_lo][0] + " | " + matrixn_lo[mttfdlist_lo][1] + "
| PL " + matrixn_lo[mttfdlist_lo][2] + " | PFHd " +
String.valueOf(matrixv_lo[mttfdlist_lo][0]) + " | MTTFd " +

```

```
String.valueOf(matrixv_lo[mttfdlist_lo][1]) + " | Technology " +
matrixn_lo[mttfdlist_lo][3] + " | B10d " +
String.valueOf(matrixv_lo[mttfdlist_lo][2]) + " | nop " +
String.valueOf(matrixv_lo[mttfdlist_lo][3]) + " | T10d " +
String.valueOf(matrixv_lo[mttfdlist_lo][4]) + " | Price " +
String.valueOf(matrixv_lo[mttfdlist_lo][5]) + " €" + "\n");
    _view.Logicdev_list.append("-----
-----" +
"\n");
    matrixn_lo[mttfdlist_lo][0] = "o";
    matrixn_lo[mttfdlist_lo][1] = "o";
    matrixn_lo[mttfdlist_lo][2] = "o";
    matrixn_lo[mttfdlist_lo][3] = "o";
    matrixv_lo[mttfdlist_lo][0] = 0;
    matrixv_lo[mttfdlist_lo][1] = 0;
    matrixv_lo[mttfdlist_lo][2] = 0;
    matrixv_lo[mttfdlist_lo][3] = 0;
    matrixv_lo[mttfdlist_lo][4] = 0;
    matrixv_lo[mttfdlist_lo][5] = 0;
}
```

B13.2.5. Ordenación según tecnología.

```
if (Techlist_lo == true) {
    Deflist_lo = false;
    PLlist_lo = false;
    PFHdlist_lo = false;
    MTTFdlist_lo = false;
    Pricelist_lo = false;
}

_view.Logicdev_list.setText("");
techcount_lo = 0;

for (techlist_lo = 0; techlist_lo < 100; techlist_lo++) {
    if (logicprint_n[techlist_lo][3].equals("Hydraulic")) {
        techcount_lo = techcount_lo + 1;
        _view.Logicdev_list.append(String.valueOf(techcount_lo) + ". " +
logicprint_n[techlist_lo][0] + " | " + logicprint_n[techlist_lo][1] +
" | PL " + logicprint_n[techlist_lo][2] + " | PFHd " +
String.valueOf(logicprint_v[techlist_lo][0]) + " | MTTFd " +
String.valueOf(logicprint_v[techlist_lo][1]) + " | Technology " +
logicprint_n[techlist_lo][3] + " | B10d " +
String.valueOf(logicprint_v[techlist_lo][2]) + " | nop " +
String.valueOf(logicprint_v[techlist_lo][3]) + " | T10d " +
String.valueOf(logicprint_v[techlist_lo][4]) + " | Price " +
String.valueOf(logicprint_v[techlist_lo][5]) + " €" + "\n");
        _view.Logicdev_list.append("-----
-----" +
-----" +
```

```

"\n");
}
else {
    techcount_lo = techcount_lo + 0;
}
}
for (techlist_lo = 0; techlist_lo < 100; techlist_lo++) {
    if (logicprint_n[techlist_lo][3].equals("Pneumatic")) {
        techcount_lo = techcount_lo + 1;
        _view.Logicdev_list.append(String.valueOf(techcount_lo) + ". " +
logicprint_n[techlist_lo][0] + " | " + logicprint_n[techlist_lo][1] +
" | PL " + logicprint_n[techlist_lo][2] + " | PFHd " +
String.valueOf(logicprint_v[techlist_lo][0]) + " | MTTFd " +
String.valueOf(logicprint_v[techlist_lo][1]) + " | Technology " +
logicprint_n[techlist_lo][3] + " | B10d " +
String.valueOf(logicprint_v[techlist_lo][2]) + " | nop " +
String.valueOf(logicprint_v[techlist_lo][3]) + " | T10d " +
String.valueOf(logicprint_v[techlist_lo][4]) + " | Price " +
String.valueOf(logicprint_v[techlist_lo][5]) + " €" + "\n");
        _view.Logicdev_list.append("-----
-----" +
"\n");
}
else {
    techcount_lo = techcount_lo + 0;
}
}
for (techlist_lo = 0; techlist_lo < 100; techlist_lo++) {
    if (logicprint_n[techlist_lo][3].equals("Mechanical")) {
        techcount_lo = techcount_lo + 1;
        _view.Logicdev_list.append(String.valueOf(techcount_lo) + ". " +
logicprint_n[techlist_lo][0] + " | " + logicprint_n[techlist_lo][1] +
" | PL " + logicprint_n[techlist_lo][2] + " | PFHd " +
String.valueOf(logicprint_v[techlist_lo][0]) + " | MTTFd " +
String.valueOf(logicprint_v[techlist_lo][1]) + " | Technology " +
logicprint_n[techlist_lo][3] + " | B10d " +
String.valueOf(logicprint_v[techlist_lo][2]) + " | nop " +
String.valueOf(logicprint_v[techlist_lo][3]) + " | T10d " +
String.valueOf(logicprint_v[techlist_lo][4]) + " | Price " +
String.valueOf(logicprint_v[techlist_lo][5]) + " €" + "\n");
        _view.Logicdev_list.append("-----
-----" +
"\n");
}
else {
    techcount_lo = techcount_lo + 0;
}
}
for (techlist_lo = 0; techlist_lo < 100; techlist_lo++) {
    if (logicprint_n[techlist_lo][3].equals("Electrical")) {
        techcount_lo = techcount_lo + 1;

```



```

        _view.Logicdev_list.append(String.valueOf(techcount_lo) + ". " +
logicprint_n[techlist_lo][0] + " | " + logicprint_n[techlist_lo][1] +
" | PL " + logicprint_n[techlist_lo][2] + " | PFHd " +
String.valueOf(logicprint_v[techlist_lo][0]) + " | MTTFd " +
String.valueOf(logicprint_v[techlist_lo][1]) + " | Technology " +
logicprint_n[techlist_lo][3] + " | B10d " +
String.valueOf(logicprint_v[techlist_lo][2]) + " | nop " +
String.valueOf(logicprint_v[techlist_lo][3]) + " | T10d " +
String.valueOf(logicprint_v[techlist_lo][4]) + " | Price " +
String.valueOf(logicprint_v[techlist_lo][5]) + " €" + "\n");
        _view.Logicdev_list.append("-----
-----" +
"\n");
    }
    else {
        techcount_lo = techcount_lo + 0;
    }
}
for (techlist_lo = 0; techlist_lo < 100; techlist_lo++) {
    if (logicprint_n[techlist_lo][3].equals("Electronics")) {
        techcount_lo = techcount_lo + 1;
        _view.Logicdev_list.append(String.valueOf(techcount_lo) + ". " +
logicprint_n[techlist_lo][0] + " | " + logicprint_n[techlist_lo][1] +
" | PL " + logicprint_n[techlist_lo][2] + " | PFHd " +
String.valueOf(logicprint_v[techlist_lo][0]) + " | MTTFd " +
String.valueOf(logicprint_v[techlist_lo][1]) + " | Technology " +
logicprint_n[techlist_lo][3] + " | B10d " +
String.valueOf(logicprint_v[techlist_lo][2]) + " | nop " +
String.valueOf(logicprint_v[techlist_lo][3]) + " | T10d " +
String.valueOf(logicprint_v[techlist_lo][4]) + " | Price " +
String.valueOf(logicprint_v[techlist_lo][5]) + " €" + "\n");
        _view.Logicdev_list.append("-----
-----" +
"\n");
    }
    else {
        techcount_lo = techcount_lo + 0;
    }
}
for (techlist_lo = 0; techlist_lo < 100; techlist_lo++) {
    if (logicprint_n[techlist_lo][3].equals("Programmable electronics"))
    {
        techcount_lo = techcount_lo + 1;
        _view.Logicdev_list.append(String.valueOf(techcount_lo) + ". " +
logicprint_n[techlist_lo][0] + " | " + logicprint_n[techlist_lo][1] +
" | PL " + logicprint_n[techlist_lo][2] + " | PFHd " +
String.valueOf(logicprint_v[techlist_lo][0]) + " | MTTFd " +
String.valueOf(logicprint_v[techlist_lo][1]) + " | Technology " +
logicprint_n[techlist_lo][3] + " | B10d " +
String.valueOf(logicprint_v[techlist_lo][2]) + " | nop " +
String.valueOf(logicprint_v[techlist_lo][3]) + " | T10d " +

```

```

String.valueOf(logicprint_v[techlist_lo][4]) + " | Price " +
String.valueOf(logicprint_v[techlist_lo][5]) + " €" + "\n");
    _view.Logicdev_list.append("-----
-----
-----" +
"\n");
    }
    else {
        techcount_lo = techcount_lo + 0;
    }
}

```

B13.2.6. Ordenación según precio.

```

if (Pricelist_lo == true) {
    MTTFdlist_lo = false;
    Deflist_lo = false;
    PLlist_lo = false;
    PFHdlist_lo = false;
    Techlist_lo = false;
}

_view.Logicdev_list.setText("");
pricecount_lo = 0;
pricecount2_lo = 0;

// Copy of original logic device matrix.
for (pricelist3_lo = 0; pricelist3_lo < 100; pricelist3_lo++) {
    if (!logicprint_n[pricelist3_lo][0].equals("o")) {
        matrixn_lo[pricelist3_lo][0] = logicprint_n[pricelist3_lo][0];
        matrixn_lo[pricelist3_lo][1] = logicprint_n[pricelist3_lo][1];
        matrixn_lo[pricelist3_lo][2] = logicprint_n[pricelist3_lo][2];
        matrixn_lo[pricelist3_lo][3] = logicprint_n[pricelist3_lo][3];
        matrixv_lo[pricelist3_lo][0] = logicprint_v[pricelist3_lo][0];
        matrixv_lo[pricelist3_lo][1] = logicprint_v[pricelist3_lo][1];
        matrixv_lo[pricelist3_lo][2] = logicprint_v[pricelist3_lo][2];
        matrixv_lo[pricelist3_lo][3] = logicprint_v[pricelist3_lo][3];
        matrixv_lo[pricelist3_lo][4] = logicprint_v[pricelist3_lo][4];
        matrixv_lo[pricelist3_lo][5] = logicprint_v[pricelist3_lo][5];
        pricecount2_lo = pricecount2_lo + 1;
    }
    else {
        break;
    }
}

// Sort according to price.
pricelist_lo = 0;
for (pricelist4_lo = 0; pricelist4_lo < pricecount2_lo;
pricelist4_lo++){
    priceref_lo = -1;

```

```

for (pricelist2_lo = 0; pricelist2_lo < 100; pricelist2_lo++) {
    if (priceref_lo < matrixv_lo[pricelist2_lo][5]) {
        priceref_lo = matrixv_lo[pricelist2_lo][5];
        pricelist_lo = pricelist2_lo;
    }
    else {
        priceref_lo = priceref_lo;
    }
}
pricecount_lo = pricecount_lo + 1;
_view.Logicdev_list.append(String.valueOf(pricecount_lo) + ". " +
matrixn_lo[pricelist_lo][0] + " | " + matrixn_lo[pricelist_lo][1] + "
| PL " + matrixn_lo[pricelist_lo][2] + " | PFHd " +
String.valueOf(matrixv_lo[pricelist_lo][0]) + " | MTTFd " +
String.valueOf(matrixv_lo[pricelist_lo][1]) + " | Technology " +
matrixn_lo[pricelist_lo][3] + " | B10d " +
String.valueOf(matrixv_lo[pricelist_lo][2]) + " | nop " +
String.valueOf(matrixv_lo[pricelist_lo][3]) + " | T10d " +
String.valueOf(matrixv_lo[pricelist_lo][4]) + " | Price " +
String.valueOf(matrixv_lo[pricelist_lo][5]) + " €" + "\n");
_view.Logicdev_list.append("-----
-----" +
"\n");
matrixn_lo[pricelist_lo][0] = "o";
matrixn_lo[pricelist_lo][1] = "o";
matrixn_lo[pricelist_lo][2] = "o";
matrixn_lo[pricelist_lo][3] = "o";
matrixv_lo[pricelist_lo][0] = -1;
matrixv_lo[pricelist_lo][1] = -1;
matrixv_lo[pricelist_lo][2] = -1;
matrixv_lo[pricelist_lo][3] = -1;
matrixv_lo[pricelist_lo][4] = -1;
matrixv_lo[pricelist_lo][5] = -1;
}

```

B13.3. Dispositivo de salida.

Nombre	Valor inicial	Tipo	Dimensión
Deflist_out	true	boolean	-
PLlist_out	false	boolean	-
PFHdlist_out	false	boolean	-
MTTFdlist_out	false	boolean	-
Techlist_out	false	boolean	-
Pricelist_out	false	boolean	-
deflist_out	0	int	-
defcount_out	0	int	-
pplist_out	0	int	-

plcount_out	0	int	-
pfhdlist_out	0	int	-
pfhdlist2_out	0	int	-
pfhdlist3_out	0	int	-
pfhdlist4_out	0	int	-
pfhdcount_out	0	int	-
pfhdcount2_out	0	int	-
pfhdref_out	0	double	-
matrixn_out	"o"	String	[100] [4]
matrixv_out	0	double	[100] [6]
mttfdlist_out	0	int	-
mttfdlist2_out	0	int	-
mttfdlist3_out	0	int	-
mttfdlist4_out	0	int	-
mttfdcount_out	0	int	-
mttfdcount2_out	0	int	-
mttfdref_out	0	double	-
techlist_out	0	int	-
techcount_out	0	int	-
pricelist_out	0	int	-
pricelist2_out	0	int	-
pricelist3_out	0	int	-
pricelist4_out	0	int	-
pricecount_out	0	int	-
pricecount2_out	0	int	-
priceref_out	0	double	-

Tabla B13.3. Parámetros relativos a los dispositivos de salida.

B13.3.1. Ordenación por defecto.

```

if (Deflist_out == true) {
    PLlist_out = false;
    PFHdlist_out = false;
    MTTFDlist_out = false;
    Techlist_out = false;
    Pricelist_out = false;
}

defcount_out = 0;
_view.Outputdev_list.setText("");
for (deflist_out = 0; deflist_out < 100; deflist_out++) {
    if (!outputprint_n[deflist_out][0].equals("o")) {
        defcount_out = defcount_out + 1;
    }
}

```

```

        _view.Outputdev_list.append(String.valueOf(defcount_out) + ". " +
outputprint_n[deflist_out][0] + " | " + outputprint_n[deflist_out][1]
+ " | PL " + outputprint_n[deflist_out][2] + " | PFHd " +
String.valueOf(outputprint_v[deflist_out][0]) + " | MTTFd " +
String.valueOf(outputprint_v[deflist_out][1]) + " | Technology " +
outputprint_n[deflist_out][3] + " | B10d " +
String.valueOf(outputprint_v[deflist_out][2]) + " | nop " +
String.valueOf(outputprint_v[deflist_out][3]) + " | T10d " +
String.valueOf(outputprint_v[deflist_out][4]) + " | Price " +
String.valueOf(outputprint_v[deflist_out][5]) + " €" + "\n");
        _view.Outputdev_list.append("-----
-----
-----" +
"\n");
    }
    else {
        break;
    }
}

```

B13.3.2. Ordenación según PL.

```

if (PLlist_out == true) {
    Deflist_out = false;
    PFHdlist_out = false;
    MTTFdlist_out = false;
    Techlist_out = false;
    Pricelist_out = false;
}

_view.Outputdev_list.setText("");
plcount_out = 0;

for (pplist_out = 0; pplist_out < 100; pplist_out++) {
    if (outputprint_n[pplist_out][2].equals("a")) {
        plcount_out = plcount_out + 1;
        _view.Outputdev_list.append(String.valueOf(plcount_out) + ". " +
outputprint_n[pplist_out][0] + " | " + outputprint_n[pplist_out][1] +
" | PL " + outputprint_n[pplist_out][2] + " | PFHd " +
String.valueOf(outputprint_v[pplist_out][0]) + " | MTTFd " +
String.valueOf(outputprint_v[pplist_out][1]) + " | Technology " +
outputprint_n[pplist_out][3] + " | B10d " +
String.valueOf(outputprint_v[pplist_out][2]) + " | nop " +
String.valueOf(outputprint_v[pplist_out][3]) + " | T10d " +
String.valueOf(outputprint_v[pplist_out][4]) + " | Price " +
String.valueOf(outputprint_v[pplist_out][5]) + " €" + "\n");
        _view.Outputdev_list.append("-----
-----
-----" +
"\n");
    }
}

```

```

else {
    plcount_out = plcount_out + 0;
}
}
for (pplist_out = 0; pplist_out < 100; pplist_out++) {
    if (outputprint_n[pplist_out][2].equals("b")) {
        plcount_out = plcount_out + 1;
        _view.Outputdev_list.append(String.valueOf(plcount_out) + ". " +
outputprint_n[pplist_out][0] + " | " + outputprint_n[pplist_out][1] +
" | PL " + outputprint_n[pplist_out][2] + " | PFHd " +
String.valueOf(outputprint_v[pplist_out][0]) + " | MTTFd " +
String.valueOf(outputprint_v[pplist_out][1]) + " | Technology " +
outputprint_n[pplist_out][3] + " | B10d " +
String.valueOf(outputprint_v[pplist_out][2]) + " | nop " +
String.valueOf(outputprint_v[pplist_out][3]) + " | T10d " +
String.valueOf(outputprint_v[pplist_out][4]) + " | Price " +
String.valueOf(outputprint_v[pplist_out][5]) + " €" + "\n");
        _view.Outputdev_list.append("-----"
-----" +
"\n");
    }
    else {
        plcount_out = plcount_out + 0;
    }
}
}
for (pplist_out = 0; pplist_out < 100; pplist_out++) {
    if (outputprint_n[pplist_out][2].equals("c")) {
        plcount_out = plcount_out + 1;
        _view.Outputdev_list.append(String.valueOf(plcount_out) + ". " +
outputprint_n[pplist_out][0] + " | " + outputprint_n[pplist_out][1] +
" | PL " + outputprint_n[pplist_out][2] + " | PFHd " +
String.valueOf(outputprint_v[pplist_out][0]) + " | MTTFd " +
String.valueOf(outputprint_v[pplist_out][1]) + " | Technology " +
outputprint_n[pplist_out][3] + " | B10d " +
String.valueOf(outputprint_v[pplist_out][2]) + " | nop " +
String.valueOf(outputprint_v[pplist_out][3]) + " | T10d " +
String.valueOf(outputprint_v[pplist_out][4]) + " | Price " +
String.valueOf(outputprint_v[pplist_out][5]) + " €" + "\n");
        _view.Outputdev_list.append("-----"
-----" +
"\n");
    }
    else {
        plcount_out = plcount_out + 0;
    }
}
}
for (pplist_out = 0; pplist_out < 100; pplist_out++) {
    if (outputprint_n[pplist_out][2].equals("d")) {
        plcount_out = plcount_out + 1;
        _view.Outputdev_list.append(String.valueOf(plcount_out) + ". " +
outputprint_n[pplist_out][0] + " | " + outputprint_n[pplist_out][1] +

```

```

" | PL " + outputprint_n[pplist_out][2] + " | PFHd " +
String.valueOf(outputprint_v[pplist_out][0]) + " | MTTFd " +
String.valueOf(outputprint_v[pplist_out][1]) + " | Technology " +
outputprint_n[pplist_out][3] + " | B10d " +
String.valueOf(outputprint_v[pplist_out][2]) + " | nop " +
String.valueOf(outputprint_v[pplist_out][3]) + " | T10d " +
String.valueOf(outputprint_v[pplist_out][4]) + " | Price " +
String.valueOf(outputprint_v[pplist_out][5]) + " €" + "\n");
    _view.Outputdev_list.append("-----
-----" +
"\n");
    }
    else {
        plcount_out = plcount_out + 0;
    }
}
for (pplist_out = 0; pplist_out < 100; pplist_out++) {
    if (outputprint_n[pplist_out][2].equals("e")) {
        plcount_out = plcount_out + 1;
        _view.Outputdev_list.append(String.valueOf(plcount_out) + ". " +
outputprint_n[pplist_out][0] + " | " + outputprint_n[pplist_out][1] +
" | PL " + outputprint_n[pplist_out][2] + " | PFHd " +
String.valueOf(outputprint_v[pplist_out][0]) + " | MTTFd " +
String.valueOf(outputprint_v[pplist_out][1]) + " | Technology " +
outputprint_n[pplist_out][3] + " | B10d " +
String.valueOf(outputprint_v[pplist_out][2]) + " | nop " +
String.valueOf(outputprint_v[pplist_out][3]) + " | T10d " +
String.valueOf(outputprint_v[pplist_out][4]) + " | Price " +
String.valueOf(outputprint_v[pplist_out][5]) + " €" + "\n");
        _view.Outputdev_list.append("-----
-----" +
"\n");
    }
    else {
        plcount_out = plcount_out + 0;
    }
}
}

```

B13.3.3. Ordenación según PFH_d.

```

if (PFHdlist_out == true) {
    Deflist_out = false;
    PList_out = false;
    MTTFdlist_out = false;
    Techlist_out = false;
    Pricelist_out = false;
}

_view.Outputdev_list.setText("");

```

```

pfhdcount_out = 0;
pfhdcount2_out = 0;

// Copy of original output device matrix.
for (pfhdlist3_out = 0; pfhdlist3_out < 100; pfhdlist3_out ++ ) {
    if (!outputprint_n[pfhdlist3_out][0].equals("o")) {
        matrixn_out[pfhdlist3_out][0] = outputprint_n[pfhdlist3_out][0];
        matrixn_out[pfhdlist3_out][1] = outputprint_n[pfhdlist3_out][1];
        matrixn_out[pfhdlist3_out][2] = outputprint_n[pfhdlist3_out][2];
        matrixn_out[pfhdlist3_out][3] = outputprint_n[pfhdlist3_out][3];
        matrixv_out[pfhdlist3_out][0] = outputprint_v[pfhdlist3_out][0];
        matrixv_out[pfhdlist3_out][1] = outputprint_v[pfhdlist3_out][1];
        matrixv_out[pfhdlist3_out][2] = outputprint_v[pfhdlist3_out][2];
        matrixv_out[pfhdlist3_out][3] = outputprint_v[pfhdlist3_out][3];
        matrixv_out[pfhdlist3_out][4] = outputprint_v[pfhdlist3_out][4];
        matrixv_out[pfhdlist3_out][5] = outputprint_v[pfhdlist3_out][5];
        pfhdcount2_out = pfhdcount2_out + 1;
    }
    else {
        break;
    }
}

// Sort according to PFHd.
pfhdlist_out = 0;
for (pfhdlist4_out = 0; pfhdlist4_out < pfhdcount2_out;
pfhdlist4_out++){
    pfhdref_out = -1;
    for (pfhdlist2_out = 0; pfhdlist2_out < 100; pfhdlist2_out++) {
        if (pfhdref_out < matrixv_out[pfhdlist2_out][0]) {
            pfhdref_out = matrixv_out[pfhdlist2_out][0];
            pfhdlist_out = pfhdlist2_out;
        }
        else {
            pfhdref_out = pfhdref_out;
        }
    }
    pfhdcount_out = pfhdcount_out + 1;
    _view.Outputdev_list.append(String.valueOf(pfhdcount_out) + ". " +
matrixn_out[pfhdlist_out][0] + " | " + matrixn_out[pfhdlist_out][1] +
" | PL " + matrixn_out[pfhdlist_out][2] + " | PFHd " +
String.valueOf(matrixv_out[pfhdlist_out][0]) + " | MTTFd " +
String.valueOf(matrixv_out[pfhdlist_out][1]) + " | Technology " +
matrixn_out[pfhdlist_out][3] + " | B10d " +
String.valueOf(matrixv_out[pfhdlist_out][2]) + " | nop " +
String.valueOf(matrixv_out[pfhdlist_out][3]) + " | T10d " +
String.valueOf(matrixv_out[pfhdlist_out][4]) + " | Price " +
String.valueOf(matrixv_out[pfhdlist_out][5]) + " €" + "\n");
    _view.Outputdev_list.append("-----
-----" +
"\n");
    matrixn_out[pfhdlist_out][0] = "o";

```



```
matrixn_out[pfhdlst_out][1] = "o";
matrixn_out[pfhdlst_out][2] = "o";
matrixn_out[pfhdlst_out][3] = "o";
matrixv_out[pfhdlst_out][0] = 0;
matrixv_out[pfhdlst_out][1] = 0;
matrixv_out[pfhdlst_out][2] = 0;
matrixv_out[pfhdlst_out][3] = 0;
matrixv_out[pfhdlst_out][4] = 0;
matrixv_out[pfhdlst_out][5] = 0;
}
```

B13.3.4. Ordenación según MTTF_d.

```
if (MTTFdlist_out == true) {
    Deflist_out = false;
    PLlist_out = false;
    PFHdlst_out = false;
    Techlist_out = false;
    Pricelist_out = false;
}

_view.Outputdev_list.setText("");
mttfdcount_out = 0;
mttfdcount2_out = 0;

// Copy of original output device matrix.
for (mttfdlist3_out = 0; mttfdlist3_out < 100; mttfdlist3_out ++) {
    if (!outputprint_n[mttfdlist3_out][0].equals("o")) {
        matrixn_out[mttfdlist3_out][0] = outputprint_n[mttfdlist3_out][0];
        matrixn_out[mttfdlist3_out][1] = outputprint_n[mttfdlist3_out][1];
        matrixn_out[mttfdlist3_out][2] = outputprint_n[mttfdlist3_out][2];
        matrixn_out[mttfdlist3_out][3] = outputprint_n[mttfdlist3_out][3];
        matrixv_out[mttfdlist3_out][0] = outputprint_v[mttfdlist3_out][0];
        matrixv_out[mttfdlist3_out][1] = outputprint_v[mttfdlist3_out][1];
        matrixv_out[mttfdlist3_out][2] = outputprint_v[mttfdlist3_out][2];
        matrixv_out[mttfdlist3_out][3] = outputprint_v[mttfdlist3_out][3];
        matrixv_out[mttfdlist3_out][4] = outputprint_v[mttfdlist3_out][4];
        matrixv_out[mttfdlist3_out][5] = outputprint_v[mttfdlist3_out][5];
        mttfdcount2_out = mttfdcount2_out + 1;
    }
    else {
        break;
    }
}

// Sort according to MTTFd.
mttfdlist_out = 0;
for (mttfdlist4_out = 0; mttfdlist4_out < mttfdcount2_out;
mttfdlist4_out++){
    mttfdref_out = -1;
    for (mttfdlist2_out = 0; mttfdlist2_out < 100; mttfdlist2_out++) {
```

```

    if (mttfdref_out < matrixv_out[mttfdlist2_out][1]) {
        mttfdref_out = matrixv_out[mttfdlist2_out][1];
        mttfdlist_out = mttfdlist2_out;
    }
    else {
        mttfdref_out = mttfdref_out;
    }
}
mttfdcount_out = mttfdcount_out + 1;
_view.Outputdev_list.append(String.valueOf(mttfdcount_out) + ". " +
matrixn_out[mttfdlist_out][0] + " | " + matrixn_out[mttfdlist_out][1]
+ " | PL " + matrixn_out[mttfdlist_out][2] + " | PFHd " +
String.valueOf(matrixv_out[mttfdlist_out][0]) + " | MTTFd " +
String.valueOf(matrixv_out[mttfdlist_out][1]) + " | Technology " +
matrixn_out[mttfdlist_out][3] + " | B10d " +
String.valueOf(matrixv_out[mttfdlist_out][2]) + " | nop " +
String.valueOf(matrixv_out[mttfdlist_out][3]) + " | T10d " +
String.valueOf(matrixv_out[mttfdlist_out][4]) + " | Price " +
String.valueOf(matrixv_out[mttfdlist_out][5]) + " €" + "\n");
_view.Outputdev_list.append("-----
-----" +
"\n");
matrixn_out[mttfdlist_out][0] = "o";
matrixn_out[mttfdlist_out][1] = "o";
matrixn_out[mttfdlist_out][2] = "o";
matrixn_out[mttfdlist_out][3] = "o";
matrixv_out[mttfdlist_out][0] = 0;
matrixv_out[mttfdlist_out][1] = 0;
matrixv_out[mttfdlist_out][2] = 0;
matrixv_out[mttfdlist_out][3] = 0;
matrixv_out[mttfdlist_out][4] = 0;
matrixv_out[mttfdlist_out][5] = 0;
}

```

B13.3.5. Ordenación según tecnología.

```

if (Techlist_out == true) {
    Deflist_out = false;
    PLlist_out = false;
    PFHdlist_out = false;
    MTTFdlist_out = false;
    Pricelist_out = false;
}

_view.Outputdev_list.setText("");
techcount_out = 0;

for (techlist_out = 0; techlist_out < 100; techlist_out++) {
    if (outputprint_n[techlist_out][3].equals("Hydraulic")) {
        techcount_out = techcount_out + 1;
    }
}

```

```

        _view.Outputdev_list.append(String.valueOf(techcount_out) + ". " +
outputprint_n[techlist_out][0] + " | " +
outputprint_n[techlist_out][1] + " | PL " +
outputprint_n[techlist_out][2] + " | PFHd " +
String.valueOf(outputprint_v[techlist_out][0]) + " | MTTFd " +
String.valueOf(outputprint_v[techlist_out][1]) + " | Technology " +
outputprint_n[techlist_out][3] + " | B10d " +
String.valueOf(outputprint_v[techlist_out][2]) + " | nop " +
String.valueOf(outputprint_v[techlist_out][3]) + " | T10d " +
String.valueOf(outputprint_v[techlist_out][4]) + " | Price " +
String.valueOf(outputprint_v[techlist_out][5]) + " €" + "\n");
        _view.Outputdev_list.append("-----
-----
-----" +
"\n");
    }
    else {
        techcount_out = techcount_out + 0;
    }
}
for (techlist_out = 0; techlist_out < 100; techlist_out++) {
    if (outputprint_n[techlist_out][3].equals("Pneumatic")) {
        techcount_out = techcount_out + 1;
        _view.Outputdev_list.append(String.valueOf(techcount_out) + ". " +
outputprint_n[techlist_out][0] + " | " +
outputprint_n[techlist_out][1] + " | PL " +
outputprint_n[techlist_out][2] + " | PFHd " +
String.valueOf(outputprint_v[techlist_out][0]) + " | MTTFd " +
String.valueOf(outputprint_v[techlist_out][1]) + " | Technology " +
outputprint_n[techlist_out][3] + " | B10d " +
String.valueOf(outputprint_v[techlist_out][2]) + " | nop " +
String.valueOf(outputprint_v[techlist_out][3]) + " | T10d " +
String.valueOf(outputprint_v[techlist_out][4]) + " | Price " +
String.valueOf(outputprint_v[techlist_out][5]) + " €" + "\n");
        _view.Outputdev_list.append("-----
-----
-----" +
"\n");
    }
    else {
        techcount_out = techcount_out + 0;
    }
}
for (techlist_out = 0; techlist_out < 100; techlist_out++) {
    if (outputprint_n[techlist_out][3].equals("Mechanical")) {
        techcount_out = techcount_out + 1;
        _view.Outputdev_list.append(String.valueOf(techcount_out) + ". " +
outputprint_n[techlist_out][0] + " | " +
outputprint_n[techlist_out][1] + " | PL " +
outputprint_n[techlist_out][2] + " | PFHd " +
String.valueOf(outputprint_v[techlist_out][0]) + " | MTTFd " +
String.valueOf(outputprint_v[techlist_out][1]) + " | Technology " +
outputprint_n[techlist_out][3] + " | B10d " +

```

```

String.valueOf(outputprint_v[techlist_out][2]) + " | nop " +
String.valueOf(outputprint_v[techlist_out][3]) + " | T10d " +
String.valueOf(outputprint_v[techlist_out][4]) + " | Price " +
String.valueOf(outputprint_v[techlist_out][5]) + " €" + "\n");
    _view.Outputdev_list.append("-----
-----
-----" +
"\n");
    }
    else {
        techcount_out = techcount_out + 0;
    }
}
for (techlist_out = 0; techlist_out < 100; techlist_out++) {
    if (outputprint_n[techlist_out][3].equals("Electrical")) {
        techcount_out = techcount_out + 1;
        _view.Outputdev_list.append(String.valueOf(techcount_out) + ". " +
outputprint_n[techlist_out][0] + " | " +
outputprint_n[techlist_out][1] + " | PL " +
outputprint_n[techlist_out][2] + " | PFHd " +
String.valueOf(outputprint_v[techlist_out][0]) + " | MTTFd " +
String.valueOf(outputprint_v[techlist_out][1]) + " | Technology " +
outputprint_n[techlist_out][3] + " | B10d " +
String.valueOf(outputprint_v[techlist_out][2]) + " | nop " +
String.valueOf(outputprint_v[techlist_out][3]) + " | T10d " +
String.valueOf(outputprint_v[techlist_out][4]) + " | Price " +
String.valueOf(outputprint_v[techlist_out][5]) + " €" + "\n");
        _view.Outputdev_list.append("-----
-----
-----" +
"\n");
    }
    else {
        techcount_out = techcount_out + 0;
    }
}
for (techlist_out = 0; techlist_out < 100; techlist_out++) {
    if (outputprint_n[techlist_out][3].equals("Electronics")) {
        techcount_out = techcount_out + 1;
        _view.Outputdev_list.append(String.valueOf(techcount_out) + ". " +
outputprint_n[techlist_out][0] + " | " +
outputprint_n[techlist_out][1] + " | PL " +
outputprint_n[techlist_out][2] + " | PFHd " +
String.valueOf(outputprint_v[techlist_out][0]) + " | MTTFd " +
String.valueOf(outputprint_v[techlist_out][1]) + " | Technology " +
outputprint_n[techlist_out][3] + " | B10d " +
String.valueOf(outputprint_v[techlist_out][2]) + " | nop " +
String.valueOf(outputprint_v[techlist_out][3]) + " | T10d " +
String.valueOf(outputprint_v[techlist_out][4]) + " | Price " +
String.valueOf(outputprint_v[techlist_out][5]) + " €" + "\n");
        _view.Outputdev_list.append("-----
-----
-----" +

```

```

"\n");
}
else {
    techcount_out = techcount_out + 0;
}
}
for (techlist_out = 0; techlist_out < 100; techlist_out++) {
    if (outputprint_n[techlist_out][3].equals("Programmable
electronics")) {
        techcount_out = techcount_out + 1;
        _view.Outputdev_list.append(String.valueOf(techcount_out) + ". " +
outputprint_n[techlist_out][0] + " | " +
outputprint_n[techlist_out][1] + " | PL " +
outputprint_n[techlist_out][2] + " | PFHd " +
String.valueOf(outputprint_v[techlist_out][0]) + " | MTTFd " +
String.valueOf(outputprint_v[techlist_out][1]) + " | Technology " +
outputprint_n[techlist_out][3] + " | B10d " +
String.valueOf(outputprint_v[techlist_out][2]) + " | nop " +
String.valueOf(outputprint_v[techlist_out][3]) + " | T10d " +
String.valueOf(outputprint_v[techlist_out][4]) + " | Price " +
String.valueOf(outputprint_v[techlist_out][5]) + " €" + "\n");
        _view.Outputdev_list.append("-----
-----
-----" +
"\n");
}
else {
    techcount_out = techcount_out + 0;
}
}
}

```

B13.3.6. Ordenación según precio.

```

if (Pricelist_out == true) {
    MTTFdlist_out = false;
    Deflist_out = false;
    PLlist_out = false;
    PFHdlist_out = false;
    Techlist_out = false;
}

_view.Outputdev_list.setText("");
pricecount_out = 0;
pricecount2_out = 0;
// Copy of original output device matrix.
for (pricelist3_out = 0; pricelist3_out < 100; pricelist3_out ++) {
    if (!outputprint_n[pricelist3_out][0].equals("o")) {
        matrixn_out[pricelist3_out][0] = outputprint_n[pricelist3_out][0];
        matrixn_out[pricelist3_out][1] = outputprint_n[pricelist3_out][1];
        matrixn_out[pricelist3_out][2] = outputprint_n[pricelist3_out][2];
        matrixn_out[pricelist3_out][3] = outputprint_n[pricelist3_out][3];
    }
}

```

```

matrixv_out[pricelist3_out][0] = outputprint_v[pricelist3_out][0];
matrixv_out[pricelist3_out][1] = outputprint_v[pricelist3_out][1];
matrixv_out[pricelist3_out][2] = outputprint_v[pricelist3_out][2];
matrixv_out[pricelist3_out][3] = outputprint_v[pricelist3_out][3];
matrixv_out[pricelist3_out][4] = outputprint_v[pricelist3_out][4];
matrixv_out[pricelist3_out][5] = outputprint_v[pricelist3_out][5];
pricount2_out = pricount2_out + 1;
}
else {
    break;
}
}

// Sort according to price.
pricelist_out = 0;
for (pricelist4_out = 0; pricelist4_out < pricount2_out;
pricelist4_out++){
    priceref_out = -1;
    for (pricelist2_out = 0; pricelist2_out < 100; pricelist2_out++) {
        if (priceref_out < matrixv_out[pricelist2_out][5]) {
            priceref_out = matrixv_out[pricelist2_out][5];
            pricelist_out = pricelist2_out;
        }
        else {
            priceref_out = priceref_out;
        }
    }
    pricount_out = pricount_out + 1;
    _view.Outputdev_list.append(String.valueOf(pricount_out) + ". " +
matrixn_out[pricelist_out][0] + " | " + matrixn_out[pricelist_out][1]
+ " | PL " + matrixn_out[pricelist_out][2] + " | PFHd " +
String.valueOf(matrixv_out[pricelist_out][0]) + " | MTTFd " +
String.valueOf(matrixv_out[pricelist_out][1]) + " | Technology " +
matrixn_out[pricelist_out][3] + " | B10d " +
String.valueOf(matrixv_out[pricelist_out][2]) + " | nop " +
String.valueOf(matrixv_out[pricelist_out][3]) + " | T10d " +
String.valueOf(matrixv_out[pricelist_out][4]) + " | Price " +
String.valueOf(matrixv_out[pricelist_out][5]) + " €" + "\n");
    _view.Outputdev_list.append("-----
-----
-----" +
"\n");
    matrixn_out[pricelist_out][0] = "o";
    matrixn_out[pricelist_out][1] = "o";
    matrixn_out[pricelist_out][2] = "o";
    matrixn_out[pricelist_out][3] = "o";
    matrixv_out[pricelist_out][0] = -1;
    matrixv_out[pricelist_out][1] = -1;
    matrixv_out[pricelist_out][2] = -1;
    matrixv_out[pricelist_out][3] = -1;
    matrixv_out[pricelist_out][4] = -1;
    matrixv_out[pricelist_out][5] = -1;
}

```

Anexo C. Estudio de los programas de diseño

El estudio de los programas es llevado a cabo a partir de la comparación del funcionamiento y de las características principales de cuatro herramientas informáticas existentes en el mercado dedicadas al diseño de sistemas de control industriales. A pesar de que los programas relativos a la seguridad en máquinas son en la actualidad escasos, se ha realizado una selección de aquéllos suponiendo un impacto en el campo relevante.

C1. SISTEMA.

SISTEMA es un software desarrollado por el Instituto de Seguridad Ocupacional y Salud (BGIA) de Alemania que se fundamenta en los criterios descritos por el estándar ISO 13849-1. La herramienta permite al usuario definir la estructura de las SRP/CS y calcular de manera automática los parámetros de fiabilidad con diferentes niveles de detalle, incluyendo el nivel de fiabilidad alcanzado. Cualquier modificación en cualquiera de los elementos definidos es reflejada en la interfaz con el impacto en el conjunto del sistema. La particularidad de SISTEMA es la implementación de una jerarquía estructurada en seis niveles: proyecto > función de seguridad > subsistema > canal > bloque > elemento. Los componentes de seguridad y sus parámetros son cargados en el software mediante bibliotecas suministrados por los fabricantes de dichos dispositivos.

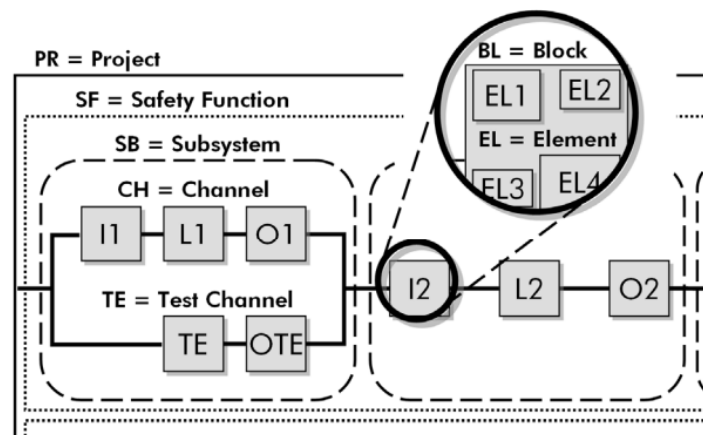


Figura C1.1. Niveles jerárquicos considerados por SISTEMA.

► Más información: <http://www.dguv.de/ifa/praxishilfen/practical-solutions-machine-safety/software-sistema/index.jsp>

C2. Safety Evaluation Tool.

Safety Evaluation Tool es una herramienta online gratuita desarrollada por Siemens y basada en las normas ISO 13849-1 e IEC 62061. El programa permite la creación de funciones de seguridad y su estudio según las normas mencionadas; el usuario es el encargado de definir tanto el estándar de base como los componentes de la función de seguridad, así como su arquitectura. Para ello, el usuario dispone directamente de una biblioteca interna con productos de la marca Siemens, aunque también puede importar bibliotecas de otros fabricantes en ficheros con un formato VDMA 66413. La tasa de fallo es determinada en base a diferentes criterios que el usuario puede escoger (B_{10d} , $MTTF_d$, λ_D , etc.). La definición de cada SRP/CS se realiza en pestañas individuales. Una pestaña adicional permite visualizar en todo momento el valor de nivel de prestaciones o de integridad requerido y compararlo con los niveles alcanzados con los dispositivos de seguridad implementados, actualizándose automáticamente.

► Más información: <http://www.industry.siemens.com/safety-evaluation-tool>

C3. PAScal.

PAScal es una calculadora que permite evaluar el nivel de prestaciones (PL) y el nivel de integridad de seguridad (SIL) de funciones de seguridad relativas a máquinas e instalaciones industriales en función de los componentes que éstas integran. El resultado de este análisis es verificado con los valores de PL y SIL requeridos extraídos de las normas ISO 13849 e IEC 62061, respectivamente. Los componentes de seguridad son introducidos en el programa en la forma de bibliotecas con un formato VDMA 66413, si bien la herramienta informática tiene un conversor para adaptar ficheros en otros formatos. Asimismo, un editor permite al usuario crear sus propias bibliotecas de dispositivos.

El software es libre y descargable de manera gratuita, aunque para acceder a una versión completa del programa con funciones adicionales se precisa de una clave de licencia.

► Más información: <https://www.pilz.com/es-ES/eshop/00105002187038/PAScal-Safety-Calculator>

C4. ProSET®.

ProSET® (*Process Safety Evaluation Toolset*) es un programa informático para la evaluación de riesgos y el análisis del ciclo de vida de la seguridad funcional de sistemas industriales, que es desarrollado y comercializado por la consultora británica ESC (*Engineering Safety Consultants*). ProSET® es una familia de tres herramientas:

- **PHA Comp.** Basada en la norma IEC 61822, permite realizar análisis HAZOP y clasificaciones de los riesgos.
- **SIL Comp®.** Basada en las normas IEC 61508 e IEC 61511, permite, por un lado, la determinación del nivel de integridad de seguridad (SIL) mediante una serie de técnicas analíticas tales como gráficos de riesgo, análisis de capas de protección (LOPA) o álgebra booleana completa (CBA). En un segundo plano, permite verificar el nivel de integridad de los sistemas estudiados.
- **MRA Comp.** Basada en las normas IEC 62061 e ISO 13849, permite la evaluación de los riesgos en maquinaria industrial. Actualmente, este módulo se encuentra en desarrollo.

Los ficheros empleados siguen una extensión *.cvs y son completamente transferibles y compatibles entre los distintos módulos presentados.

► **Más información:** <http://www.esc.uk.net/proset-software/>

La Tabla C4.1. recoge el conjunto de elementos principales caracterizando y diferenciando los programas presentados anteriormente, tal que su lectura facilite la comparación entre los mismos y la determinación de sus puntos comunes y contrarios.

	SISTEMA	Safety Evaluation Tool	PAScal	ProSET®
<i>Normativa</i>	ISO	ISO / IEC	ISO / IEC	IEC
<i>Proceso</i>	Definición de los componentes > verificación del PL	Definición de los componentes > verificación del PL o SIL	Definición de los componentes > verificación del PL o SIL	Definición de los componentes > verificación del SIL
<i>Formato biblioteca</i>	*.slb	*.xml	*.xml (admite otros formatos)	*.cvs
<i>Precio</i>	Gratuito	Gratuito (online – requiere crear una cuenta)	Gratuito. Para una versión completa hace falta una licencia	Contactar con empresa

Tabla C4.1. Comparativa de los elementos principales de los distintos programas.

C5. Consideraciones.

Los programas presentados siguen un funcionamiento similar. El usuario parte de un fichero conteniendo una lista de dispositivos de seguridad que utiliza para definir su sistema en la interfaz de usuario, seleccionando del listado introducido aquellos elementos que son de su interés. El programa asimila en cada situación las características de los componentes escogidos, calculando automáticamente el nivel de fiabilidad (o de integridad, según el caso) del sistema y comparándolo con los niveles requeridos según estipula la normativa correspondiente.

Por el contrario, el programa desarrollado con EjsS sigue el funcionamiento opuesto. El usuario no parte de los componentes de seguridad, sino que éstos constituyen las salidas del proceso de diseño. El usuario concibe un sistema, del cual tiene un nivel de fiabilidad requerido. A partir del nivel de fiabilidad requerido, se impone un valor mínimo de probabilidad de fallo peligroso por hora (PFH_d) que el sistema debe cumplir. Mediante diferentes gráficas, el usuario define de qué manera se distribuye este requerimiento de probabilidad, del cual se extrae una lista de dispositivos de seguridad permitiendo alcanzar dichos valores.

Pliego de condiciones

I. Especificaciones de los elementos constitutivos del proyecto.

En esta sección se presentan todos los elementos necesarios para la materialización del proyecto, que se corresponden con aquellos instrumentos empleados para la elaboración de la herramienta informática.

Concepto	Descripción	Modelo
Ordenador	Soporte físico sobre el que se realizarán las tareas de programación y se ejecutarán los programas informáticos pertinentes.	Asus Intel Core i5 CPU 2.5GHz, 4GB RAM. Sistema operativo Windows 7 o versiones posteriores.
Licencia EjsS	Programa informático que sirve de soporte y herramienta principal para las tareas de programación.	EjsS versión 5.2, build 151201
Licencia MS Excel	Programa informático para el estudio de las curvas de PFH _d y MTTF _d .	Microsoft Excel 2016 Office

Tabla I. Elementos constitutivos del proyecto.

II. Reglamentación y normativas aplicables.

La normativa de cumplimiento obligatorio para la consecución del proyecto corresponde a:

- International Standardisation Organization (ISO). *Safety of machinery – Safety-related parts of control systems – Part 1: General principles for design (ISO 13849-1:2006)* [pdf]. 2ª ed. Ginebra (Suiza): ISO, 1 de noviembre de 2006. 103 p.

Presupuesto

El presupuesto para la realización y materialización del presente proyecto se fundamenta en el Estado de Mediciones descrito a continuación. En éste, se exponen cada uno de los elementos que deben adquirirse y en qué cantidad, junto con una breve explicación que lo describe.

I. Estado de mediciones.

Concepto	Descripción	Cantidad	Unidad
Ordenador	Soporte físico sobre el que se realizarán las tareas de programación y se ejecutarán los programas informáticos pertinentes.	1	€
Licencia EjsS	Programa informático que sirve de soporte y herramienta principal para las tareas de programación.	1	€
Licencia MS Excel	Programa informático para el estudio de las curvas de PFH _d y MTTF _d .	1	€
Horas hombre (desarrollo)	Técnicos relacionados con las tareas de programación. Se estima necesario 1 técnico trabajando durante 3 meses.	480 h 1 técnico	€/técnico·hora

Tabla I. Elementos del proyecto empleados.

II. Presupuesto.

Concepto	Cantidad	Coste unitario [€/u]	Coste total [€]
Ordenador	1	500,00	500,00
Licencia EjsS	1	0,00	0,00
Licencia MS Excel	1	79,00	69,00
Horas hombre (desarrollo)	480	30,00	14.400,00
TOTAL			14.979,00

Tabla II. Coste total del programa.

III. Estudio económico.

El impacto de la implantación del programa desarrollado en la actividad industrial es evaluado a partir de una de las nueve funciones de seguridad descritas en el primer caso práctico del Capítulo 6. La función de seguridad escogida está constituida por un interruptor bloqueado, una unidad lógica y un robot industrial. El estudio del caso permite determinar las características teóricas y analíticas de los distintos componentes indicados, éstas últimas determinadas por el intermedio del programa informático elaborado. La Tabla III muestra el precio asociado a cada dispositivo de seguridad, tanto en el caso teórico como analítico, así como la diferencia existente entre ambos productos.

	Entrada [€]	Lógica [€]	Salida [€]	Total [€]
<i>Resultado teórico</i>	315,00	439,45	8.612,47	9.366,92
<i>Resultado analítico</i>	211,36	172,34	8.612,47	8.996,17
<i>Diferencia</i>	103,64	267,11	0,00	370,75

Tabla III. Precio asociado a los componentes de seguridad.

La tabla anterior evidencia un ahorro significativo de la inversión en la adquisición de los componentes de seguridad debido al uso del nuevo programa informático y a la consiguiente optimización de la solución de seguridad.

Por otro lado, el empleo de una nueva herramienta permite igualmente ahorrar en términos de tiempo dedicado al diseño de los sistemas de control. De manera conservativa, se estima que el intervalo de tiempo invertido en el uso del programa pasa de ser aproximadamente una hora, mediante programas de diseño convencionales, a ser 30 minutos, mediante la nueva herramienta, es decir, la dedicación se reduce a la mitad de la duración original. Este hecho se traduce en una inversión menor en cuanto a horas

de ingeniería requeridas. El coste asociado al uso de cada tipo de programa en función del tiempo dedicado a su manejo se muestra en la Tabla IV.

	Tiempo [min]	Coste [€/h]	Total [€]
<i>Programa convencional</i>	60	20,00	20,00
<i>Programa desarrollado</i>	30	20,00	10,00
<i>Diferencia</i>	30	0,00	10,00

IV. Precio asociado al uso del programa.

La Tabla V constituye el resumen general de los costes asociados a la utilización de los dos tipos de programa informático señalados, recogiendo el coste relativo a la explotación de los mismos, así como la inversión relacionada con la adquisición de los componentes de seguridad constuyendo la solución de seguridad encontrada por éstos. El valor resultante en cada situación puede ser entendido como el coste que supone la implantación de una función de seguridad dada.

	Componentes [€]	Uso programa [€]	Total [€]
<i>Programa convencional</i>	9.366,92	20,00	9.386,92
<i>Programa desarrollado</i>	8.996,17	10,00	9.006,17
<i>Diferencia</i>	370,75	10,00	380,75

Tabla V. Precio asociado a la implementación de una función de seguridad.

A partir de los valores mostrados en la tabla precedente y del coste que supone el desarrollo de la herramienta informática, presentado en el Apartado II de este mismo capítulo, puede determinarse que la inversión inicial realizada en el presente proyecto es recuperable a partir de la evaluación y futura implantación de 40 funciones de seguridad.

$$N^{\circ} \text{ funciones} = \frac{\text{Inversión inicial}}{\text{Coste / función}} = \frac{14.979,00 \text{ €}}{380,75 \text{ €/función}} = 39,34 \approx 40 \text{ funciones} \quad (\text{III.1})$$