



GRADO EN INGENIERÍA INFORMÁTICA

TRABAJO FINAL DE GRADO

---

**Desarrollo de una aplicación para la  
consulta y gestión de datos de tuberías**

---

*Autor:*  
Jorge GONZÁLEZ LÓPEZ

*Supervisor:*  
Paco FALOMIR  
*Tutor académico:*  
José Manuel BADIA CONTELLES

Fecha de lectura: 4 de Noviembre de 2016  
Curso académico 2015/2016

## Resumen

La empresa UBE es un grupo industrial líder en el sector químico de la Unión Europea situado en la localidad del Grao de Castellón. Esta empresa necesitaba mejorar la gestión de una serie de archivos denominados *Pipping Class*, de manera que los departamentos de Fiabilidad, Sistemas de Control de Procesos, Oficina Técnica y Compras de la empresa pudieran consultar y gestionar eficientemente estos datos.

De esta necesidad surge este proyecto, el cual consiste en desarrollar una aplicación que permita la consulta eficiente de la información referente a las tuberías que utiliza la empresa. Además de esto, la aplicación también permitirá tanto la gestión de estos datos como la gestión de los usuarios que tengan acceso a dicha aplicación. Cabe destacar que en este proyecto se han realizado todas las fases del desarrollo de una aplicación de gestión de estos aspectos. Para ello se ha seguido una metodología tradicional en cascada. Este proyecto se ha implementado utilizando un servidor de base de datos SQL Server, y MS Access como herramienta para crear la aplicación usando su lenguaje de programación, Visual Basic.

Como resultado, se ha podido mejorar el sistema utilizado previamente por la empresa. Cabe destacar que actualmente esta aplicación está en funcionamiento y que está siendo utilizada principalmente por los departamentos de Inspección y Oficina Técnica. De este modo, tanto la gestión, como la consulta de los datos mejora notablemente ya que anteriormente toda la información estaba dispuesta en una hoja de cálculo con varias pestañas vinculadas entre sí.

La aplicación acabada es capaz de gestionar todos los datos del Pipping Class y consultarlos eficientemente, además de permitir consultar un historial de los cambios realizados por parte del personal de mantenimiento. Finalmente cabe destacar que en el caso de que los usuarios no sepan cómo manejar la aplicación, se ha creado un manual de usuario que se incluye en el Anexo A de esta memoria.

## Palabras clave

Pipping Class, Ratings, SAP, Schedule

## Keywords

Pipping Class, Ratings, SAP, Schedule

# Índice general

<b>1. Introducción</b>	<b>7</b>
1.1. Contexto y motivación del proyecto . . . . .	7
1.2. Objetivos del proyecto . . . . .	11
1.3. Estructura de la memoria . . . . .	12
<b>2. Descripción del proyecto</b>	<b>13</b>
2.1. Arquitectura del Proyecto . . . . .	13
2.2. Tecnologías Utilizadas . . . . .	14
2.2.1. Servidor . . . . .	14
2.2.2. Cliente . . . . .	15
2.2.3. Entorno de Trabajo . . . . .	15
<b>3. Planificación del proyecto</b>	<b>17</b>
3.1. Metodología . . . . .	17
3.2. Definición de Tareas . . . . .	18
3.3. Estimación Temporal . . . . .	19
3.4. Estimación de recursos y costes del proyecto . . . . .	21
3.4.1. Recursos Hardware . . . . .	21
3.4.2. Recursos Software . . . . .	21

3.5. Estimación de Costes . . . . .	22
3.6. Seguimiento del Proyecto . . . . .	22
3.7. Principales Desviaciones Justificadas . . . . .	23
<b>4. Análisis del sistema</b>	<b>25</b>
4.1. Definición de Requisitos . . . . .	25
4.1.1. Casos de Uso . . . . .	25
4.1.2. Requisitos No Funcionales . . . . .	33
<b>5. Diseño</b>	<b>35</b>
5.1. Diseño del Sistema . . . . .	35
5.1.1. Modelo Conceptual . . . . .	35
5.1.2. Modelo Lógico de datos . . . . .	36
5.1.3. Diagrama de clases . . . . .	39
5.2. Diseño del Interfaz . . . . .	39
5.2.1. Prototipos . . . . .	41
5.2.2. Navegación . . . . .	44
<b>6. Implementación y pruebas</b>	<b>45</b>
6.1. Detalles de implementación . . . . .	45
6.1.1. Proceso de Login . . . . .	46
6.1.2. Módulo de Mantenimiento . . . . .	48
6.1.3. Módulo de Administración . . . . .	61
6.1.4. Módulo Historico . . . . .	63
6.1.5. Módulo de Consulta . . . . .	63
6.1.6. Funcionalidades Extra . . . . .	65

6.2. Seguridad de la Aplicación . . . . .	67
6.3. Pruebas . . . . .	68
6.4. Documentación . . . . .	68
6.5. Implantación . . . . .	69
6.6. Formación . . . . .	69
<b>7. Conclusiones</b>	<b>71</b>
7.0.1. Conclusiones Técnicas . . . . .	71
7.0.2. Conclusiones Personales . . . . .	72
<b>A. Manual de Usuario</b>	<b>75</b>
A.1. Índice . . . . .	75
A.2. Diseño General . . . . .	75
A.3. Acceso al Pipping Class . . . . .	76
A.4. Validación de Usuario . . . . .	76
A.5. Pantalla Principal . . . . .	77
A.6. Formulario de Consulta . . . . .	77
A.6.1. Tabla de Conexiones . . . . .	82
A.6.2. Generar Informe Clase . . . . .	83
A.6.3. Información General . . . . .	83
A.7. Histórico . . . . .	84
A.8. Formulario de Mantenimiento . . . . .	85
A.8.1. Mantenimiento de Servicios . . . . .	85
A.8.2. Mantenimiento de Accesorios . . . . .	88
A.8.3. Mantenimiento de Clases . . . . .	100



# Capítulo 1

## Introducción

### 1.1. Contexto y motivación del proyecto

El proyecto cuya propuesta se presenta en este documento se ha desarrollado en la compañía UBE, un grupo industrial líder en el sector químico de la Unión Europea con varias sedes en distintas partes del mundo.



Figura 1.1: Sedes de la empresa UBE

La sede localizada en el Grao de Castellón sobre la cual se ha realizado la estancia en prácticas, básicamente se dedica a la creación de polímeros industriales que otras empresas puedan utilizar. Un ejemplo de estos polímeros, es la caprolactama, que es la molécula que se utiliza para crear nylon, que como es ampliamente conocido, tiene múltiples aplicaciones. Esta empresa como cualquier multinacional, está dividida en varios departamentos. A continuación se listan los departamentos de la sede de Castellón, con los cuales se ha tratado durante la estancia en prácticas:

- Departamento Recursos Humanos
- Departamento de Inspección y Fiabilidad

- Departamento de Compras
- Departamentos de Oficina Técnica
- Departamento de Producción
- Departamento de I+D
- Departamento de Sistemas de Control de Procesos

Este proyecto se ha realizado en el Departamento de Sistemas de Control de Procesos. No obstante, es un proyecto que está diseñado principalmente para que sea usado por parte del departamento de Inspección y Fiabilidad de la empresa con la colaboración de los departamentos de Sistemas de Control de Procesos, Oficina Técnica y Compras.

Este proyecto consiste principalmente en desarrollar una aplicación que permita la consulta eficiente de la información referente a las tuberías que utiliza la empresa. Además de esto, la aplicación también permitirá tanto la gestión de estos datos como la gestión de los usuarios que tengan acceso a la misma.

Para lograr una mejor comprensión de la funcionalidad de la aplicación, a continuación se va explicar detalladamente el funcionamiento de la empresa y los datos con los que trabaja diariamente.

La sede de UBE localizada en el Grao de Castellón se encarga principalmente de crear polímeros industriales. Para poder crear los diferentes polímeros, la planta química está dividida en una serie de zonas en la que cada una se pueden localizar una larga red de tuberías. En la imagen inferior podemos ver la sede del Grao de Castellón.



Figura 1.2: Imagen de la sede de UBE en Castellón

Por cada red de tuberías circula un tipo de **fluido/servicio**, por ejemplo, agua condensada o ácido sulfúrico, a través del cual se generará el polímero deseado. Cabe destacar que cada vez que se quiere generar un nuevo tipo de polímero, el departamento de Oficina Técnica, diseña una nueva línea de tuberías. Esta consiste en una serie de tramos de tuberías que contienen una serie de **accesorios** para poder unirlos y para poder conseguir el estado óptimo en la generación del polímero. Es decir, si el servicio está localizado en la zona A, y para poder conseguir el estado óptimo para generar el polímero, debe llegar a la zona B, el departamento de Oficina Técnica diseña la cantidad de tuberías y accesorios necesarios (bridas, codos, válvulas, juntas..) para obtener el resultado final. También es destacable que según el polímero que se quiera obtener, el servicio circulará con unas **condiciones de temperatura y presión** específicas (ratings) y que tanto las tuberías como los accesorios tendrán una serie de características concretas (material, diámetro, espesor...).

Los servicios que trabajan sobre unas condiciones iniciales de temperatura y presión iguales y que circulan sobre tuberías del mismo material se agrupan en "**Clases**". Por tanto los datos que utiliza la empresa para almacenar todo este tipo de información se llama Pipping Class. Visto de otra manera, este Pipping Class está dividido en una serie de clases sobre las cuales se asignan una serie de servicios (los diferentes fluidos que circulan por las tuberías) y están asociadas a distintos tipos de accesorios según sus condiciones iniciales de temperatura y presión. Resumiendo, el sistema debe permitir tanto la consulta eficiente de la información como su gestión, ya sea de las clases, servicios o accesorios.

A continuación se muestran una imagen representativa de la composición de tuberías y accesorios. En gris podemos ver el depósito que contiene el servicio, en verde las tuberías por las que circula y en azul las válvulas que regulan su presión.



Figura 1.3: Esquema con tuberías y accesorios

El alcance de este proyecto incluye a varios departamentos: Inspección y Fiabilidad, Oficina Técnica y Compras.

Respecto al departamento de Inspección y Fiabilidad, estos serán los encargados de usar la aplicación, ya que principalmente, realizan los procesos de inspección de las tuberías y quieren

saber y consultar en todo momento la información del Pipping Class. Por ello tendrán acceso a toda la funcionalidad de la aplicación.

Respecto a los departamentos de Oficina Técnica y Compras, solo tendrán acceso a la consulta de los datos y al histórico, por si la persona encargada del mantenimiento de la aplicación ha realizado algún cambio que haga que no se encuentre la información deseada.

No obstante, el alcance de este proyecto no incluye la integración completa de esta aplicación con las distintas aplicaciones del departamento de Oficina Técnica y las del departamento de Compras.

Cabe destacar que la integración con Oficina Técnica no se realiza porque su aplicación maneja una gran cantidad de datos y estos están guardados en otro lugar diferente a los utilizados en la nuestra. Esto conlleva que haya que realizar tanto un traslado como una integración de sus datos con los nuestros para que realmente se fusionen las aplicaciones. Por el momento esto quedará fuera de nuestros objetivos debido al coste temporal aunque en futuro se podría plantear. También hay que destacar que la integración con Compras queda fuera del alcance porque su aplicación utiliza una tecnología totalmente distinta a la nuestra.

Retomando la motivación del proyecto, cabe destacar que el principal motivo por el cual se va a realizar esta aplicación, es para reducir, en la medida de lo posible, el error humano. En la actualidad la empresa utiliza un archivo Excel donde tiene toda la información guardada, de manera que este archivo tiene aproximadamente unas 25 pestañas para así poder diferenciar entre unos datos u otros. Es decir, los datos se han ido añadiendo sobre las diferentes hojas de cálculo a medida que se iban obteniendo, generando así muchas pestañas. Además cuando se quería relacionar información en común, se creaban hipervínculos entre las diferentes pestañas y llegaba un punto en que no se encontraba la información deseada o se accedía a información errónea. Estos problemas no se pueden tolerar en un empresa química que trabaja con una gran cantidad de productos peligrosos y donde el mínimo error podría provocar una catástrofe.

En definitiva, esta forma de consultar los datos es poco habitual y frecuente hoy en día, sin contar que es muy poco eficiente (hay mucha información duplicada) y que puede llevar a equívocos. Estos posibles errores son debidos a que las pestañas del archivo Excel se relacionan entre sí a través de hipervínculos y llega un momento en que el usuario está sobrecargado con tanta información que puede no recordar lo que acaba de consultar. En la figura 1.4 se muestra una imagen del Excel utilizado como consulta de información.

Clase de material según producto		MARGEN CORROSION (mm)	PROPUESTA DEFINITIVA	TIPO DE JUNTA						TIPO DE VALVULA: GLOBO						
Símbolo	Servicio			PRESIÓN (DISEÑO) (kg/cm²)						PRESIÓN (DISEÑO) (kg/cm²)						
				P < 25		25 < P < 200		P > 200		P < 25		25 < P < 200				
				TEMPERATURA (DISEÑO) (°C)						TEMPERATURA (DISEÑO) (°C)						
T < 150°C		T > 150°C		T < 150°C		T > 150°C		T < 150°C		T > 150°C		T < 80°C	T > 80°C	T < 150°C	T > 150°C	
35	P	AGUAS MADRES DE AS	0.762	D90	1.3	-	-	-	-	-	-	-	-	-	-	-
36	WW	AGUAS RESIDUALES (WASTE WATER)	1.588	A10	1.3	-	-	-	-	-	-	-	-	-	-	-
37	WW	AGUAS RESIDUALES (WASTE WATER) (AISI-316)	0.762	D90	1.3	-	-	-	-	-	-	-	-	-	-	-
38	AS	AIR & SOLID BRIDAS LMF (ASK)	0.762	D90	1.3	-	-	-	-	-	-	-	-	-	-	-
39	P	AIRE ATMOSFERICO	3.175	A10	1.3	-	-	-	-	-	-	-	-	-	-	-
40	UA	AIRE CALENTE DE SECADO	0.762	D90	1.3	-	-	-	-	-	-	-	-	-	-	-
41	UA	AIRE DE PLANTA	0.762	D90	1.3	-	-	-	-	-	-	-	-	-	-	-
42	UAESP	AIRE DE PLANTA (COMPRESOR SA)	0.762	D90	1.3	-	-	-	-	-	-	-	-	-	-	-
43	P	AIRE DILUION	3.175	A10	1.3	-	-	-	-	-	-	-	-	-	-	-
44	IA	AIRE INSTRUMENTOS (AISI-304)	0.762	D90	1.3	-	-	-	-	-	-	-	-	-	-	-
45	IA	AIRE INSTRUMENTOS (CS)	3.175	A10	1.3	-	-	-	-	-	-	-	-	-	-	-
46	P	AMONIACO (PARA REFRIGERACION)	0.762	D90	2.1	-	2.1	-	-	-	-	-	-	-	-	-
47	P	AMONIACO (PARA REFRIGERACION)	1.588	A09	2.1	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
48	P	AMONIACO GAS	0.762	D90	2.1	2.2	2.1	2.2	-	-	-	-	-	-	-	-
49	P	AMONIACO GAS, BAJA TEMPERATURA LINEA DEL PUERTO	1.588	A17	2.1	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
50	P	AMONIACO LIQUIDO	0.762	D90	2.1	-	2.1	-	-	-	-	-	-	-	-	-
51	P	AMONIACO LIQUIDO	1.588	A09	2.1	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA
52	P	AZUFRE LIQUIDO	3.175	A10	2.2 & 1.9	2.2 & 1.9	2.2 & 1.9	2.2 & 1.9	-	-	-	-	-	-	-	-
53	P	BISULFITO AMONICO	0.762	D90	1.3	-	-	-	-	-	-	-	-	-	-	-
54	BD	BOILER BLOW DOWN	3.175	A10	2.2 & 1.9	2.2 & 1.9	2.2 & 1.9	2.2 & 1.9	-	-	-	-	-	-	-	-
55	BW	BOILER FEED WATER	3.175	A10	2.2 & 1.9	2.2 & 1.9	2.2 & 1.9	2.2 & 1.9	-	-	-	-	-	-	-	-
56	BR	BRINE	3.175	A10	1.4	-	-	-	-	-	-	-	-	-	-	-
57	BR	BRINE (AISI 304)	0.762	D90	1.4	-	-	-	-	-	-	-	-	-	-	-
58	P	CATALIZADOR	0.762	D90	2.1	2.2	2.1	2.2	-	-	-	-	-	-	-	-
59	CH	CHEMICALS (A1)	0.762	D90	1.1 & 1.3	-	-	-	-	-	-	-	-	-	-	-
60	CHF	CHILLED WATER	0.762	D90	1.3	-	-	-	-	-	-	-	-	-	-	-
61	P	CICLOHEXANO	1.588	D90	2.2	2.2	2.2	2.2	-	-	-	-	-	-	-	-

Figura 1.4: Excel utilizado de consulta

Por ello una aplicación como la desarrollada en este proyecto permitirá, además de evitar errores, que el tiempo que anteriormente se dedicaba a la consulta de estos datos, se emplee en la resolución eficiente de los posibles problemas que puedan surgir en la planta Química, de modo que se optimizan los costes de tiempo de trabajo.

## 1.2. Objetivos del proyecto

El principal objetivo de este proyecto es desarrollar una aplicación que permita la consulta eficiente de la información de la empresa, es decir, los fluidos que utiliza la empresa y los diferentes accesorios y tuberías sobre las cuales circulan estos, de manera que se optimice el acceso y se minimice el posible error humano.

Este objetivo principal se puede desglosar en los siguientes objetivos específicos:

1. Diseño de formularios con una interfaz amigable y fácil de utilizar para cualquier usuario, de manera que se facilite la navegación entre pestañas.
2. Almacenar los datos de la empresa en un sistema de base datos, de manera que su programación sea más segura y robusta.
3. Controlar los accesos a la aplicación, es decir, solo podrán utilizar la aplicación los usuarios que tengan una cuenta en la misma, y tengan asignado un rol con permisos para ejecutar dicha aplicación en el servidor de base de datos.
4. Dividir la aplicación en diferentes módulos de manera que cada uno tenga su propia funcionalidad.
5. Realizar un manual de funcionamiento, para que el usuario sepa claramente los pasos que debe seguir para manejar adecuadamente la aplicación.

6. Realizar diferentes informes que permitan obtener la información consultada en un documento para poder imprimirla, y así, se pueda llevar a planta de una manera útil y eficaz. Con esto se consigue una consulta rápida y eficiente en el lugar de trabajo y no existen pérdidas de tiempo porque no es necesario volver al despacho para volver a consultar los datos, ya que, toda la información está presente en el informe impreso.

### **1.3. Estructura de la memoria**

Esta memoria se estructura en los siguientes capítulos:

- En el capítulo 2 se presenta una descripción de las principales tecnologías usadas en el proyecto.
- En el capítulo 3 se muestra la metodología, planificación y seguimiento que se ha realizado del proyecto.
- En el capítulo 4, se detalla el análisis del proyecto realizando los diferentes diagrama de casos de uso a través de la definición de los requisitos.
- En el capítulo 5, se presenta el diseño poniendo especial hincapié en sus diferentes fases y en el diagrama de clases.
- En el capítulo 6, se muestra la implementación de la aplicación en MS Access.
- En el capítulo 7, se muestran tanto las conclusiones técnicas como personales obtenidas a lo largo de la estancia.
- Por último, se incluye un anexo que contiene el manual de usuario desarrollado.

## Capítulo 2

# Descripción del proyecto

A continuación se detallarán una serie de aspectos del proyecto. En primer lugar se explica la arquitectura general del proyecto. Y a continuación, se hace un repaso de las diferentes tecnologías seleccionadas y utilizadas en el proyecto, así como la justificación de su uso.

### 2.1. Arquitectura del Proyecto

El proyecto se ha desarrollado siguiendo una arquitectura de cliente-servidor [1] porque se permite a los usuarios acceder a la misma para realizar actividades o tareas independientes de una manera concurrente. En esta arquitectura, la funcionalidad se separa entre la parte del cliente y la parte del servidor. Dicha funcionalidad se encuentra dividida en tres capas diferenciadas entre sí, las cuales se describirán a continuación:

- **Capa de presentación.** En esta capa se presentan todas las pantallas con las que los diferentes usuarios de la aplicación irán interaccionando. Cuando el usuario accede a la aplicación a través MS Access desde su directorio personal de red, interactúa directamente con esta capa.
- **Capa de lógica.** En esta capa se encuentra toda la funcionalidad a nivel de servidor de la aplicación. En este caso esta capa se encarga de recibir las peticiones del usuario, ejecutar los programas deseados, y de enviar las respuestas tras el proceso. Un ejemplo de esto sería un procedimiento almacenado, en el cual el usuario rellena un formulario de alta. Cuando le dé al botón de aceptar, se enviará la instrucción SQL al servidor que la recogerá y ejecutará el programa interno correspondiente para insertar esta nueva información. Una vez hecho esto se transmitirá de nuevo estos datos a la aplicación de manera que cuando se consulten los datos, esta nueva información estará disponible. Esta capa interconecta las otras dos, tal y como puede verse en la Figura 2.1.
- **Capa de persistencia.** Aquí es donde se accede a toda la información de la aplicación almacenada en la base de datos. La capa de lógica puede modificar, borrar, añadir y consultar esta información.

De forma resumida, podemos definir la arquitectura de esta aplicación como una arquitectura de tres capas. Se puede ver un esquema de la arquitectura en la Figura 2.1.

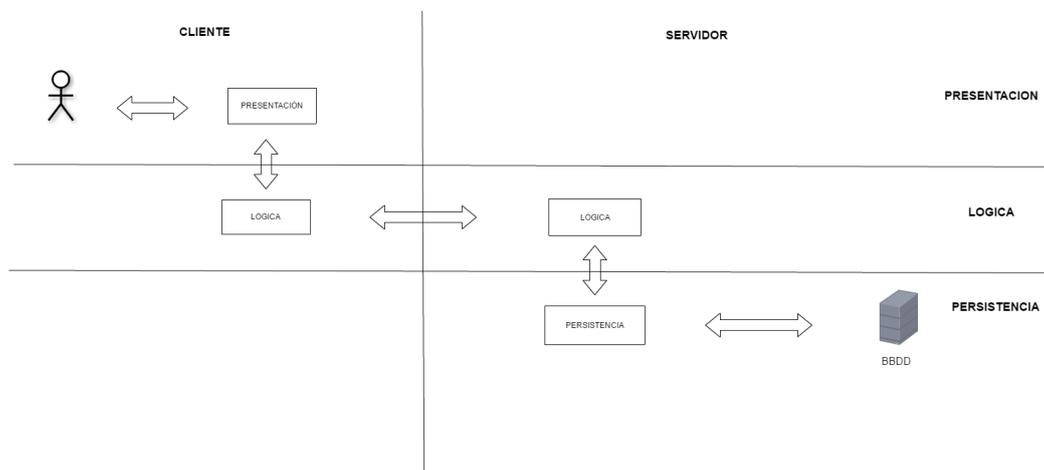


Figura 2.1: Vista general de la arquitectura de la aplicación

## 2.2. Tecnologías Utilizadas

A lo largo del proyecto se han utilizado diversas tecnologías, tanto en el lado del cliente como en el del servidor. Es muy importante elegir bien estas tecnologías, ya que servirán como base para el funcionamiento del mismo. A continuación se detallan las utilizadas, así como la justificación de uso.

### 2.2.1. Servidor

Las tecnologías utilizadas en el lado del servidor son las que se encargan del almacenamiento de los datos de la empresa y gestionan el diseño de la aplicación.

- **SQL Server:** Sistema de base datos que nos permitirá guardar todos los datos del departamento de fiabilidad de la empresa, es decir, el PippingClass. El lenguaje de desarrollo utilizado (por línea de órdenes o mediante la interfaz gráfica de Management Studio) es Transact-SQL (TSQL), una implementación del estándar ANSI del lenguaje SQL, utilizado para manipular y recuperar datos (DML), crear tablas y definir relaciones entre ellas (DDL).

Se decidió usar SQL Server por los siguientes motivos:

- Tiene soporte de transacciones.
- Soporta procedimientos almacenados

- Tiene implementado un entorno gráfico de administración, que permite el uso de órdenes DDL y DML gráficamente.
- Permite trabajar en modo cliente-servidor, donde la información y datos se alojan en el servidor y los terminales o clientes de la red sólo acceden a la información.
- Permite administrar información de otros servidores de datos, es decir, permite importar información de otros servidores de base de datos.

### 2.2.2. Cliente

Las tecnologías utilizadas en el lado del cliente son las que se visualizan y ejecutan directamente sobre la aplicación del usuario.

- MS Access: Es un gestor de datos que recopila información relativa a un asunto o propósito particular, que en nuestro caso es la información contenida en el el Pipping Class. Además esta herramienta nos permitirá diseñar las diferentes pantallas de navegación, los formularios de gestión de los datos (consulta y mantenimiento) y los informes.
- Visual Basic: Lenguaje de programación dirigido por eventos específico de MS Access con el que se ha implementado toda la aplicación [2].

Cabe destacar que se valoraron varias opciones para programar esta aplicación. La primera era realizar una aplicación web y la segunda era realizar la aplicación mediante MS Access. Tras hablarlo con el supervisor de la empresa, se decidió usar MS Access debido a su gran facilidad de uso y a que en los ordenadores de la empresa ya tenían instalada esta herramienta ofimática.

### 2.2.3. Entorno de Trabajo

Este proyecto se ha realizado para el departamento de Inspección y fiabilidad de la empresa y se ha desarrollado e implementado en el departamento de Sistemas de Control de Procesos. Este proyecto ha surgido debido a una necesidad del departamento de fiabilidad, y para saber realmente que es lo que querían, se han hecho varias reuniones con personal de este departamento, incluyendo a su director. También se ha trabajado con el departamento de Oficina Técnica para saber su opinión respecto al aspecto y funcionalidad de la aplicación ya que, después del departamento de Fiabilidad, son los que con más frecuencia van a utilizarla.

Respecto al puesto de trabajo, tenía un ordenador disponible en la sala del departamento de Sistemas de Control de Procesos con las características definidas en el apartado 3.4.1. En este departamento, cada personal se encargaba de realizar funciones diferentes sobre los procesos que se producen en la empresa, por lo que no se ha tenido ninguna colaboración con ellos para poder realizar la aplicación. Simplemente se ha colaborado con el jefe de este departamento, que era mi supervisor, ya que era el único que tenía algunos conocimientos de MS Access.

Respecto a la infraestructura de la empresa, podemos destacar dos aspectos: hardware y software. Si hablamos de la infraestructura hardware de la red, podemos destacar que todos

los puestos de trabajo tienen el mismo ordenador, que es un portátil HP, que se conecta por puerto HDMI a un monitor. Esta empresa tiene varios servidores locales, que se utilizan tanto para realizar copias de seguridad semanales como para establecer las diferentes conexiones de red locales. Cabe destacar que la empresa tiene una intranet, con varios servidores en los que se almacenan diferentes carpetas y que permiten el acceso de los archivos a los diferentes usuarios del sistema según los roles y permisos que tienen en la empresa.

Respecto a la infraestructura software, podemos destacar que todos los ordenadores tienen instalado el paquete ofimático completo de Microsoft Office, Windows 8.1 Enterprise, una herramienta que permite realizar la copia de seguridad semanal y un programa específico que permite a los operarios acceder al ordenador remotamente en el caso de que exista algún tipo de problema software.

Cabe destacar que la empresa tiene contratado personal informático para resolver problemas de tipo hardware o software en el caso de se produzca cualquier tipo de error o fallo. No obstante, este personal no tiene departamento físico en el empresa por lo que los errores de tipo software se corrigen vía escritorio remoto mientras que los problemas vía hardware se resuelven llamando al correspondiente técnico informático.

## Capítulo 3

# Planificación del proyecto

En la fase de planificación del proyecto se han definido los requisitos y se han realizado las correspondientes estimaciones temporales, de costes y de recursos del proyecto. En primer lugar, se explica la técnica de desarrollo de proyectos utilizada. A continuación se concretan las tareas a realizar en el proyecto. Finalmente se realizan estimaciones de los diferentes aspectos que afectan al desarrollo de la aplicación.

### 3.1. Metodología

La metodología de trabajo utilizada en este proyecto se basa en el modelo de desarrollo clásico tomando como referencia en el Project Management Book of Knowledge [3], pero adaptándole algunos aspectos de la metodología ágil Scrum [4]. La razón de esto es que los requisitos han sido establecidos en los diferentes reuniones iniciales celebradas con el personal del departamento de fiabilidad desde el comienzo del proyecto y son invariables. No obstante, existe la posibilidad de realizar cambios menores en cuanto a preferencias estéticas o mejoras funcionales pequeñas. Esta decisión se ha tomado dada a la cercanía del cliente, que en este caso eran los trabajadores del Departamento de Fiabilidad, y a la posibilidad de realizar una estrecha y constante colaboración. Las fases necesarias para seguir esta metodología son:

1. **Análisis de requisitos.** En esta fase se recopilan y refinan los requisitos del sistema. Lo más importante de esta fase es conseguir que los requisitos sean claros y viables. En este proyecto se definieron los requisitos a través de varias reuniones celebradas inicialmente con el departamento de Inspección y Fiabilidad, de manera que, se aclararon las dudas con el cliente y se acordaron unos requisitos finales. Estos requisitos se pueden ver en los Casos de uso (Apartado 4.1.1) y los Requisitos no funcionales (Apartado 4.1.2).
2. **Diseño.** El objetivo de esta fase es utilizar los requisitos recopilados previamente en el análisis para poder realizar un diseño óptimo de la base de datos que permita cumplirlos. Esto permite el inicio de la fase de implementación. En esta fase se realizó el modelo conceptual de la base de datos (Apartado 5.1.1), el modelo lógico de la base de datos (Apartado 5.1.3), el Diagrama de clases (Apartado 5.1.2), y el diseño de la interfaz (Apartado 5.2)

3. **Implementación y validación.** En la implementación se codifican los requisitos refinados en la fase de diseño. Tras esto, es importante comprobar que todo funciona correctamente. Para ello, se realizan pruebas en la fase de validación.
4. **Formación y Documentación.** En esta fase se formó a los usuarios que van a utilizar la aplicación y se realizó un manual de usuario que sirva como ayuda para la utilización correcta de la aplicación.

Estas fases son necesarias para cualquier proyecto software, ya que aportan la documentación imprescindible para cualquier proyecto, ya sea en mayor o menor cantidad. Estos pasos han sido desarrollados en varias iteraciones siguiendo una variante de la metodología scrum, tomando retroalimentación constante por parte del cliente y los usuarios. A medida que ha ido avanzando el proceso de desarrollo de la aplicación, ha sido imprescindible la colaboración por parte de los mismos.

## 3.2. Definición de Tareas

Para poder dividir el proyecto en segmentos más pequeños y estimables, se ha realizado un listado de las tareas concretas que se requieren. En este listado se muestran todas las fases por las que el proyecto debe pasar, y las tareas estimadas inicialmente para el mismo. Para definir con mayor exactitud la planificación, cada fase se ha dividido en una serie de tareas que se listarán bajo el mismo. A continuación se muestran las tareas estimadas como necesarias para este proyecto:

Identificador	Fase	Tarea
T01	Definición de Requisitos	Definir Requisitos de Datos
T02		Crear diagrama de casos de uso
T03		Seguir la variante de la metodología Scrum con tareas a realizar
T04	Análisis	Crear esquema conceptual de la base de datos
T05		Crear diagrama de clases
T06		Validar el análisis
T07	Diseño	Crear prototipos de las interfaces de usuario
T08		Adaptar diagrama de clases con el diseño
T09		Retroalimentación del diseño con el cliente
T10	Implementación	Programación de las funcionalidades esenciales
T11		Programación de módulos adicionales
T12		Pruebas unitarias
T13		Pruebas de aceptación del cliente
T14	Puesta en Marcha	Implantar en el sistema real
T15		Entrega y formación de los usuarios

Cuadro 3.1: Tareas del Proyecto

### 3.3. Estimación Temporal

Para la estimación de la duración del proyecto y sus diferentes fases, se ha realizado una lista de tareas organizadas temporalmente. En este apartado se va a presentar la estimación temporal de la duración de las diferentes fases del proyecto y un desglose de las tareas realizadas en cada fase. Las fases a seguir durante este proyecto son: definición de requisitos, análisis, diseño, desarrollo e implementación y puesta en marcha. En el cuadro 3.2 se muestra más específicamente el desglose de tareas de cada fase, su estimación temporal y la dependencia entre ellas.

Desglose de Tareas			
Número	Tarea	Tiempo(h.)	Dependencias
<b>1</b>	<b>Desarrollo técnico del proyecto</b>	<b>300h</b>	
<b>1.1</b>	<b>Estudio y Formación sobre el proyecto</b>	<b>71h</b>	
1.1.1	Entrevistas para definir el Objetivo de la Aplicación	5	
1.1.2	Definición de requisitos	5	1.1.1
1.1.3	Definir método trabajo y documentación	5	1.1.1
1.1.4	Definir formato y estándar de trabajo	5	1.1.1
1.1.5	Revisar Contexto	8	1.1.1
1.1.6	Identificar alcance y objetivos	8	1.1.1
1.1.7	Estudio y formación de MS Access	15	
1.1.8	Estudio y formación de Visual Basic	20	
<b>1.2</b>	<b>Análisis</b>	<b>28h</b>	
1.2.1	Crear Diagrama de Clases	10	
1.2.2	Documentar Clases	8	1.2.1
1.2.3	Validar Análisis	10	1.2.2
<b>1.3</b>	<b>Diseño</b>	<b>41h</b>	1.2.4
1.3.1	Diseño Conceptual	5	
1.3.2	Diseño Lógico	5	1.3.1
1.3.3	Diseño Físico	5	1.3.2
1.3.4	Aprobación del diseño de la base de datos	3	1.3.3
1.3.5	Rediseño base de datos	5	1.3.4
1.3.6	Diseño de los diferentes prototipos de Navegación	8	
1.3.7	Validar Diseño	10	1.3.6
<b>1.4</b>	<b>Implementación y Validación</b>	<b>150h</b>	
1.4.1	Implementación del diseño de la base datos en SQL Server	3	
1.4.2	Implementación del módulo de consulta	25	1.4.2
1.4.3	Implementación del módulo de administración	15	1.4.3
1.4.4	Gestión de accesos a la aplicación	4	1.4.2
1.4.5	Implementación del módulo de Históricos	7	1.4.4
1.4.6	Implementación del módulo de mantenimiento	30	1.4.5
1.4.6	Integración y unión de los módulos	3	1.4.6
1.4.8	Creación vistas y consultas	15	1.4.7
1.4.9	Creación de informes de Consulta	10	1.4.8
1.4.10	Implementación de funcionalidades extra	20	1.4.9
1.4.11	Pruebas de funcionamiento	10	1.4.10
1.4.12	Creación del manual de la aplicación	8	1.4.11
<b>1.5</b>	<b>Puesta en Marcha</b>	<b>10h</b>	
1.5.1	Implantación	5	
1.5.2	Formación	5	1.5.1

Cuadro 3.2: Lista de tareas

### 3.4. Estimación de recursos y costes del proyecto

A la hora de realizar este proyecto, han sido necesarios una serie de recursos físicos. Estos incluyen los recursos hardware y software involucrados tanto para su realización como para su despliegue. A continuación se detallan las especificaciones técnicas de estos recursos.

#### 3.4.1. Recursos Hardware

**Ordenador de sobremesa.** Este recurso ha sido utilizado para la implementación del proyecto por el alumno y como ordenador local de pruebas. Sus especificaciones técnicas son:

- Procesador Intel® Core i5-4460T con gráficos HD Intel 4600
- Memoria RAM 2 GB DDR3
- Tarjeta gráfica ATI Radeon 9550 256 MD DDR
- Disco duro 200 GB Seagate
- Sistema operativo Windows 8

**Servidor.** Este servidor pertenece al servicio de almacenamiento de la base de datos. Los detalles del tipo de alojamiento son:

- HP ProLiant ML310e G8 XE E3-1220
- 2 TB de espacio.
- Memoria RAM hasta 8 GB.

#### 3.4.2. Recursos Software

Estos recursos engloban todos los programas y sistemas operativos utilizados. A continuación se describen los utilizados:

- **Microsoft Windows 8.** Sistema operativo de licencia privativa utilizado tanto en el servidor como en el computador utilizado para la implementación.
- Microsoft SQL Server
- Visual Basic
- SQL
- Licencia Microsoft Office Completa( Word, Excel Access..)
- Magic Draw

### 3.5. Estimación de Costes

Los costes de un proyecto son un aspecto muy importante a tener en cuenta. Este factor puede influir en si es viable y si se debe seguir adelante con el. Para estimar el coste del proyecto, se ha hecho suposición de un sueldo medio de programador junior en España en 2016 [5] para el alumno autor del proyecto. A continuación se muestra una estimación de los costes:

- Alumno en prácticas:  $1067.40 \text{ euros} \times 3 \text{ meses} = 3202,2 \text{ euros}$

Cabe destacar que sobre los costes estimados no se incluye ni la licencia Windows 8 Pro 64 Bits, ni la licencia completa de Microsoft Office y ni el coste del ordenador sobremesa (CPU, monitor y periféricos), ya que, el ordenador estaba en la empresa antes de empezar la estancia en prácticas. Es decir, no se preparó un equipo a propósito para trabajar sobre este proyecto.

De estos valores calcularemos la contingencia de la Seguridad Social en 2016 para el sueldo, siendo esta del 28,3. Tras estimar los gastos necesarios para realizar el proyecto en cuestión, se puede decir que el coste final ascenderá a la cantidad de 2295,98 euros. Debe recalarse que es un precio estimado y que no incluye gastos adicionales dependientes de riesgos como podría ser una avería del ordenador de sobremesa o sus periféricos.

### 3.6. Seguimiento del Proyecto

A la hora de crear la aplicación, al ser tan cercano el cliente (departamento de Inspección y Fiabilidad), se probaron varias versiones preliminares directamente con los usuarios finales. Estos usuarios se reunían periódicamente conmigo (cada 4 semanas) para dar retroalimentación de posibles defectos o fallos que encontrasen en la aplicación, así como ideas para mejorarla en cuestiones de diseño y funcionalidad de la aplicación. Esta retroalimentación sirvió de mucha ayuda para crear la aplicación a medida para estos usuarios, además de servir como un buen aprendizaje de los gustos en general de los usuarios. Se realizaron un total de tres reuniones. A continuación se resume lo ocurrido en cada una de ellas.

En la primera reunión realizada, al finalizar el primer mes, se reunieron todos los usuarios que iban a utilizar la aplicación para mostrarles una pequeña demostración del uso de la misma. En esta reunión expusieron dudas y sugerencias iniciales. También se mostraron varios ejemplos de gestión de elementos desde cero con el objetivo de ver la correcta funcionalidad de la aplicación como por ejemplo, la inserción de nuevas clases del Pipping Class. Durante esta reunión, se descubrieron unos cuantos detalles que no eran obvios inicialmente, pero que tras esto se implementaron. Uno ejemplo de ello, era si las nuevas clases referentes al servicio de nylon iban a implementarse por que tenían unas nomenclaturas y características diferentes al resto de servicios.

Las siguientes dos reuniones celebradas eran simplemente informativas y servían para mostrar a los departamentos involucrados, como iba progresando el proyecto. Estas dos reuniones fueron bastante breves, y era para comentar algo que viesan que fallaba o alguna propuesta de mejora. Por lo general, se mostraron satisfechos con las funcionalidades que ofrecía la aplicación y con el ahorro de tiempo que involucraba respecto a su sistema antiguo. No obstante, se solicitó realizar un menú lateral que incluyera una serie de funcionalidades extra en la aplicación (Generación de diferentes informes de consulta). Esto se puede ver en el apartado 5.5.1 Aunque estas reuniones fueron breves, se comentó al cliente pequeños bugs que se habían encontrado durante el uso de la aplicación con una simple finalidad informativa. En la segunda reunión se comentó que los bugs encontrados, consistían en fallos de actualización de los datos de manera que cuando se intentaba refrescar la pantalla, no se guardaban los cambios realizados. En la tercera reunión se comentó que los bugs encontrados se debían a que las pantallas diseñadas no se amoldaban correctamente a la resolución de la pantalla cuando su comportamiento debería ser correcto, por lo que se tuvieron que reajustar varios componentes.

Cabe destacar que antes de realizar estas tres reuniones, se llevaron a cabo una serie de reuniones iniciales con el departamento de Inspección y Fiabilidad, para establecer los requisitos que debía de tener la aplicación a desarrollar. En total se realizaron unas 4 reuniones iniciales.

### **3.7. Principales Desviaciones Justificadas**

Sobre este proyecto no se realizó una planificación inicial detallada tal y como aparece en la memoria, pero sí que se planificaron las principales tareas a llevar a cabo y su duración prevista aproximada. Todo ello siguiendo la metodología clásica de desarrollo de un proyecto [6, 7, 8]. No obstante, podemos destacar varias desviaciones que se produjeron durante el desarrollo del proyecto con respecto a las tareas básicas planificadas y a la funcionalidad de ciertas pantallas de la aplicación.

La primera desviación se debe a que tanto las pruebas unitarias como el cifrado de las contraseñas de los usuarios no se pudieron realizar por falta de tiempo. La segunda desviación del proyecto se produjo porque el cliente solicitó implementar una serie de funcionalidades extra, y que se pudieran acceder a estas, desde un menú lateral en el formulario de consulta de la aplicación. Esto supuso un rediseño parcial de la base de datos.



## Capítulo 4

# Análisis del sistema

En esta fase se recopilan los requisitos, y se refinan y clarifican para que resulten óptimos para su utilización para el proyecto. De esta fase se obtienen los casos de uso (Apartado 4.1.1), que son un resultado de este refinamiento de requisitos, así como una serie de requisitos no funcionales (Apartado 4.1.2) que complementan a los casos de uso.

### 4.1. Definición de Requisitos

A la hora de definir los requisitos de la aplicación, es importante tener claros cuales son los objetivos que se quieren conseguir con esta aplicación y que necesidades se pretenden cubrir. Esto ayuda a clarificar las ideas que el cliente pueda tener inicialmente, para guiarle hacia el producto que realmente cubra lo que el necesita sin exceder el presupuesto que se tenga. En este proyecto se tuvo un contacto cercano con el cliente (departamento de Inspección y Fiabilidad), por tanto se pudo clarificar estas necesidades de forma ágil y sencilla. Cabe destacar que la definición de estos requisitos se realizó a través de una serie de reuniones/entrevistas iniciales realizadas con el cliente, de manera que se concretara la funcionalidad de la aplicación.

#### 4.1.1. Casos de Uso

Los casos de uso se definen como las acciones o procesos que el sistema debe poder realizar, formando parte de los requisitos funcionales de un sistema. Estos casos de uso están relacionados a los actores, que son los agentes que pueden interactuar con el sistema. Para esta aplicación se han definido una serie de casos de uso relacionándolos con los actores definidos previamente. Estos actores, disponibles en la tabla 4.1, son los roles que la aplicación permite utilizar de forma interna. Un usuario de la aplicación puede estar ligado a más de un rol.

Actor	Descripción
Consulta	Este tipo de usuarios solo tendrá acceso a la consulta de los datos de la aplicación y al histórico de cambios realizados tanto por las personas de mantenimiento como por el administrador
Mantenimiento	Este usuario será el encargado de realizar el mantenimiento adecuado de todos los datos de la aplicación, ya sea, de los servicios disponibles, de las clases, de los accesorios asociados a dichas clases y de las condiciones iniciales (ratings). Además este usuario podrá acceder a la consulta y al histórico.
Administrador	El administrador tiene acceso y permiso de edición a todos los datos, incluida la gestión de usuarios.

Cuadro 4.1: Diagrama de Casos de Uso

A continuación se puede ver el diagrama de casos de uso de nuestro proyecto:

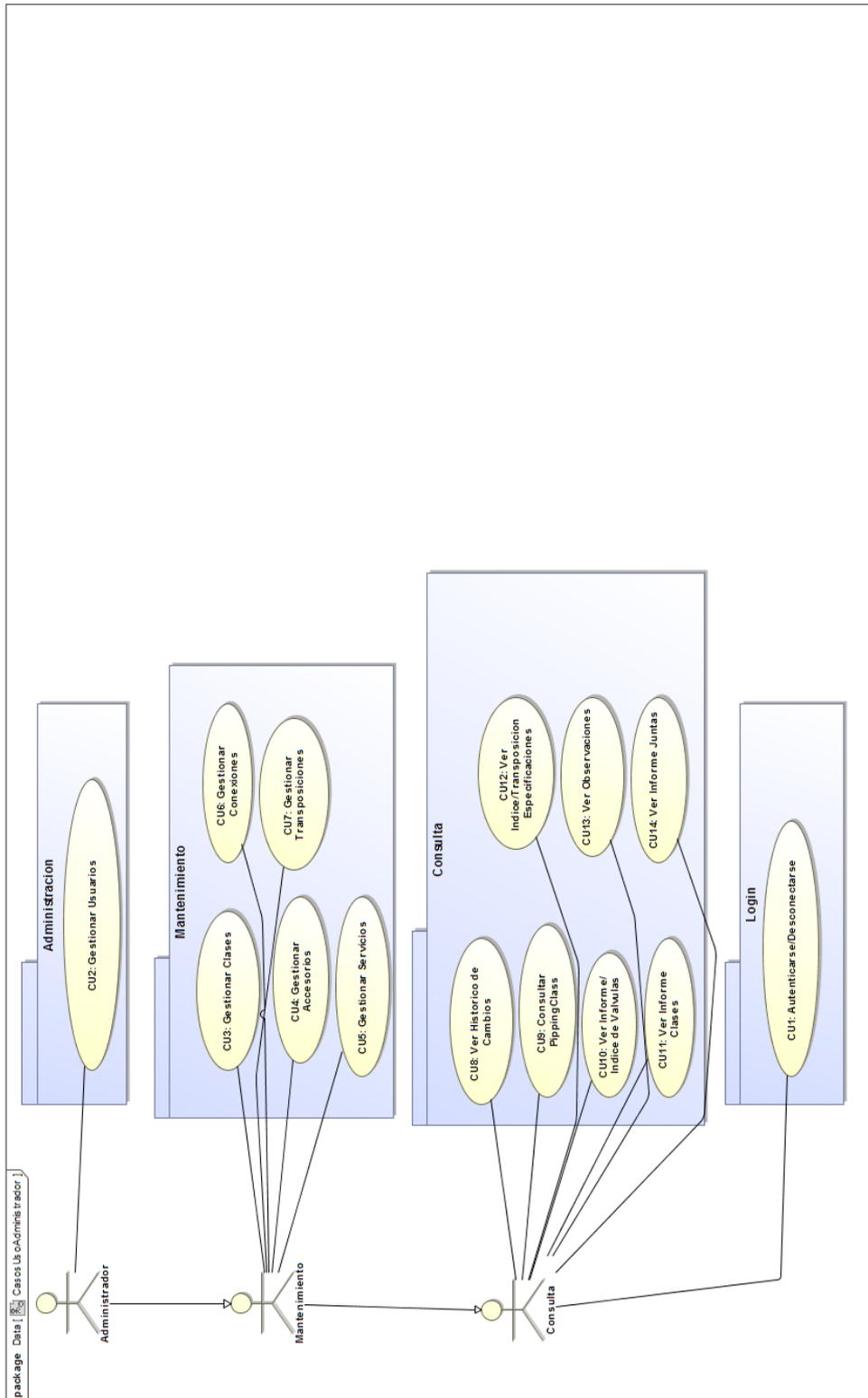


Figura 4.1: Casos de Uso

En las siguientes páginas vamos a describir brevemente cada caso de uso.

Identificador	CU1
Caso de Uso	Autenticarse/Desconectarse
Actores	Administradores, Responsables del Mantenimiento y Usuarios de Consulta
Casos Relacionados	Ninguno
Descripción	El sistema debe permitir a los usuarios acceder a la aplicación usando sus credenciales
Precondición	El usuario debe estar registrado en el sistema
Prioridad	Alta

Cuadro 4.2: Caso de Uso CU1

Identificador	CU2
Caso de Uso	Gestionar Usuarios
Actores	Administradores
Casos Relacionados	Ninguno
Descripción	El sistema debe permitir dar de alta, baja, consultar o modificar a un usuario
Precondición	En caso de alta no debe existir. En caso de baja, consulta o modificación debe existir previamente
Prioridad	Alta

Cuadro 4.3: Caso de Uso CU2

Identificador	CU3
Caso de Uso	Gestionar Clases
Actores	Administradores y Responsables del Mantenimiento
Casos Relacionados	CU4, CU5
Descripción	El sistema debe permitir dar de alta, baja, consultar o modificar las clases del Pipping Class. Cabe destacar que una clase contiene servicios, accesorios y condiciones de temperatura y presión
Precondición	En caso de alta no debe existir. En caso de baja, consulta o modificación debe existir previamente
Prioridad	Alta

Cuadro 4.4: Caso de Uso CU3

Identificador	CU4
Caso de Uso	Gestionar Accesorios
Actores	Administradores y Responsables del Mantenimiento
Casos Relacionados	Ninguno
Descripción	El sistema debe permitir dar de alta, baja, consultar o modificar los accesorios del Pipping Class
Precondición	En caso de alta no debe existir. En caso de baja, consulta o modificación debe existir previamente
Prioridad	Alta

Cuadro 4.5: Caso de Uso CU4

Identificador	CU5
Caso de Uso	Gestionar Servicios
Actores	Administradores y Responsables del Mantenimiento
Casos Relacionados	Ninguno
Descripción	El sistema debe permitir dar de alta, baja, consultar o modificar los servicios del Pipping Class
Precondición	En caso de alta no debe existir. En caso de baja, consulta o modificación debe existir previamente
Prioridad	Alta

Cuadro 4.6: Caso de Uso CU5

Identificador	CU6
Caso de Uso	Gestionar Conexiones
Actores	Administradores y Responsables del Mantenimiento
Casos Relacionados	CU3
Descripción	El sistema debe permitir dar de alta, baja, consultar o modificar las conexiones de cada una de las clases del Pipping Class
Precondición	En caso de alta no debe existir. En caso de baja, consulta o modificación debe existir previamente
Prioridad	Media

Cuadro 4.7: Caso de Uso CU6

Identificador	CU7
Caso de Uso	Gestionar Transposiciones
Actores	Administradores y Responsables del Mantenimiento
Casos Relacionados	CU3
Descripción	El sistema debe permitir dar de alta, baja, consultar o modificar los transposiciones del Pipping Class
Precondición	En caso de alta no debe existir. En caso de baja, consulta o modificación debe existir previamente
Prioridad	Media

Cuadro 4.8: Caso de Uso CU7

Identificador	CU8
Caso de Uso	Ver Histórico de Cambios
Actores	Administradores, Responsables del Mantenimiento y Usuarios de Consulta
Casos Relacionados	CU3, CU4, CU5, CU6, CU7
Descripción	El sistema debe permitir ver un registro que contenga el historial de todos los cambios realizados en las clases del Pipping Class. Estos eventos sirven para tener un control de lo que ocurre en la aplicación y facilitan la búsqueda al usuario en el caso de que no encuentren lo que busquen
Precondición	Ninguna
Prioridad	Alta

Cuadro 4.9: Caso de Uso CU8

Identificador	CU9
Caso de Uso	Consultar Pipping Class
Actores	Administradores, Responsables del Mantenimiento y Usuarios de Consulta
Casos Relacionados	CU3
Descripción	El sistema debe poder dar acceso a consultar toda la información almacenada en la base de datos
Precondición	Para poder consultar la información del Pipping Class anteriormente deber haberse gestionado por parte del personal de mantenimiento
Prioridad	Alta

Cuadro 4.10: Caso de Uso CU9

Identificador	CU10
Caso de Uso	Ver Informe/ Índice Válvulas
Actores	Administradores, Responsables del Mantenimiento y Usuarios de Consulta
Casos Relacionados	CU4
Descripción	El sistema debe permitir generar informes sobre las válvulas almacenadas en el Pipping Class.
Precondición	Las válvulas deben estar registradas previamente. El usuario debe estar registrado previamente.
Prioridad	Alta

Cuadro 4.11: Caso de Uso CU10

Identificador	CU11
Caso de Uso	Ver Informe Clases
Actores	Administradores, Responsables del Mantenimiento y Usuarios de Consulta
Casos Relacionados	CU3
Descripción	El sistema debe permitir generar informes sobre las clases almacenadas en el Pipping Class.
Precondición	Las clases deben estar registradas previamente. El usuario debe estar registrado previamente.
Prioridad	Alta

Cuadro 4.12: Caso de Uso CU11

Identificador	CU12
Caso de Uso	Ver Índice/ Transposición Especificaciones
Actores	Administradores, Responsables del Mantenimiento y Usuarios de Consulta
Casos Relacionados	CU7
Descripción	El sistema debe permitir generar informes sobre las transposiciones almacenadas en el Pipping Class.
Precondición	Las transposiciones deben estar registradas previamente. El usuario debe estar registrado previamente.
Prioridad	Media

Cuadro 4.13: Caso de Uso CU12

Identificador	CU13
Caso de Uso	Ver Observaciones
Actores	Administradores, Responsables del Mantenimiento y Usuarios de Consulta
Casos Relacionados	CU4
Descripción	El sistema debe permitir ver un registro que contenga las observaciones de todos los accesorios Pipping Class.
Precondición	En caso de alta no debe existir. En caso de baja, consulta o modificación debe existir previamente
Prioridad	Media

Cuadro 4.14: Caso de Uso CU13

Identificador	CU14
Caso de Uso	Ver Informe Juntas
Actores	Administradores, Responsables del Mantenimiento y Usuarios de Consulta
Casos Relacionados	CU4
Descripción	El sistema debe permitir generar informes sobre las juntas almacenadas en el Pipping Class.
Precondición	Las juntas deben estar registradas previamente. El usuario debe estar registrado previamente.
Prioridad	Alta

Cuadro 4.15: Caso de Uso CU14

Sobre los requisitos mencionados anteriormente, hay que hacer una mención especial al caso de uso CU7( Gestión de Transposiciones). El cliente solicita que antes de que se haga la migración completa de las clases antiguas a las clases nuevas, almacenar en una tabla dicho cambio. Es decir, que se almacena en una tabla la equivalencia de nombres entre las clases antiguas y las nuevas migradas. Cabe destacar que el cliente lo solicita expresamente por que quiere que las clases antiguas permanezcan por motivos de seguridad. Por este motivo, la gestión de transposiciones se trata como un caso de uso.

#### 4.1.2. Requisitos No Funcionales

A la hora de especificar los requisitos, existen una serie de ellos que no forman parte de las acciones y funcionalidades directas de la aplicación. Por ello, este apartado recoge los requisitos que son imprescindibles para la aplicación, pero que no corresponde a casos de uso.

Identificador	RNF1
Caso de Uso	Diseño Adaptable
Descripción	La interfaz gráfica del sistema debe ser adaptable a las diferentes resoluciones de pantalla.

Cuadro 4.16: Requisito no Funcional RNF1

Identificador	RNF2
Caso de Uso	Aplicación Ligera
Descripción	El sistema debe ser ligero en cuanto en tiempo de conexión, ya que deben mostrar la información al momento ya sea cuando se consulta o cuando se gestiona

Cuadro 4.17: Requisito no Funcional RNF2

Identificador	RNF3
Caso de Uso	Usabilidad de la Aplicación
Descripción	El sistema debe estar diseñado para ser fácilmente usable por todo tipo de usuarios

Cuadro 4.18: Requisito no Funcional RNF3



# Capítulo 5

## Diseño

A la hora de realizar la fase de diseño, se recogen los requisitos refinados en la fase de análisis y se convierten en elementos más cercanos a la fase de implementación [9, 10].

### 5.1. Diseño del Sistema

El diseño del sistema comprende el modelado de los elementos necesarios para el funcionamiento interno de la aplicación, su lógica y su persistencia. De ello se puede extraer los esquemas y conceptos mostrados a continuación.

#### 5.1.1. Modelo Conceptual

En este apartado se muestra el diseño conceptual simplificado de la base de datos definido a través de los requisitos establecidos en la reuniones iniciales.

##### **Definición de Requisitos**

Nuestro cliente quiere desarrollar una aplicación para gestionar la información referente a las tuberías que utiliza. Para ello, nos ha contratado para que hagamos el diseño e implementación de la base de datos que dará soporte a la aplicación de gestión de tuberías.

Los usuarios de la aplicación se clasifican en personal de consulta, personal de mantenimiento y administradores. Un administrador puede realizar las funciones del personal de mantenimiento y de consulta. Un personal de mantenimiento puede realizar las funciones de consulta pero no las del administrador. Y un personal de consulta no puede realizar ni las funciones de administrador ni las funciones del personal del mantenimiento.

Por las red de tuberías que utiliza la empresa circulan un serie de servicios o fluidos sobre los cuales se quiere almacenar tanto su nombre como su símbolo. Estos servicios se agrupan por clases según el material sobre el que circulan y según las condiciones de temperatura y

presión a las que circula. Para crear correctamente una línea de tuberías, existen una serie de accesorios. Sobre estos se quiere almacenar la siguiente información: nombre, tipo, diámetro mínimo, diámetro máximo, schedule (espesor), material, norma, observaciones, extremos, acabado y código. De todos los tipos de accesorios disponibles, sobre las juntas y válvulas se quiere guardar la información de los fabricantes e información característica del accesorio.

Cuando se gestiona la información de la base de datos, debe almacenarse en un historial los cambios realizados. En el histórico hemos de almacenar la información anterior y posterior del cambio, además de quién y cuando se realizó

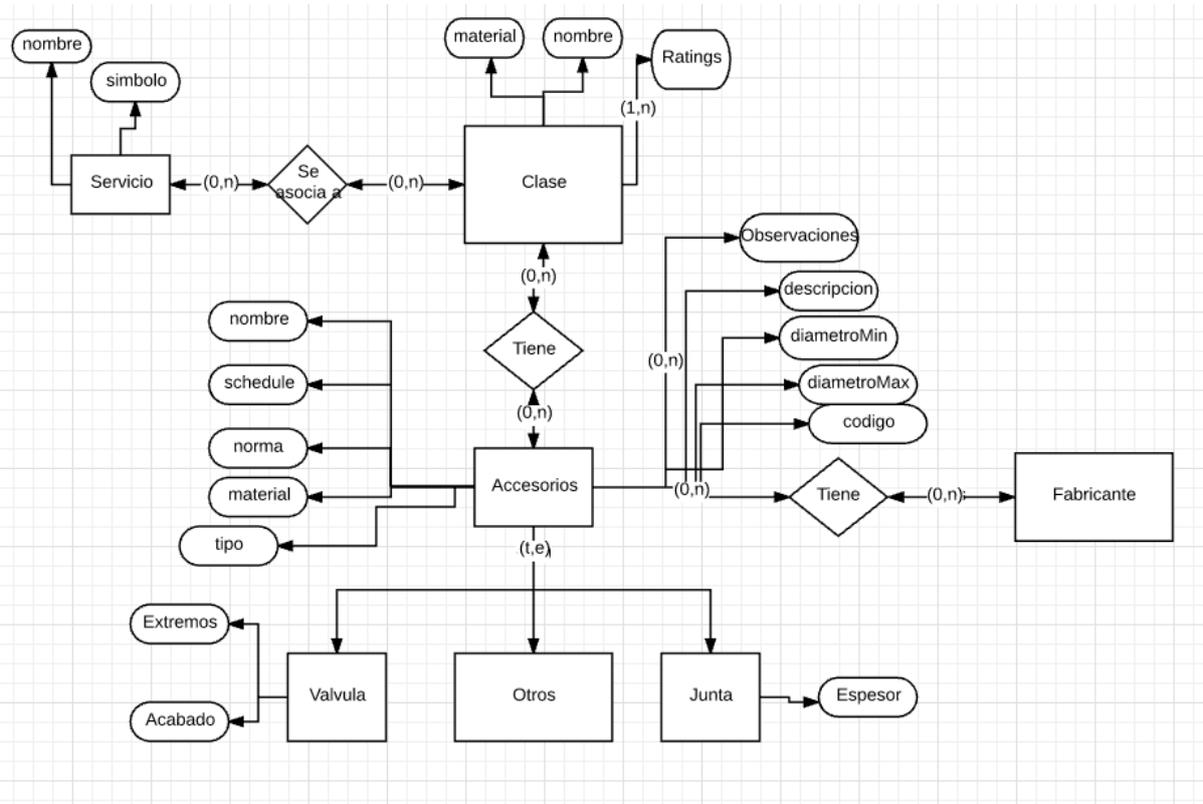


Figura 5.1: Modelo Conceptual Simplificado

### 5.1.2. Modelo Lógico de datos

A la hora de diseñar la base de datos, se han tenido que cumplir ciertas normas impuestas por el framework SQL Server. La más importante es que todas las tablas deben tener un campo llamado ID y que este sea la clave primaria. A pesar de ser a veces innecesario, como es el caso de una relación muchos a muchos, el framework en cuestión obliga a hacer uso de esta clave. Esto es debido a que utiliza una abstracción interna de la base de datos para simplificar su acceso y uso.

Para el diseño del diagrama se ha usado la herramienta incorporada del SQL Server ( SQL Management Studio). Esta herramienta es muy versátil y permite editar de forma gráfica el modelado de una base de datos MySQL. También permite la exportación e importación de la

misma, por tanto ha sido posible extraer las sentencias SQL correspondiente al modelado.

La tabla 5.1 describe brevemente todas las tablas de la base de datos.

Tabla	Descripción
Servicio	Tabla para registrar los servicios que circulan por las tuberías
Clases	Tabla que contiene las clases que agrupan servicios con características comunes. En este caso son tanto condiciones de temperatura y presión 'ratings', como tipo de material de los accesorios
TransposicionClases	Tabla para registrar el cambio de las clases antiguas a las nuevas
Accesorios	Tabla para registrar todos los elementos utilizados en la planta química
Ratings	Tabla que contiene las diferentes condiciones de temperatura y presión sobre las cual se trabaja.
Material	Tabla que contiene los diferentes materiales utilizados de los accesorios.
Junta	Tabla para registrar un tipo de accesorio en concreto
Válvulas	Tabla para registrar un tipo de accesorio en concreto
FabricanteJunta	Tabla que contiene los distintos proveedores de juntas
FabricanteValvula	Tabla que contiene los distintos proveedores de válvulas

Cuadro 5.1: Pipping Class

Cabe destacar que el diseño lógico resultante no se obtiene literalmente a partir del diseño conceptual definido anteriormente, si no que existen varias entidades que se han separado fuera del núcleo relacionado del diseño por cuestión de programación. En este caso tanto las válvulas como las juntas son unos tipos de accesorio específicos con una serie de atributos concretos, que se han dispuesto de forma separada en el diseño lógico por que de esta manera su programación era más sencilla y así se podía abarcar una mayor funcionalidad. Un ejemplo de esto son la asignación de fabricantes sobre estos accesorios específicos, ya que sobre el resto de accesorios, no se quiere informar de ello.

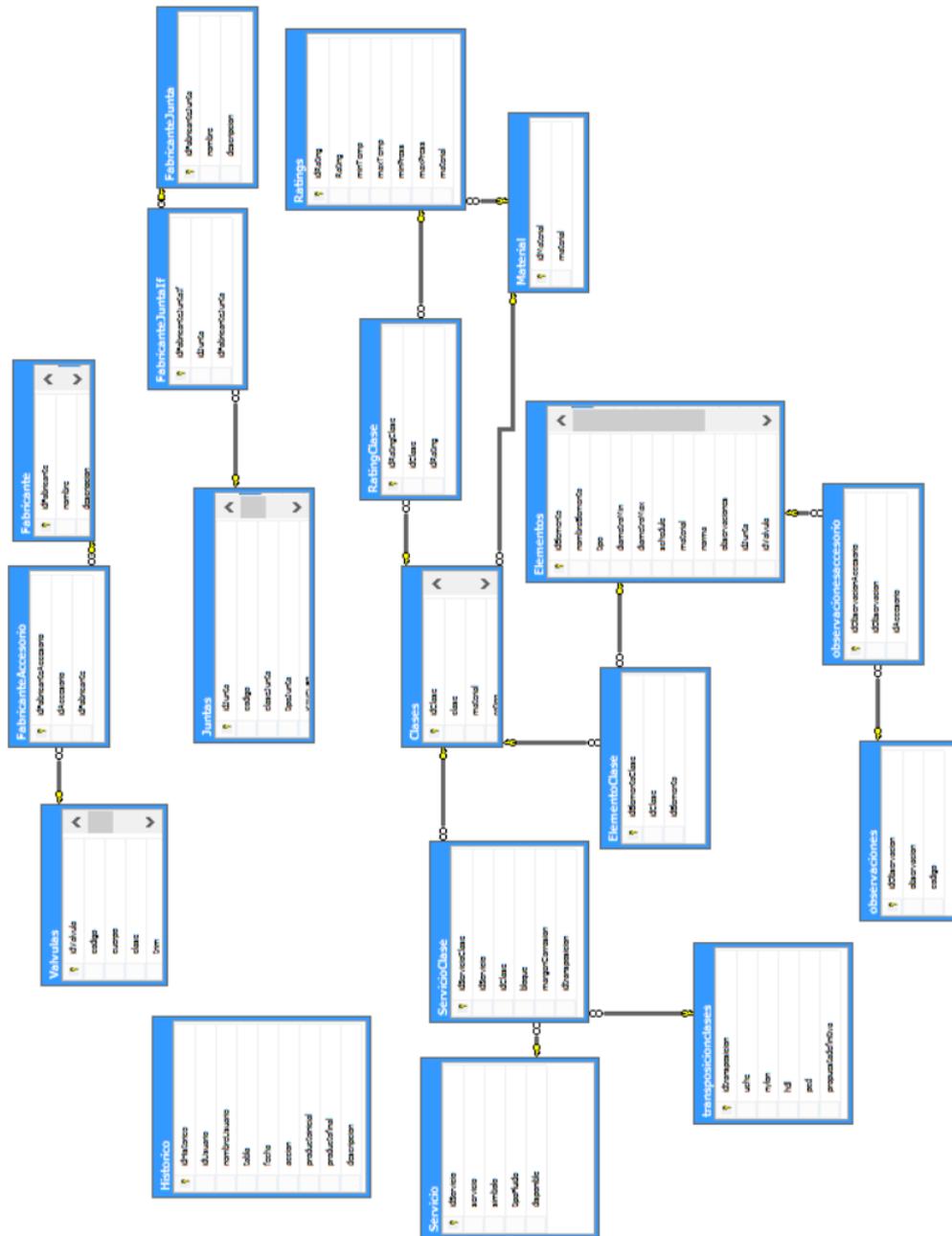


Figura 5.2: Modelo Lógico

### 5.1.3. Diagrama de clases

El diagrama de clases de la aplicación, corresponde directamente con el modelo lógico de la base de datos presentado anteriormente. Habitualmente existen algunas diferencias entre ambos diagramas, al no realizarse una conversión directa de la base de datos a las clases del lenguaje seleccionado. No obstante, en este caso el framework mapea todas las tablas de la base de datos y utiliza un generador para crear una clase correspondiente a cada una de ellas. Por ello, se ha optado por mostrar el diagrama de clases de varias entidades en particular que tienen alguna relación (Figura 5.2), siendo esta extrapolable al resto de entidades de la base de datos.

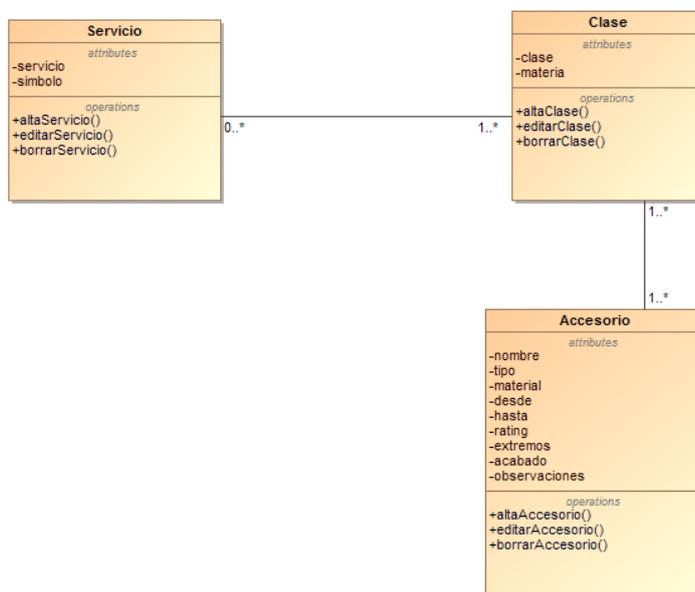


Figura 5.3: Diagrama de Clases

## 5.2. Diseño del Interfaz

A la hora de elegir qué diseño utilizar, era imprescindible el uso de un diseño que fuese simple, intuitivo y adaptable. Para diseñar la interfaz, se siguieron algunas pautas básicas [11, 12]. Estas pautas se describen a continuación:

- **Proximidad.** Los elementos que se agrupan entre sí dan la sensación de pertenecer a un único grupo. Este principio se basa en que el cerebro humano asocia las agrupaciones de elementos a esto, y permite separar categorías de elementos entre sí. Esto se puede apreciar en la agrupación de los campos de los formularios o elementos de los menús.
- **Alineación.** Otro aspecto importante es la alineación de los elementos entre sí. Si los

elementos se posicionan sin tener relación entre sí, da una sensación de caos en la interfaz.

- **Consistencia.** Es importante repetir de forma consistente los tipos de fuentes, los colores y los estilos de la aplicación en general. Si se varía mucho, puede parecer desordenado. En esta aplicación se han utilizado estilos homogéneos a lo largo de la aplicación.
- **Contraste.** Es importante tener contraste entre los elementos del diseño. Elementos que sean similares pueden dar lugar a confusión. En la aplicación, los elementos como los botones contrastan mucho con el fondo para dar esa diferenciación clave.
- **Usabilidad.** Es importante que la aplicación puede ser usada por cualquier tipo de usuario.
- **Dar Control al Usuario.** Es importante que el usuario que está usando la aplicación sienta que en todo momento tiene control sobre la interfaz de usuario.
- **Deshacer Cambios.** Es importante habilitar tanto la opción de deshacer cambios como la de la navegación óptima para dar tanto control al usuario como la sensación de seguridad, de modo que si se equivoca en algo, no sucede nada ya que se pueden revertir los cambios.

Algunos ejemplos que siguen las pautas anteriormente mencionadas son los siguientes:

- La pantalla de consulta se puede dividir en tres partes diferenciadas.
  - *Cabecera.* En esta se encuentra el logo de la empresa, el título, los botones de navegación y la sesión iniciada.
  - *Cuerpo.* Donde se encuentra toda la funcionalidad de la pantalla.
  - *Pie.* En esta se indica alguna información de la empresa.
- Todas las pantallas están alineadas y tienen un tono azulado.
- Los botones de navegación permiten dar control al usuario y permiten revertir los cambios realizados.
- El logo de la empresa siempre está en la parte superior izquierda de las pantallas.

Todos esto se puede ver pantalla de consulta (ver Figura 5.4).

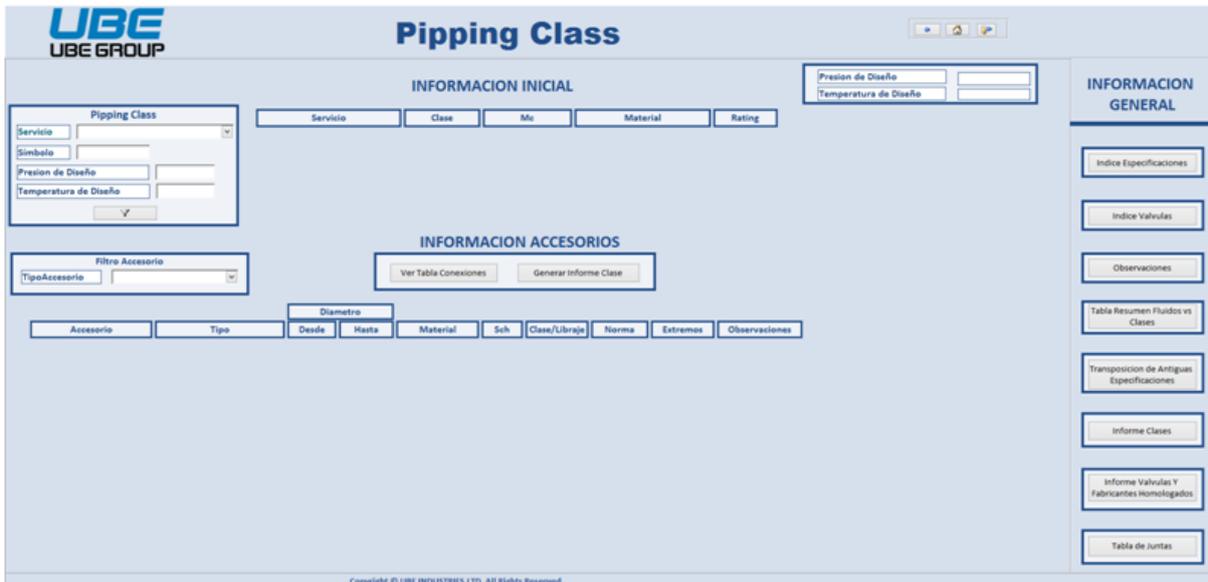


Figura 5.4: Pantalla de Consulta

### 5.2.1. Prototipos

Los prototipos de la aplicación son una primera versión de las interfaces gráficas que van a estar disponibles en la aplicación. Tomándolas como base, se realizaron pequeñas modificaciones en la versión final, pero siempre manteniendo la esencia del prototipo desarrollado. Estos prototipos están mostrados por orden, comenzando por el acceso (Figura 5.5). Esta interfaz está pensada para ser simple y directa, sin distracciones de por medio.

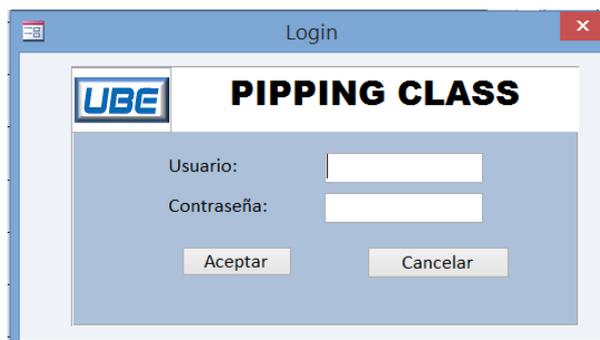


Figura 5.5: Pantalla de Login

Una vez se entra en la aplicación, se accede a la pantalla principal (Figura 5.6) mediante la cual se puede navegar a los distintos componentes de la aplicación. En las imágenes a continuación (ver Figuras 5.7 a 5.10) se puede ver un prototipo de cada uno de los componentes en cuestión.

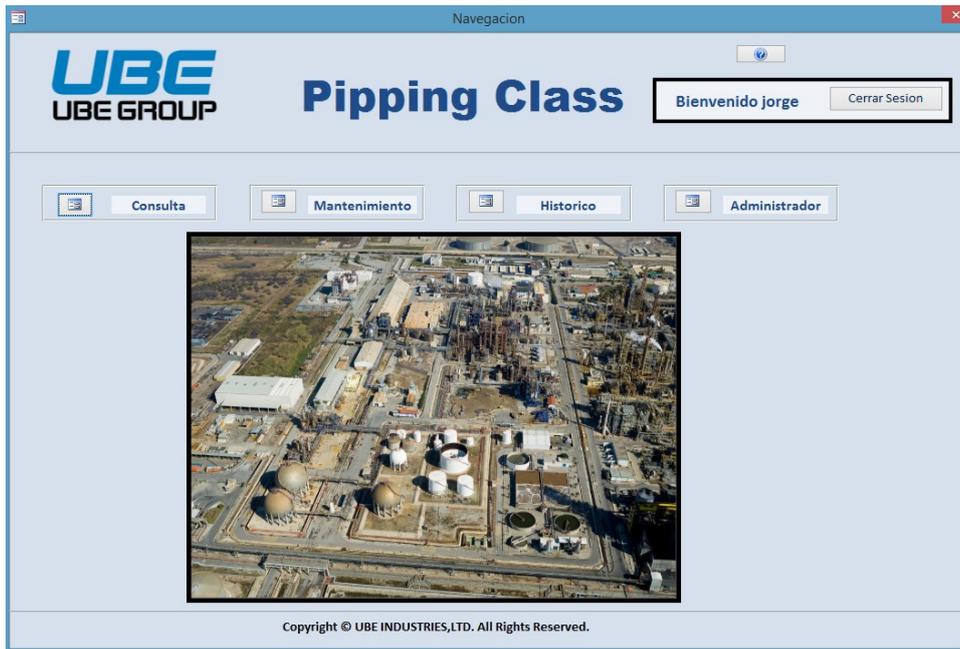


Figura 5.6: Pantalla Principal

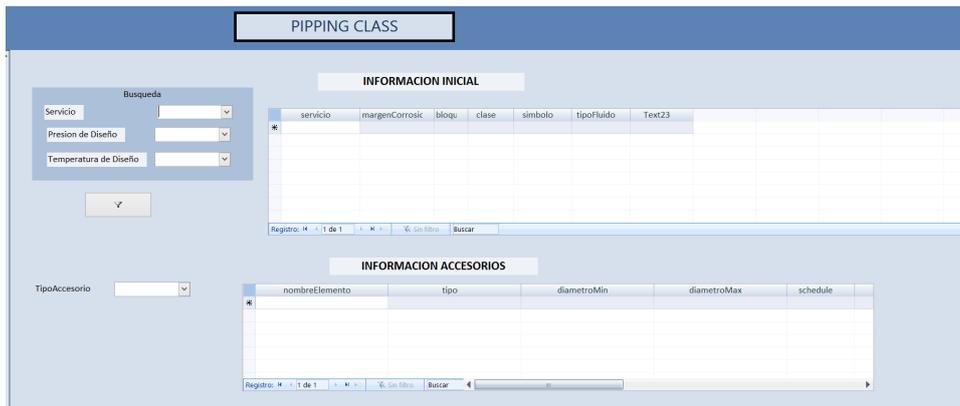


Figura 5.7: Pantalla de Consulta

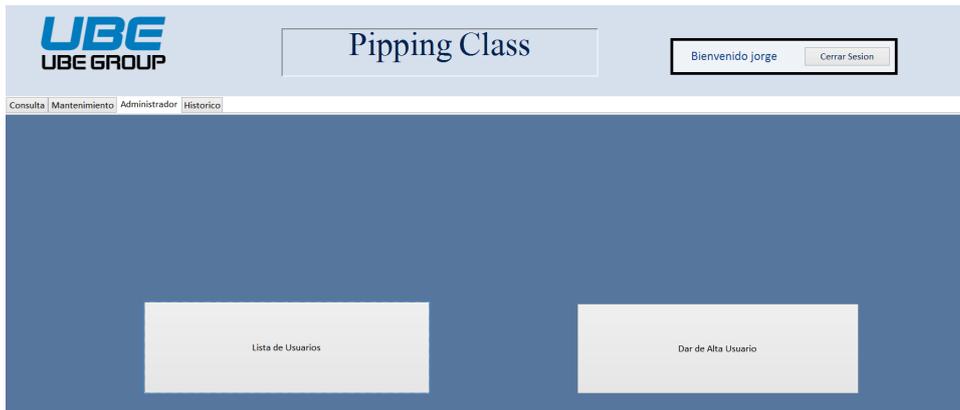


Figura 5.8: Pantalla de Administración

Historico						
nombreUsuario	tabla	accion	fecha	productoinicial	productofinal	descripcion
jorge	Elementos	Insertion			cao	
jorge	Elementos	Insertion	17/12/2015 8:57:07		cosa	
jorge	Elementos	Insertion	17/12/2015 9:23:26		tipo	Se añadido un cap con diametro minimo 12" diametro maximo 22" y de material hierro
jorge	Elementos	Modificar	17/12/2015 9:48:26		tipo	Se añadido un cap con diametro minimo 12" diametro maximo 22" y de material hierro
jorge	Elementos	Modificar	17/12/2015 10:35:58	90	99	Se añadido un cap con diametro minimo 12" diametro maximo 22" y de material hierro
jorge	Elementos	Modificar	17/12/2015 10:40:29			Se añadido un brida con diametro minimo 2" diametro maximo 35" y de material queeq
jorge	Elementos	Modificar	17/12/2015 10:45:28	Diametro Minimo : 12" Material :	Diametro Minimo : 12" Materu	Se ha modificado un cap con diametro minimo 12" diametro maximo 22" y de material f
jorge	Elementos	Borrado	17/12/2015 10:53:13	cao		Se ha borrado un codo con diametro minimo 12 diametro maximo 33 y de material dpe
jorge	Servicio	Insertion	17/12/2015 12:20:49		Nitrato Sodico	Se añadido el servicio Nitrato Sodico con el simbolo No2
jorge	Servicio	Modificar	17/12/2015 12:21:02	Simbolo : No2	Simbolo : N	Se ha modificado el servicio Nitrato Sodico con simbolo No2 por el servicio Nitrato
jorge	Servicio	Borrado	17/12/2015 12:21:10	Agua Caliente		Se ha borrado el servicio Agua Caliente con simbolo HW
jorge	Servicio	Borrado	17/12/2015 12:24:23	ewr		Se ha borrado el servicio ewr con simbolo epeq
jorge	Servicio	Modificar	17/12/2015 13:24:02	Simbolo : fdf	Simbolo : yt	Se ha modificado el servicio few con simbolo fdf por el servicio few con simbolo yt
jorge	Servicio	Insertion	17/12/2015 13:24:40		tres	Se añadido el servicio tres con el simbolo uy
jorge	Servicio	Borrado	17/12/2015 13:24:50	tres		Se ha borrado el servicio tres con simbolo uy
jorge	Servicio	Insertion	18/12/2015 8:39:03		Acido Nitrico	Se añadido el servicio Acido Nitrico con el simbolo N
jorge	Servicio	Modificar	18/12/2015 8:39:20	Simbolo : N	Simbolo : No2	Se ha modificado el servicio Acido Nitrico con simbolo N por el servicio Acido Nitr
jorge	Servicio	Borrado	18/12/2015 8:39:30	Acido Nitrico		Se ha borrado el servicio Acido Nitrico con simbolo N
jorge	Servicio	Insertion	18/12/2015 9:21:36		Nitrico	Se añadido el servicio Nitrico con el simbolo No2
jorge	Elementos	Insertion	21/12/2015 10:36:14		tyu	Se añadido un brida con diametro minimo 12" diametro maximo 22" y de material AS
jorge	Servicio	Insertion	21/12/2015 11:11:43		a2	Se añadido el servicio a2 con el simbolo a2
jorge	Servicio	Insertion	21/12/2015 11:11:55		a3	Se añadido el servicio a3 con el simbolo a3
jorge	Servicio	Insertion	21/12/2015 11:12:52		a4	Se añadido el servicio a4 con el simbolo a4
jorge	Servicio	Insertion	21/12/2015 11:13:03		a5	Se añadido el servicio a5 con el simbolo a5
jorge	Servicio	Borrado	21/12/2015 11:13:24	a5		Se ha borrado el servicio a5 con simbolo a5

Figura 5.9: Pantalla del Histórico

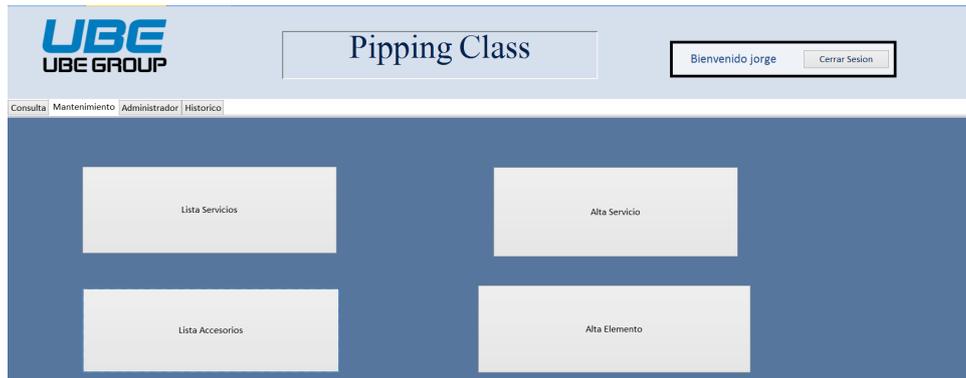


Figura 5.10: Pantalla de Mantenimiento

Cabe destacar que estos prototipos se presentaron inicialmente al cliente para obtener su aprobación y para saber si le parecía correcta la disposición y el funcionamiento de las pantallas.

### 5.2.2. Navegación

En este apartado se va presentar el correspondiente sitemap (mapa de navegación) de la aplicación.

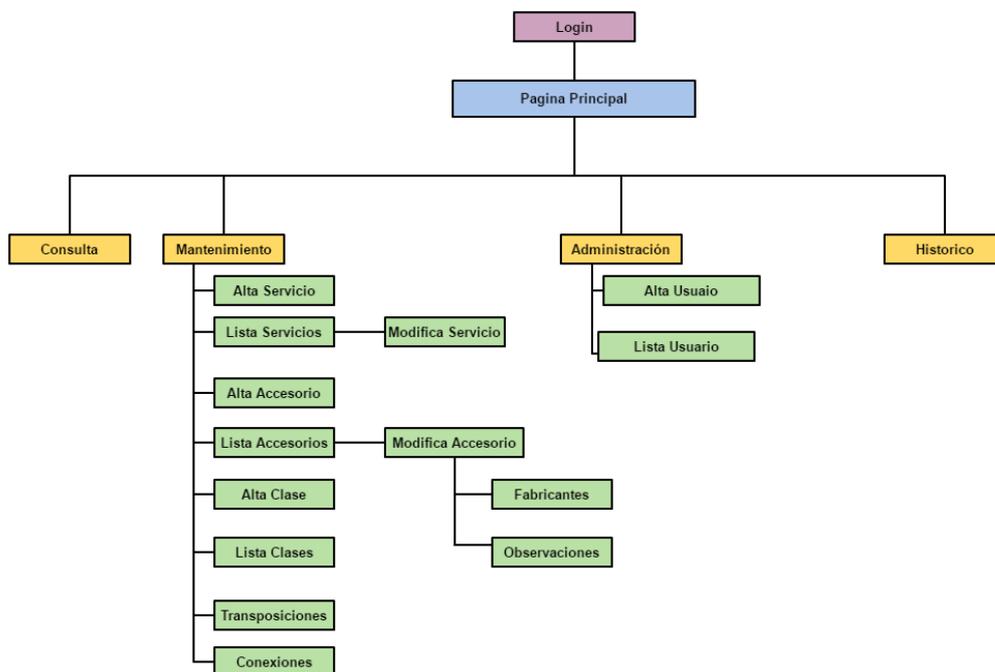


Figura 5.11: Mapa de Navegación

## Capítulo 6

# Implementación y pruebas

En este capítulo se explica el proceso llevado a cabo para la elaboración del producto final del proyecto. También se comprueba si el producto diseñado es el que realmente necesita el cliente, o por el contrario, se ha invertido tiempo y recursos en un producto carente de utilidad. Además se va a explicar la implementación de cada uno de los módulos como su interconexión y una serie de funcionalidades extra.

### 6.1. Detalles de implementación

En este apartado vamos a explicar la implementación de algunos de los procesos incluidos en cada uno de los diferentes módulos implementados a lo largo de la aplicación. Los módulos implementados en la aplicación son:

- **Módulo de Consulta:** Módulo que permite la consulta de los todos los elementos implementados en la aplicación.
- **Módulo de Mantenimiento:** Módulo que permite la gestión completa de todos elementos de la aplicación.
- **Módulo de Histórico:** Módulo que permite la consulta del histórico de cambios
- **Módulo de Administración :** Módulo que permite la gestión tanto de los usuarios como de sus roles que van a acceder a la aplicación.

Cabe destacar que estos módulos se han implementado utilizando el lenguaje que utiliza internamente MS Access que en este caso es Visual Basic, que cómo ya habíamos recordado en el apartado 2.2.2, Visual Basic es un lenguaje de programación dirigido por eventos específicos. Un ejemplo de este funcionamiento sería el siguiente:

- Se designa como objeto cualquier elemento, por ejemplo, un formulario, una imagen, un control, tal como una caja de texto. Estos objetos tienen propiedades, que en el caso de la caja de texto una de ellas es la propiedad "text" que se encarga de contener el texto que aparecerá en la caja. A los objetos se les puede asociar eventos que comúnmente, son las acciones que realizan los usuarios sobre los objetos, que como resultado pueden, por ejemplo, provocar un cambio en alguna propiedad de un objeto. Por ejemplo: Visual Basic tiene un evento llamado KeyPress, que ocurre cuando el usuario presiona una tecla; ese evento se puede asociar a la caja de texto, y en él definirá (por programación) qué acción se tomará cuando se oprima una tecla.

A continuación vamos a explicar cómo se han programado cada uno de los módulos individualmente.

### 6.1.1. Proceso de Login

En la pantalla de login podemos destacar varias funciones esenciales, que nos permitirán gestionar la mayor parte de la aplicación.

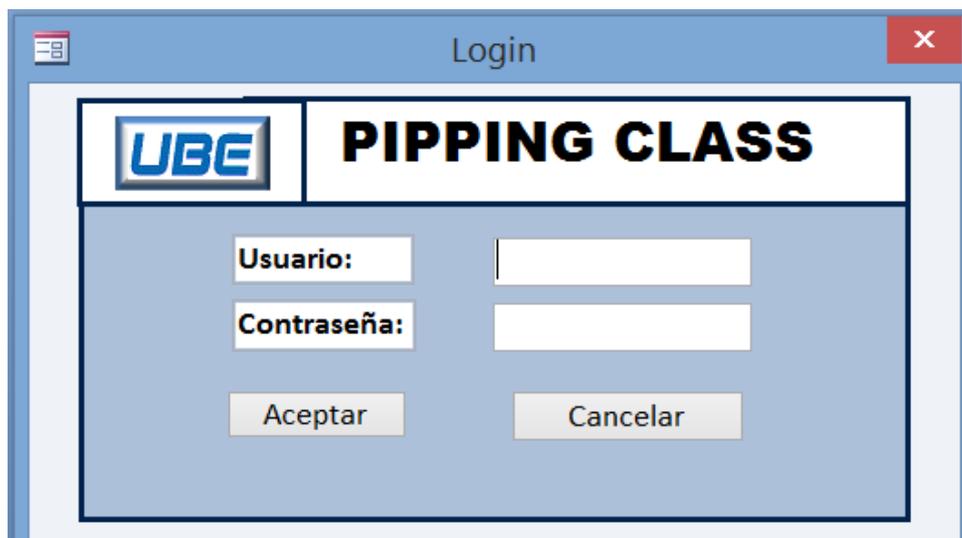


Figura 6.1: Pantalla de Login

En primer lugar, tenemos la función conectar que establece la conexión con la base de datos. Esta función se ejecuta cuando se carga el formulario de login. Cabe destacar que la conexión con la base de datos se realiza de una forma segura debido a las características internas del SQL Server y debido a los motivos explicados en el apartado 6.2.

Una vez establecida, la conexión con la base de datos, debemos introducir el usuario y contraseña para acceder correctamente a la aplicación. La función aceptar (ver código 6.1), valida que el usuario y el password introducido existan en la base de datos. Si es así, se abrirá correctamente la aplicación. Si por el contrario los datos introducidos no son correctos, se notificará al usuario. Cabe destacar que si este usuario se equivoca más de 3 veces en introducir

los datos, se le informará de que no tiene acceso a la aplicación y que debe consultar con el administrador de la aplicación.

En el caso de que se hayan introducido correctamente los datos y antes de abrir la aplicación, se generará una variable global con los datos del usuario que ha hecho el login, para gestionar sus permisos en la aplicación. Esta funcionalidad se realiza en la función "GeneraUsuario" (ver Código 6.2).

```

1 Private Sub Aceptar_Click()
2 On Error GoTo Mierror
3 'Control de acceso del usuario
4 Dim strSQL As String
5
6 Dim RsInicio As ADODB.recordSet
7 strSQL = "SELECT idUsuario FROM Usuarios WHERE Usuario=" & Me.usuario.Value & "
      ' AND Contraseña=" & Me.contraseña.Value & "'
8
9 Set RsInicio = New ADODB.recordSet
10
11 RsInicio.Open strSQL, Conexion, adOpenKeyset, adLockOptimistic
12
13 If RsInicio.RecordCount = 0 Then
14 MsgBox intento
15 If intento < 2 Then
16     MsgBox 'El nombre de Usuario o la Contraseña son incorrectos '
17     Me.usuario.SetFocus
18     intento = intento + 1
19 Else
20     MsgBox 'No tiene acceso a la aplicación, consulte con el Administrador '
21     Set RsInicio = Nothing
22     Set Conexion = Nothing
23     Application.Quit
24 End If
25 Else
26     'Acierto. Hay que generar el recordset del usuario de inicio, con todos sus
      datos.
27     GeneraUsuario RsInicio("idusuario")
28     DoCmd.OpenForm "Navegacion", acNormal, , , , acWindowNormal
29     DoCmd.Close acForm, "Login"
30
31 End If
32 Set RsInicio = Nothing
33 Exit Sub
34 Mierror:
35 MsgBox Err.Description
36 End Sub
37
38
39 Private Sub cancelar_Click()
40 DoCmd.Close acForm, "Login"
41 Application.Quit
42 End Sub
43
44 Private Sub Form_Load()
45 intento = 0
46 'establecemos la conexion
47
48 Conectar
49 End Sub

```

```

50
51 Private Sub Form_Open(Cancel As Integer)
52 contraseña.InputMask = "password"
53 End Sub

```

Código 6.1: Control de Login Correcto

```

1 Public Conexion As ADODB.Connection
2 Public RsUsuario As ADODB.recordSet
3 Public contador As Integer
4
5 Public Sub Conectar()
6 'conexion OLEDB para los adodb de la aplicación
7 On Error GoTo Mierror
8 Set Conexion = New ADODB.Connection
9 Conexion.CursorLocation = adUseClient
10 Conexion.ConnectionString = "Provider=SQLOLEDB.1;Integrated_Security=SSPI;Persist
   _Security_Info=False;Initial_Catalog=_PippingClass;Data_Source=ctopro\
   sqlexpress"
11 'Conexion.ConnectionString = "driver={SQL Server};server=ctopro\sqlexpress;
   database=MOC;uid=F_FALOMIR;Trusted_Connection=Yes"
12 Conexion.Open
13
14 Exit Sub
15 Mierror:
16 MsgBox Err.Description
17 End Sub
18
19
20 Public Sub GeneraUsuario(idUsuario As Integer)
21 'generamos el recordset con los datos del usuario que ha hecho el login, para
   gestionar los permisos en la aplicación
22 On Error GoTo Mierror
23 Dim Cadena As String
24
25 Set RsUsuario = New ADODB.recordSet
26 Cadena = "SELECT*_*_From_Usuarios_._WHERE_idUsuario=" & idUsuario & "__"
27
28 RsUsuario.Open Cadena, Conexion, adOpenStatic, adLockReadOnly
29
30
31 Exit Sub
32 Mierror:
33 MsgBox Err.Description
34 End Sub

```

Código 6.2: Establece la conexión con la base de datos y genera la variable global con los permisos del usuario

### 6.1.2. Módulo de Mantenimiento

En este apartado se va a explicar detalladamente el funcionamiento del módulo de mantenimiento. Para acceder a este módulo, el usuario debe tener asignado permisos en la base de datos y podrá acceder al módulo a través de la pantalla principal pulsando el botón de Mantenimiento.

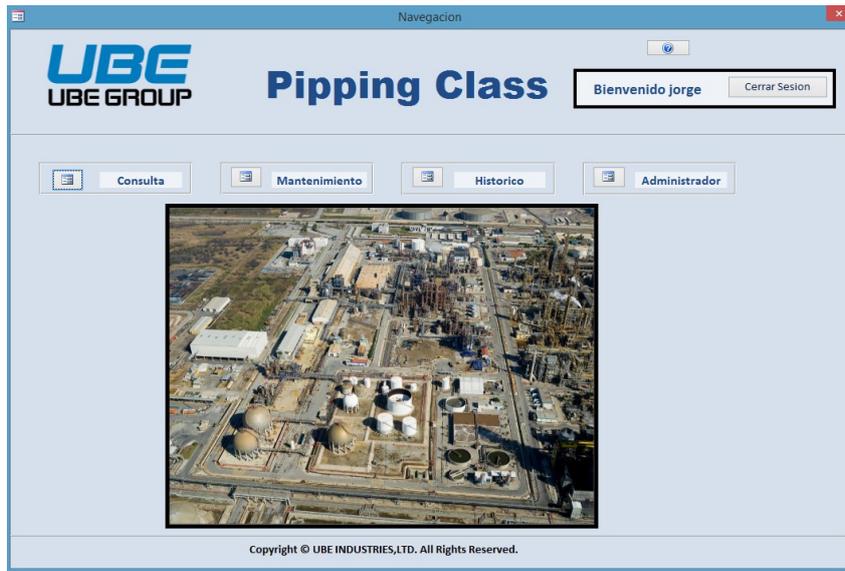


Figura 6.2: Pantalla Principal de la Aplicación

Si el usuario tiene permisos asignados accederá a la siguiente pantalla de mantenimiento.

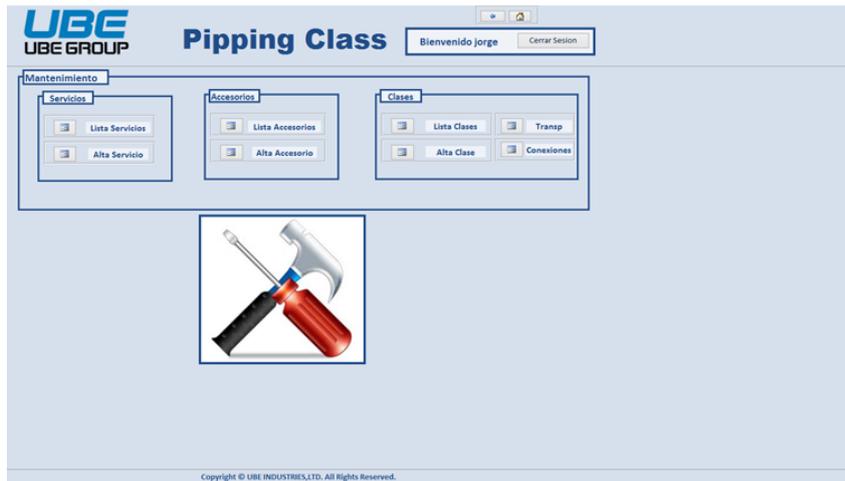


Figura 6.3: Pantalla Principal de Mantenimiento

La programación para lograr la interconexión entre la página principal y la página de mantenimiento la podemos ver a continuación:

```

1 Private Sub mantenimiento_Click()
2 If RsUsuario("rol") = "Consulta" Then
3 MsgBox "No tiene permisos para el mantenimiento de la aplicacion. Consulte con el administrador"
4
5 Else
6 DoCmd.OpenForm "Mantenimiento", acNormal, , , , acWindowNormal
7 DoCmd.Close acForm, "Navegacion"
8
9 End If
10 End Sub

```

Código 6.3: Acceso a la pantalla de mantenimiento

Como el mantenimiento de la aplicación se realiza sobre varios elementos vamos a seleccionar dos de ellos para mostrar su funcionamiento, ya que el resto están implementados de la misma forma. En este caso veremos como ejemplo la gestión de los accesorios y de las clases.

### Mantenimiento de Accesorio

En este apartado vamos a ver cómo se ha implementado tanto la inserción de los nuevos accesorios como su edición y borrado. A continuación se muestra parte del algoritmo utilizado para insertar un nuevo accesorio siempre que se cumplan las condiciones de filtrado del accesorio. En el manual de Usuario se pueden ver estas condiciones.

```

1 Private Sub registrar_Click()
2
3 DoCmd.SetWarnings False
4 Dim sentencia As String
5 Dim Historico As String
6 Dim idValvula As Boolean
7 idValvula = True
8 Dim idJunta As Boolean
9 idJunta = True
10 Dim micomando As ADODB.Command
11 Set micomando = New ADODB.Command
12 Dim MiParam As ADODB.Parameter
13
14
15 ' Algoritmo Para Insertar el nuevo accesorio
16
17 If Not IsNull(Me.nombreElemento) And Not IsNull(Me.tipo) And Not IsNull(Me.diametroMin) And Not IsNull(Me.diametroMax) Then 'And Me.tipo.Value <> "VALVULA" Then
18     If (Me.tipo.Value = "VALVULA") Then
19         If (IsNull(Me.idValvula.Value)) Then
20             MsgBox "Seleccione una especificacion de valvula"
21             idValvula = False
22         End If
23
24     ElseIf (Me.tipo.Value = "JUNTA") Then
25         If (IsNull(Me.idJunta.Value)) Then
26             MsgBox "Seleccione una especificacion de junta"
27             idJunta = False
28         End If

```

```

29 End If
30
31 If (idValvula) And (idJunta) Then
32
33
34     sentencia = "INSERT INTO _db_ Elementos (nombreElemento, _tipo, _diametroMin
        , _diametroMax, material, norma, schedule, rating, extremos, acabado,
        observaciones, _codigo, idJunta, idValvula) VALUES (' & Me.nombreElemento
        .Value & " ', ' & Me.tipo.Value & " ', ' & Me.diametroMin.Value & " ', '
        , ' & Me.diametroMax.Value & " ', ' & Me.material.Value & " ', ' &
        Me.norma.Value & " ', ' & Me.schedule.Value & " ', ' & Me.rating.Value
        & " ', ' & Me.extremos.Value & " ', ' & Me.acabado.Value & " ', ' & Me
        .observaciones.Value & " ', ' & Me.codigo.Value & " ', ' & Me.idJunta.
        Value & " ', ' & Me.idValvula.Value & " ') _"
35
36
37     ' Algoritmo Para Guardar en el Historico esta Nueva Insercion
38     micomando.ActiveConnection = Conexion
39     micomando.CommandText = "spAddHistorico"
40     micomando.CommandType = adCmdStoredProc
41     Set MiParam = micomando.CreateParameter("idUsuario", adInteger,
        adParamInput, , RsUsuario("idUsuario"))
42     micomando.Parameters.Append MiParam
43     Set MiParam = micomando.CreateParameter("nombreUsuario", adVarChar,
        adParamInput, 50, RsUsuario("usuario"))
44     micomando.Parameters.Append MiParam
45     Set MiParam = micomando.CreateParameter("tabla", adVarChar, adParamInput,
        50, "Elementos")
46     micomando.Parameters.Append MiParam
47     Set MiParam = micomando.CreateParameter("fecha", adDate, adParamInput,
        50, Now)
48     micomando.Parameters.Append MiParam
49     Set MiParam = micomando.CreateParameter("accion", adVarChar, adParamInput
        , 50, "Insercion")
50     micomando.Parameters.Append MiParam
51     Set MiParam = micomando.CreateParameter("productofinal", adVarChar,
        adParamInput, 50, Me.nombreElemento.Value)
52     micomando.Parameters.Append MiParam
53     Set MiParam = micomando.CreateParameter("descripcion", adVarChar,
        adParamInput, 100, "Se_añadido_ _" & Me.tipo.Value & "_con_diametro_
        minimo_" & Me.diametroMin.Value & "_diametro_maximo_" & Me.diametroMax
        .Value & "_y_de_material_" & Me.material.Value & "_")
54     micomando.Parameters.Append MiParam
55
56
57     micomando.Execute
58
59     DoCmd.RunSQL sentencia
60     MsgBox ("El_registro_ha_sido_guardado")
61
62         ' Se resetan los valores introducidos por si desea
        ' insertar mas de un accesorio a la vez
63     Call resetear
64
65         ' Se redirige a la pantalla adecuada segun desde donde se
        ' ha accedido
66     If (Me.OpenArgs = 0) Then
67         DoCmd.OpenForm "AltaClases", acNormal, , , acWindowNormal
68         DoCmd.Close acForm, "AltaAccesorio"
69         Form_ListaElementosM.Requery

```

```

70      ElseIf (Me.OpenArgs = 1) Then
71
72          DoCmd.OpenForm "Mantenimiento", acNormal, , , , acWindowNormal
73          DoCmd.Close acForm, "AltaAccesorio"
74          Form_ListaElementos.Requery
75      ElseIf (Me.OpenArgs = 2) Then
76          DoCmd.OpenForm "ListaElementos", acNormal, , , , acWindowNormal
77          DoCmd.Close acForm, "AltaAccesorio"
78          Form_ListaElementos.Requery
79      End If
80
81
82      End If
83
84      ' Se resetan los estilos de los campos por si desea insertar mas de un accesorio
      ' a la vez
85
86      Else
87          If (IsNull(Me.nombreElemento.Value)) Then
88              Me.nombreElemento.BackColor = vbRed
89              Me.nombreElemento.SetFocus
90              errornombreelemento.Visible = True
91          Else
92              Me.nombreElemento.BackColor = vbWhite
93              errornombreelemento.Visible = False
94
95          End If
96
97          .
98          .
99          .
100
101
102
103      End If
104      End Sub
105
106
107      Private Sub resetear()
108          nombreElemento.Value = Null
109          tipo.Value = Null
110
111          .
112          .
113          .
114
115          nombreElemento.SetFocus
116
117          nombreElemento.BackColor = vbWhite
118          tipo.BackColor = vbWhite
119
120          .
121          .
122          .
123
124
125
126
127          errornombreelemento.Visible = False
128          errortipo.Visible = False

```

```

129
130
131
132
133
134 End Sub

```

Código 6.4: Inserción de nuevo accesorio

Simplemente el algoritmo se encarga de realizar una nueva inserción en la base de datos, tanto del nuevo accesorio como del registro de la nueva inserción en el histórico. Posteriormente se resetean los campos del formulario por si se quiere añadir más accesorios y en caso contrario se redirige correctamente a la pantalla desde donde se ha accedido al formulario.

Respecto a la edición del accesorio, podemos destacar el siguiente algoritmo

```

1 Private Sub modificar_Click()
2 On Error GoTo Mierror
3 MsgBox " _Elemento_Modificado"
4
5 Dim prodIni As String
6 Dim prodFin As String
7 Dim descrip As String
8 prodIni = ""
9 prodFin = ""
10 Dim micomando As ADODB.Command
11 Set micomando = New ADODB.Command
12 Dim MiParam As ADODB.Parameter
13
14 If (Me.nombreElemento.OldValue <> Me.nombreElemento.Value) Then
15     prodIni = prodIni + " _Nombre_Elemento:_ " + Me.nombreElemento.OldValue
16     prodFin = prodFin + " _Nombre_Elemento:_ " + Me.nombreElemento.Value
17 End If
18
19 If (Me.tipo.OldValue <> Me.tipo.Value) Then
20     prodIni = prodIni + " _Tipo:_ " + Me.tipo.OldValue
21     prodFin = prodFin + " _Tipo:_ " + Me.tipo.Value
22 End If
23
24 .
25 .
26 .
27
28 ' Insercion del registro modificado en el historico
29 Historico = "INSERT INTO dbo_Historico (idUsuario , nombreUsuario , tabla , fecha ,
    accion , productoinicial , productofinal , _descripcion)_VALUES( _" & RsUsuario("
    idUsuario") & ", '" & RsUsuario(" usuario") & "' , ' _Elementos ', Now, ' Modificar ', '"
    & prodIni & "' , ' _" & prodFin & "' , ' _Se_ha_modificado_un_ " & Me.tipo.Value &
    " _con_diametro_minimo_ " & Me.diametroMin.Value & " _diametro_maximo_ " & Me.
    diametroMax.Value & " _y_de_material_ " & Me.material.Value & " _'"
30 DoCmd.RunSQL Historico
31
32 ' Redireccion a la pantalla Lista Elementos
33 DoCmd.OpenForm "ListaElementos", acNormal, , , , acWindowNormal
34 DoCmd.Close acForm, "ModificaElemento"
35 Form_ListaElementos.Requery
36
37 Exit Sub

```



Figura 6.5: Formulario de modificación del accesorio seleccionado

Esto es posible mediante la siguiente instrucción:

```
1 DoCmd.OpenForm "ModificaElemento", acNormal, , "idElemento=" & Me.idElemento .
  Value, , acWindowNormal
```

Código 6.6: Modificación de un accesorio

En definitiva, realmente MS Access internamente carga el formulario de modificación con el registro indicado.

Respecto al borrado del accesorio, podemos destacar el siguiente algoritmo.

```
1 Private Sub borrar_Click ()
2 On Error GoTo Mierror
3 Dim mensaje As String
4 Dim micomando As ADODB.Command
5 Dim respuesta As String
6 Dim Historico As String
7
8 ' Se avisa al usuario se realmente quiere borrar el accesorio
9 mensaje = "¿Seguro que desea eliminar el accesorio?" & Me.nombreElemento.Value
  & " ¿?"
10 respuesta = MsgBox(mensaje, vbOKCancel)
11
12 'Insertamos el borrado del accesorio en el historico
13 Set micomando = New ADODB.Command
14 micomando.ActiveConnection = Conexion
15 Historico = "INSERT INTO dbo_Historico (idUsuario, nombreUsuario, tabla, fecha,
  accion, producto inicial, descripcion) VALUES (" & RsUsuario("idUsuario") & ", " &
  & RsUsuario("usuario") & ", 'Elementos', Now, 'Borrado', " & Me.nombreElemento
```

```

16     .Value & " ', ' _Se_ha_borrado_un_" & Me.tipo.Value & " _con_diametro_minimo_" &
17     Me.diametroMin.Value & " _diametro_maximo_" & Me.diametroMax.Value & " _y_de_"
18     material_" & Me.material.Value & " ')"
19
20 ' Se borra el accesorio si el usuario confirma su borrado
21 If respuesta = vbOK Then
22     micomando.CommandText = "DELETE_From_Elementos_Where_idElemento=" & Me.
23     idElemento.Value & " '"
24     micomando.Execute
25     MsgBox " _Elemento_Borrado_"
26     Me.Requery
27
28     DoCmd.RunSQL Historico
29
30 End If
31
32 Set micomando = Nothing
33
34 Exit Sub
35
36 Mierror:
37 MsgBox "No_puede_borrar_este_accesorio_por_que_esta_siendo_utilizado_por_otras_"
38     clases"
39
40 End Sub

```

Código 6.7: Borrar un accesorio

Este algoritmo se encarga de borrar al accesorio seleccionado. En primer lugar se pide confirmación de borrado por si el usuario se ha equivocado. En el caso de que lo confirme, se procede a borrar el accesorio, insertar el registro borrado en el histórico y redirigir a la pantalla correspondiente.

## Mantenimiento de Clases

En primer lugar veremos un algoritmo implementado para gestionar correctamente el formulario de inserción de clases, de manera que no se puedan insertar nuevas clases con nombres ya existentes.

Se realiza un función que permite comprobar si la clase ya se encuentra en la base de datos de modo que si se intenta añadir una nueva con el mismo nombre de clase, se informe correctamente de que la introducida, no se puede añadir. Esto lo podemos ver en la función 'CompruebaClase' (ver código 6.9).

```

1 Function compruebaClase(miClase As String) As Boolean
2     Dim recordSet As ADODB.recordSet
3     Set recordSet = New ADODB.recordSet
4     Dim instruccion As String
5     Dim existe_clase As Boolean
6
7     instruccion = "SELECT_clase_from_Clases"
8
9     recordSet.Open instruccion , Conexion , adOpenDynamic , adLockOptimistic
10     'adOpenDynamic , adLockOptimistic
11     'adOpenStatic , adLockReadOnly

```

```

12
13   Do While recordSet.EOF = False
14       If recordSet!clase = miclase Then
15           existe_clase = True
16           Exit Do
17       End If
18       recordSet.MoveNext
19   Loop
20
21   recordSet.Close
22   Set recordSet = Nothing
23
24   compruebaClase = existe_clase
25
26 End Function

```

Código 6.8: Comprobar existencia de una clase

Ahora vamos a pasar a explicar la funcionalidad de la gestión de clases. Tanto la inserción como la modificación y el borrado se programan de igual manera que la descrita anteriormente en los accesorios. No obstante, se han programado algunos algoritmos en la inserción de nuevas clases que resulta interesante destacarlos por su complejidad y por ser algoritmos que el cliente quería expresamente que estuviesen (no el algoritmo en sí, si no la funcionalidad mostrada en la aplicación).

El algoritmo principal implementado, se ha utilizado tanto en servicios como accesorios y ratings, por lo que se va explicar el de servicios ya que el resto están programados del mismo modo. Este algoritmo sirve principalmente para asignar o eliminar servicios a la clase y que el usuario pueda visualizar los cambios al instante. A continuación se van a presentar dos códigos de programación: el primero es el que se encarga de insertar los servicios y el segundo es el que se encarga de borrarlos.

```

1  Dim sql As String
2  Dim sentencia As Command
3  Set sentencia = New Command
4  Dim recordSet As ADODB.recordSet
5  Set recordSet = New ADODB.recordSet
6  Dim micomando As ADODB.Command
7  Set micomando = New ADODB.Command
8  Dim Historico As String
9  Dim MiParam As ADODB.Parameter
10 Dim texto As String
11
12
13
14  'comprobación si la clase es vacía
15 If IsNull(Forms!AltaClases!clase.Value) Then
16     MsgBox "Introduce la clase anteriormente"
17     Forms!AltaClases!clase.SetFocus
18 ElseIf IsNull(Forms!AltaClases!material.Value) Then
19     MsgBox "Introduce el material anteriormente"
20     Forms!AltaClases!material.SetFocus
21 ElseIf IsNull(Forms!AltaClases!margenCorrosion.Value) Then
22     MsgBox "Introduce el margen de corrosion"
23     Forms!AltaClases!margenCorrosion.SetFocus
24
25 Else

```

```

26     If (compruebaServicioClase(Forms!AltaClases!idClase.Value, Me.idServicio.
27         Value)) Then
28         MsgBox " _Esta _Clase _ya _contiene _dicho _servicio. _Por _favor _seleccione _
29             otro"
30
31     ElseIf (Form_ListaInfoInicial2.RecordsetClone.RecordCount = 0) Then
32
33         Form_AltaClases.mc.Value = Replace(Form_AltaClases.margenCorrosion.Value,
34             ". ", ", ")
35
36         sentencia.ActiveConnection = Conexion
37         sentencia.CommandText = "INSERT INTO _ServicioClase (_idServicio, idClase,
38             margenCorrosion) _VALUES(" & Me.idServicio.Value & ", _" & Forms!
39             AltaClases!idClase.Value & ", _" & Forms!AltaClases!margenCorrosion.
40             Value & " _)"
41         sentencia.Execute
42
43         micomando.ActiveConnection = Conexion
44         micomando.CommandText = "INSERT INTO _Historico (idUsuario, _nombreUsuario,
45             tabla, fecha, accion, productofinal, _descripcion) _VALUES(?, ?, ?, ?, ?, ?, ?)"
46
47         texto = " _Se _añadido _el _servicio _" + Me.servicio.Value + " _a _la _clase _"
48             + Form_AltaClases.clase.Value
49
50     Set MiParam = micomando.CreateParameter("idUsuario", adInteger,
51         adParamInput, , RsUsuario("idusuario"))
52     micomando.Parameters.Append MiParam
53         .
54         .
55         .
56         micomando.Parameters.Append MiParam
57     micomando.Execute
58     Form_ListaInfoInicial2.Form.RecordSource = "Select *_ _from _
59         dbo_ListaInicial _where _clase=" & Forms!AltaClases!clase.Value & " _"
60     Form_ListaInfoInicial2.Requery
61
62     ElseIf (Form_ListaInfoInicial2.RecordsetClone.RecordCount <> 0) Then
63
64     If (Form_ListaInfoInicial2.margenCorrosion.Value = Replace(
65         Form_AltaClases.margenCorrosion.Value, ". ", ", ")) Then
66         sentencia.ActiveConnection = Conexion
67         sentencia.CommandText = "INSERT INTO _ServicioClase (_idServicio,
68             idClase, margenCorrosion) _VALUES(" & Me.idServicio.Value & ", _"
69             & Forms!AltaClases!idClase.Value & ", _" & Forms!AltaClases!
70             margenCorrosion.Value & " _)"
71         sentencia.Execute
72
73         micomando.ActiveConnection = Conexion
74         micomando.CommandText = "INSERT INTO _Historico (idUsuario, _
75             nombreUsuario, tabla, fecha, accion, productofinal, _descripcion) _
76             VALUES(?, ?, ?, ?, ?, ?, ?)"

```

```

70         texto = "Seañadido el servicio" + Me.servicio.Value + " de la
71             clase" + Form_AltaClases.clase.Value
72         Set MiParam = micomando.CreateParameter("idUsuario", adInteger,
73             adParamInput, , RsUsuario("idusuario"))
74         micomando.Parameters.Append MiParam
75         .
76         .
77         .
78         micomando.Parameters.Append MiParam
79
80
81
82         micomando.Execute
83         Form_ListaInfoInicial2.Form.RecordSource = "Select * from
84             dbo_ListaInicial where clase=" & Forms!AltaClases!clase.Value
85             & " "
86         Form_ListaInfoInicial2.Requery
87
88     Else
89         MsgBox "Introduce el mismo valor de margen de corrosion"
90     End If
91 End If
92
93     Me.Requery
94 End If
95
96 End Sub

```

Código 6.9: Inserción de servicios en una clase

```

1 Private Sub borrar_Click()
2     Dim sentencia As Command
3     Set sentencia = New Command
4     sentencia.ActiveConnection = Conexion
5     sentencia.CommandText = "DELETE FROM ServicioClase Where idServicioClase=" &
6         Me.idServicioClase.Value & " "
7     sentencia.Execute
8     Dim micomando As ADODB.Command
9     Set micomando = New ADODB.Command
10    Dim MiParam As ADODB.Parameter
11    Dim texto As String
12    texto = "Seañadido borrado el servicio" + Me.servicio.Value + " de la clase"
13        + Form_AltaClases.clase.Value
14        micomando.ActiveConnection = Conexion
15        micomando.CommandText = "INSERT INTO Historico (idUsuario, nombreUsuario,
16            tabla, fecha, accion, productofinal, descripcion) VALUES(?,?,?,?,?,?)"
17
18        Set MiParam = micomando.CreateParameter("idUsuario", adInteger,
19            adParamInput, , RsUsuario("idusuario"))
20        micomando.Parameters.Append MiParam
21        .
22        .
23        .
24        micomando.Parameters.Append MiParam

```

```

23         micomando.Execute
24
25 Form_ListaInfoInicial2.Form.RecordSource = "Select * from dbo_ListaInicial where
        clase=" & Forms!AltaClases!clase.Value & "''"
26 Form_ListaInfoInicial2.Requery
27
28 If (Form_ListaInfoInicial2.RecordsetClone.RecordCount = 0) Then
29     Form_AltaClases.mc.Value = ""
30
31 End If
32
33 End Sub

```

Código 6.10: Borrar servicios de una clase

Lo que se hace en estos algoritmos es insertar o eliminar el registro seleccionado de la tabla que tiene las relaciones entre clases y servicios de manera que cuando se hagan los cambios se haga un refresco de la base de datos y así el usuario pueda ver estos cambios al instante.

Aparte de estos algoritmos podemos destacar dos más. El primero es un algoritmo programado por que el usuario a la hora de realizar pruebas en la aplicación, solía introducir el margen de corrosión de los productos con comas en vez de puntos. Y como era un error frecuente, se ha programado el algoritmo descrito a continuación para evitar este problema de consistencia en la base de datos.

```

1 Private Sub margencorrosion_LostFocus()
2 Dim coma, i As Integer
3 If (Not IsNull(Me.margenCorrosion.Value)) Then
4 For i = 1 To Len(Me.margenCorrosion.Value) Step 1
5     If (Mid(Me.margenCorrosion.Value, i, 1) = ",") Then
6         MsgBox "Warning: Se debe introducir el valor del margen de corrosion con
            punto si se desea que el valor tenga decimales"
7         Me.margenCorrosion.Value = Null
8         Me.clase.SetFocus
9         Me.margenCorrosion.SetFocus
10        Exit For
11        End If
12
13 Next i
14
15 End If
16 End Sub

```

Código 6.11: Comprobar la introducción correcta del margen de corrosión

El segundo algoritmo a destacar es el utilizado para filtrar las columnas de una tabla de accesorios, ya que, facilita al usuario la búsqueda de información. Por ejemplo, filtrar los accesorios de una clase por su tipo de material

```

1 Private Sub material_MouseDown(Button As Integer, Shift As Integer, X As Single,
    Y As Single)
2 If Button = acRightButton Then
3     CommandBars("Form_View_Control").ShowPopup
4 End If
5 End Sub

```

Código 6.12: Comprobar la introducción correcta del margen de corrosión

### 6.1.3. Módulo de Administración

Para acceder a este módulo, el usuario debe tener asignado permisos de administrador en la base de datos y podrá acceder al módulo a través de la pantalla principal pulsando el botón de Administración cómo se puede ver la figura 6.2.

Si el usuario tiene permisos asignados accederá a la siguiente pantalla de administración.

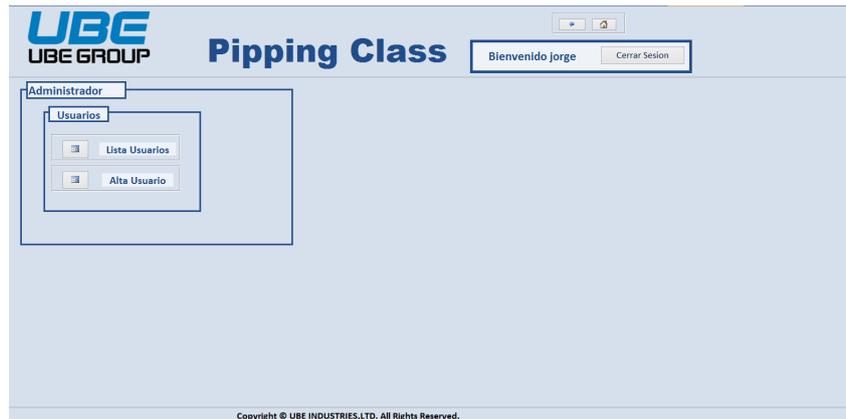


Figura 6.6: Pantalla de Administración

La programación para lograr la interconexión entre la página principal y la página de administración la podemos ver a continuación:

```
1 Private Sub administrador_Click()  
2 If RsUsuario("rol") = "Consulta" Or RsUsuario("rol") = "Mantenimiento" Then  
3     MsgBox "No tiene permisos de administrador. Por favor consulte con el  
4     administrador"  
5 Else  
6     DoCmd.OpenForm "Administrador", acNormal, , , , acWindowNormal  
7     DoCmd.Close acForm, "Navegacion"  
8 End If  
9 End Sub
```

Código 6.13: Acceso a la pantalla de administración

Este módulo está implementado de la misma forma que el módulo de mantenimiento, ya que la principal funcionalidad de este módulo es gestionar los usuarios que acceden a la aplicación y los roles que tienen cada uno de ellos. En este caso no se puede ver el funcionamiento del módulo de administración en el manual de usuario, ya que, este módulo solo podrá ser usado por los administradores de la aplicación. Cabe destacar que aunque no aparezca en el manual, a estos usuarios se les realizó una formación especial para que comprendieran la correcta utilización de dicho módulo.

En este apartado se realiza una función que permite comprobar si el usuario ya se encuentra en la base de datos de modo que si se intenta añadir uno nuevo con el mismo nombre de usuario, se informe correctamente de que el introducido, no se puede añadir. Esto lo podemos ver en la función 'CompruebaUsuario' (ver código 6.14).

The image shows a web browser window with the title 'AltaUsuario'. The main heading is 'ALTA USUARIO'. The form contains the following fields:

- NIF: [input field]
- Nombre: [input field]
- Apellidos: [input field]
- Usuario: [input field]
- Contraseña: [input field]
- Confirmación Contraseña: [input field]
- Rol: [dropdown menu]
- Correo: [input field]

At the bottom of the form are two buttons: 'Registrar' and 'Cancelar'.

Figura 6.7: Formulario de alta de un usuario

```

1
2 Function compruebaUsuario(miusuario As String) As Boolean
3   Dim recordSet As ADODB.recordSet
4   Set recordSet = New ADODB.recordSet
5   Dim instruccion As String
6   Dim existe_usuario As Boolean
7
8   instruccion = "SELECT_usuario_from_Usuarios"
9
10  recordSet.Open instruccion , Conexion , adOpenDynamic , adLockOptimistic
11  'adOpenDynamic , adLockOptimistic
12  'adOpenStatic , adLockReadOnly
13
14  Do While recordSet.EOF = False
15    If recordSet!usuario = miusuario Then
16      existe_usuario = True
17      'Exit Do
18    End If
19    recordSet.MoveNext
20  Loop
21
22  recordSet.Close
23  Set recordSet = Nothing
24
25  compruebaUsuario = existe_usuario
26
27
28 End Function

```

Código 6.14: Comprobar la existencia de un usuario

### 6.1.4. Módulo Historico

Para acceder a este módulo, el usuario debe tener asignado permisos en la base de datos y podrá acceder al modulo a través de la pantalla principal pulsando el botón de Histórico cómo se puede ver la figura 6.2.

A continuación se muestra la función que redirige correctamente a la pantalla histórico:

```

1 Private Sub historico_Click()
2 DoCmd.OpenForm "Historico", acNormal, , , , acWindowNormal
3 DoCmd.Close acForm, "Navegacion"
4 End Sub

```

Código 6.15: Acceso al formulario del histórico

Este módulo simplemente se encarga de mostrar en un tabla el histórico de todos los cambios realizados en la base de datos cómo se puede ver a continuación

nombreUsuario	tabla	accion	fecha	productoInicial	productoFinal	descripcion
jorge	Elementos	Insercion			cao	
jorge	Elementos	Insercion	17/12/2015 8:57:07		cosa	
jorge	Elementos	Insercion	17/12/2015 9:23:36		tipo	Se añadido un cap con diametro minimo 12" diametro maximo 22" y de material hierro
jorge	Elementos	Modificar	17/12/2015 9:48:26		tipo	Se añadido un cap con diametro minimo 12" diametro maximo 22" y de material hierro
jorge	Elementos	Modificar	17/12/2015 10:35:58	90	99	Se añadido un cap con diametro minimo 12" diametro maximo 22" y de material hierro
jorge	Elementos	Modificar	17/12/2015 10:40:29			Se añadido un brida con diametro minimo 2" diametro maximo 33" y de material qweq
jorge	Elementos	Modificar	17/12/2015 10:45:28	Diametro Minimo : 12" Material :	Diametro Minimo : 12" Materu	Se ha modificado un cap con diametro minimo 12" diametro maximo 22" y de material ?
jorge	Elementos	Borrado	17/12/2015 10:53:13	cao		Se ha borrado un codo con diametro minimo 12 diametro maximo 33 y de material dqw
jorge	Servicio	Insercion	17/12/2015 12:20:49		Nitrato Sodico	Se añadido el servicio Nitrato Sodico con el simbolo No2
jorge	Servicio	Modificar	17/12/2015 12:21:02	Simbolo : No2	Simbolo : N	Se ha modificado el servicio Nitrato Sodico con simbolo No2 por el servicio Nitrato
jorge	Servicio	Borrado	17/12/2015 12:21:30	Agua Caliente		Se ha borrado el servicio Agua Caliente con simbolo HW
jorge	Servicio	Borrado	17/12/2015 12:26:23	ewr		Se ha borrado el servicio ewr con simbolo egeq
jorge	Servicio	Modificar	17/12/2015 13:24:02	Simbolo : fdf	Simbolo : yt	Se ha modificado el servicio few con simbolo fdf por el servicio few con simbolo yt
jorge	Servicio	Insercion	17/12/2015 13:24:40		tres	Se añadido el servicio tres con el simbolo uy
jorge	Servicio	Borrado	17/12/2015 13:24:50	tres		Se ha borrado el servicio tres con simbolo uy
jorge	Servicio	Insercion	18/12/2015 8:39:03		Acido Nitrico	Se añadido el servicio Acido Nitrico con el simbolo N
jorge	Servicio	Modificar	18/12/2015 8:39:20	Simbolo : N	Simbolo : No2	Se ha modificado el servicio Acido Nitrico con simbolo N por el servicio Acido Nitr
jorge	Servicio	Borrado	18/12/2015 8:39:30	Acido Nitrico		Se ha borrado el servicio Acido Nitrico con simbolo N
jorge	Servicio	Insercion	18/12/2015 9:21:36		Nitrico	Se añadido el servicio Nitrico con el simbolo No2
jorge	Elementos	Insercion	21/12/2015 10:36:14		tyu	Se añadido un brida con diametro minimo 12" diametro maximo 22" y de material A5
jorge	Servicio	Insercion	21/12/2015 11:11:43		a2	Se añadido el servicio a2 con el simbolo a2
jorge	Servicio	Insercion	21/12/2015 11:11:55		a3	Se añadido el servicio a3 con el simbolo a3
jorge	Servicio	Insercion	21/12/2015 11:12:52		a4	Se añadido el servicio a4 con el simbolo a4
jorge	Servicio	Insercion	21/12/2015 11:13:03		a5	Se añadido el servicio a5 con el simbolo a5
jorge	Servicio	Borrado	21/12/2015 11:13:24	a5		Se ha borrado el servicio a5 con simbolo a5

Figura 6.8: Ejemplo de histórico de cambios

### 6.1.5. Módulo de Consulta

En esta pantalla(ver figura 6.10) se utiliza realmente el filtro de consulta de los datos. A continuación vamos a explicar el funcionamiento de este código. Cómo podemos ver, todos los campos del filtrado son necesarios, por lo que si el usuario se deja alguno sin introducir se le informará para que lo haga. En primer lugar el usuario debe introducir el servicio, y una vez hecho esto tendrá que elegir sobre que presiones y temperaturas trabaja dicho fluido. En el caso de que estas presiones y temperaturas no sean correctas por que no entran dentro de los rangos de valores permitidos, se le informará de nuevo para que las introduzca de nuevo.

Figura 6.9: Formulario de filtrado de clases

Cabe destacar que están presiones y temperaturas están almacenadas previamente bajo un estándar de seguridad por lo que solo se filtraran clases cuyos ratings sean válidos.

```

1 Private Sub Filtrar_Click ()
2 On Error GoTo Mierror
3 Dim db As DAO.Database
4 Dim rs As DAO.recordSet
5 Dim mensajeError As String
6 Dim rating As String
7 Dim iClase As String
8 Dim sentencia As String
9
10 If (IsNull(servicio.Value) Or IsNull(presionDiseño.Value) Or IsNull(
    temperaturaDiseño.Value)) Then
11     GoTo Mierror
12 End If
13
14 sentencia = "Select * _From_ dbo_InfoInicial _where_ idServicio=" & Me.servicio.
    Column(0) & " _AND_ minTemp<" & Me!temperaturaDiseño & " _AND_ maxTemp_>" & Me!
    temperaturaDiseño & " _AND_ minPress<" & Me!presionDiseño & " _AND_ maxPress_>"
    & Me!presionDiseño & ""
15 Set db = CurrentDb
16 Set rs = db.OpenRecordset(sentencia)
17
18
19
20 If rs.RecordCount <= 0 Then
21     'Mensaje de error o informativa al usuario
22     MsgBox ("Introduce _datos_ de _presion_ y _temperatura_ validos")
23 Else
24
25     Me.ListaInfoInicial.Form.RecordSource = sentencia
26     Me.prueba.Value = Me.temperaturaDiseño.Value
27     Me.prueba2.Value = Me.presionDiseño.Value
28
29
30     Me.ListaInfoInicial.Form.Requery
31     Me.Accesorio.Value = Null
32     Form!InformacionAccesorios.Form.RecordSource = "Select * _from_
    dbo_InformacionAccesorios _where_ tipo='nada'"
33
34
35 End If

```

```

36
37 Exit Sub
38 Mierror :
39 mensajeError = " _Rellena _los _siguientes _campos: _"
40 If (IsNull(servicio.Value)) Then
41     mensajeError = mensajeError + " Servicio _"
42 End If
43 If (IsNull(presionDiseño.Value)) Then
44     mensajeError = mensajeError + " Presion _"
45 End If
46 If (IsNull(temperaturaDiseño.Value)) Then
47     mensajeError = mensajeError + " Temperatura _"
48 End If
49
50 MsgBox (mensajeError)
51
52
53 End Sub

```

Código 6.16: Condiciones de filtrado

### 6.1.6. Funcionalidades Extra

En la parte derecha de la pantalla principal de la aplicación (ver Figura A.3), tenemos un menú vertical con diferentes funcionalidades.

- Índice Especificaciones. Nos abre un informe con la información principal asociada a cada clase. Esta información es la siguiente:
  - Nombre de la clase
  - Servicios Asociados a la Clase
  - Material
  - Condiciones Iniciales tanto de temperatura como de presión
  - Rating
- Índice Válvulas. Nos abre un informe con la información principal asociada a cada válvula. Esta información es toda la especificación de dicha válvula.
- Observaciones. Nos abre un informe que nos permite consultar la tabla de las distintas observaciones de los accesorios.
- Tabla Resumen Fluidos vs Clases. Nos abre un informe que nos permite consultar las clases a las que pertenecen los distintos fluidos.
- Transposición de Antiguas Especificaciones. Nos abre un informe que nos permite consultar la transposición de las antiguas especificaciones a las nuevas, por si se quiere consultar una clase en concreto.
- Informe Clases. Nos abre un informe que nos permite consultar toda la información referente a una clase en concreto. Esta información es:
  - Los servicios asociados

- Las condiciones iniciales
  - La información de todos los accesorios
  - La tabla de conexiones
- Informe Válvulas y Fabricantes Homologados. Nos abre un informe que nos permite consultar toda la información referente a una válvula en concreto. Esta información es:
    - La especificación completa de la válvula
    - Los fabricantes homologados asociados
  - Tabla de Juntas. Nos abre un informe que nos permite consultar la información referente a todas las juntas (clase, tipo y nombre) y sus juntas homologadas.

Como la forma de programar el acceso a los informes es común a todas las funcionalidades comentadas anteriormente, vamos a explicar la de Informe de Clases

```

1 Private Sub informeclases_Click()
2 On Error GoTo Mierror:
3
4 Dim mensaje, titulo, defaultValue As String
5 Dim myValue As String
6 ' Set prompt.
7 mensaje = "Introduce_la_clase"
8 ' Set title.
9 titulo = "Informe_Clase"
10 defaultValue = "1" ' Set default value.
11
12 ' Display message, title, and default value.
13 myValue = InputBox(mensaje, titulo)
14 ' If user has clicked Cancel, set myValue to defaultValue
15 If IsNull(myValue) Then
16     myValue = defaultValue
17     MsgBox "Introduce_una_clase_Valida"
18 Else
19     Form_PippingClassv8.claseocultainforme.Value = myValue
20     DoCmd.OpenReport "InformeClases2", acViewPreview, , "clase=" & Forms!
        PippingClassv8!claseocultainforme.Value & "'", acWindowNormal, 1
21 End If
22
23 Exit Sub
24
25 Mierror:
26     MsgBox "Introduce_una_clase_Valida"
27 End Sub

```

Código 6.17: Acceso al informe de clases

Esta función simplemente solicita la introducción del nombre de un clase válida. Si es así, abrirá el informe correctamente.

Condiciones Iniciales

Rating	minT	maxT	minP	maxP	material
150#	-29	100	0	17,7	Acero Carbono

Tuberías y Nipples

Nombre	Tipo	Desde	Hasta	Sch	material	norma	extremos	observaciones
TUBERIA & NIPPLE	tuberia sin soldadura	1/2 "	1 1/2"	40s	A-106 Gr.B	ASME B 36.10	Planos	
TUBERIA & NIPPLE	tuberia sin soldadura	1/2 "	1 1/2"	40s	A-106 Gr.B	ASME B 36.10	Planos	
TUBERIA & NIPPLE	tuberia sin soldadura	1/2 "	1 1/2"	40s	A-106 Gr.B	ASME B 36.10	Planos	
TUBERIA & NIPPLE	Nipples	1/2"	1 1/2"	80s	A-106 Gr.B	ASME B 36.10	TBE	5
TUBERIA & NIPPLE	Nipples	1/2"	1 1/2"	80s	A-106 Gr.B	ASME B 36.10	TBE	5
TUBERIA & NIPPLE	Nipples	1/2"	1 1/2"	80s	A-106 Gr.B	ASME B 36.10	TBE	5

Codos

Nombre	Tipo	Desde	Hasta	Sch	material	norma	extremos	observaciones
CODO	Codo 90º	8"	24"	20s	A-234 Gr. WPB	ASME B 16.9	Biselados	
CODO	Codo 90º	8"	24"	20s	A-234 Gr. WPB	ASME B 16.9	Biselados	
CODO	Codo 90º	8"	24"	20s	A-234 Gr. WPB	ASME B 16.9	Biselados	

Figura 6.10: Ejemplo de informe de la clase A10

## 6.2. Seguridad de la Aplicación

Una aplicación de gestión empresarial, al tratar con datos relativos a ella, debe procurar ser lo más segura y robusta posible. Por ello, a la hora de configurar y desplegar la aplicación en el entorno real, se aseguraron algunos puntos clave del framework en temas de seguridad. Estas medidas y configuraciones permiten tener una protección básica de la aplicación. Para proteger la aplicación en un mayor nivel, lo ideal sería realizar una auditoría de seguridad en la que se encontrasen las vulnerabilidades a las que se encuentra expuesta y cubrirlas. Las medidas de seguridad que se tomaron para proteger la aplicación pretendieron cubrir algunas de las vulnerabilidades de aplicaciones más extendidas y utilizadas:

- **Inyección SQL.** Esta vulnerabilidad es la más típica y extendida en las aplicaciones. Consiste en introducir una sentencia SQL maliciosa en el transcurso de una consulta legítima de la aplicación al recibir variables del usuario. A pesar de ser muy grave, es fácilmente evitable si se filtran las variables a utilizar en una consulta SQL antes de utilizarse. Para este caso, lo ideal es usar **prepared statement**. Es una sentencia parametrizada que permite solamente introducir parámetros y no otras sentencias, ya que se filtran de antemano. Esto evita por completo esta vulnerabilidad, y ha sido utilizado a lo largo de la aplicación. En el código 6.9 se muestra un ejemplo de ello.

A pesar de tener cubiertas estas vulnerabilidades, existe una gran cantidad de aspectos que pueden fallar a la hora de proteger una aplicación de personas no gratas. Por ello, además de cubrir estas vulnerabilidades, se han tomado otras medidas de seguridad para proteger aún más la aplicación:

- Proteger contraseñas. Se programó la funcionalidad para que las contraseñas almacenadas se cifraran automáticamente aunque no dio tiempo a insertarla en la aplicación. Esto permite que, aunque se accediesen a los datos sensibles de la aplicación no pudiesen conocerse al estar cifrados.

### 6.3. Pruebas

A la hora de realizar un proyecto de dimensiones considerables, es de especial importancia utilizar un sistema de pruebas que facilite la depuración del sistema. A largo plazo, lo ideal es realizar pruebas unitarias en conjunto con el proyecto. Esto puede resultar tedioso al principio, ya que requiere una mayor cantidad de tiempo invertido para realizar una funcionalidad de forma inicial, pero esto se compensa a largo plazo evitando fallos futuros. En este proyecto no se ha hecho uso de pruebas unitarias debido a la escasez de tiempo pero se han utilizado otros tipos de pruebas explicadas a continuación.

Básicamente yo como usuario principal de mantenimiento realicé todo tipo de pruebas para comprobar que la aplicación funcionara en los distintos casos de uso. Estas pruebas consistieron en realizar múltiples inserciones de datos y verificar que la gestión completa de la aplicación funcionaba al detalle. Un ejemplo de ello, era crear una nueva clase con una serie de servicios, accesorios y ratings, y una vez creada, acceder al módulo de consulta para ver si realmente esa clase estaba integrada en la aplicación o no. Cabe destacar que estas pruebas se realizaron una vez finalizada la fase de implementación por si había que corregir algún código de programación en particular.

Además de ello, se dedicaron varias horas de formación a los usuarios de mantenimiento en la última semana del proyecto de manera que ellos también pudieran probar la aplicación e informar en el caso de encontrar algún tipo de error.

### 6.4. Documentación

Para este proyecto, a petición del cliente, era necesario realizar un manual de usuario que sirva como ayuda para la correcta utilización de la aplicación. Esta puede ser accedida por todos los tipos de usuarios en función de sus roles. El manual también describe de forma detallada los pasos a seguir para realizar cada una de las acciones generales de las que dispone la aplicación. Para el acceso a esta ayuda, se implementó un botón en la aplicación que abría un archivo Word con el contenido del manual de ayuda. El resultado final de esta ayuda puede encontrarse en el anexo A adjunto.

## 6.5. Implantación

Para realizar la implantación correcta de la aplicación, simplemente se consulto con los diferentes departamentos que usuarios iban a hacer uso de ella. Una vez concretado esto, se creo una carpeta de red llamada PippingClass de manera que para poder acceder a la aplicación Pipping Class se deberá entrar en la ruta `\\ctopro\aplicaciones\PippingClass\PippingClass` y ejecutar en el archivo `FluidoPilotoDefinitivaFinal.accdr`. Además de esto, se dieron permisos de red (sobre la carpeta indicada) a cada uno de los usuarios para poder acceder a la ruta indicada anteriormente.

## 6.6. Formación

En la última semana de desarrollo del proyecto, se realizaron varias reuniones con el departamento de Inspección y Fiabilidad para explicarles la forma correcta de usar la aplicación. Además se les entregó a cada uno un manual de usuario en formato escrito como complemento de ayuda. No obstante, tal y cómo se ha comentado una vez entren en la aplicación pueden acceder a este manual de usuario pulsando en el botón 



## Capítulo 7

# Conclusiones

En este apartado se van a presentar conclusiones en varios aspectos: conclusiones en el ámbito técnico y conclusiones en el ámbito personal.

### 7.0.1. Conclusiones Técnicas

El proyecto ha cumplido con las expectativas definidas inicialmente aunque algún apartado no se ha podido tratar por falta de tiempo como es el caso de las pruebas unitarias. Todos los objetivos y requisitos planteados por el departamento de Inspección y Fiabilidad se han implementado satisfactoriamente. De hecho los usuarios están muy contentos con la aplicación y a día de hoy, la aplicación está en funcionamiento.

Dicho de otra manera, el proyecto ha tenido una gran aceptación por parte de los usuarios, ya que en la actualidad está siendo utilizada para consultar diferentes clases con mayor facilidad e llevar informes impresos que resumen claramente las clases sobre las que van a trabajar según la zona de la planta química.

Este proyecto, al pertenecer a una gran aplicación de gestión y consultas eficientes e útiles de información, puede seguir ampliándose poco a poco hasta cubrir todas las necesidades que requiere los diferentes departamentos de la empresa. Esto es posible debido a que cada departamento de la empresa trabaja con una base de datos individual, es decir, separada de las demás, por lo que en un futuro se podría integrar esta aplicación con el resto de las aplicaciones usadas en los distintos departamentos ya que la información a consultar es común a todas. Un ejemplo de ello, sería añadir a los accesorios consultados el código SAP de modo que cuando se pulse al botón indicado se abriría la información del accesorio seleccionado en SAP (sistema de la información que utiliza la empresa). Así se integraría esta aplicación con la aplicación usada en el departamento de Compras. Además, los conocimientos adquiridos son extrapolables a prácticamente cualquier aplicación de MS Access que se requiera, lo cual me ayuda a tener mayor confianza a la hora de afrontar nuevos proyectos de ese ámbito.

En definitiva, como resultado de este proyecto, se ha obtenido una aplicación fácil de usar, amigable, robusta, segura y eficiente en cuestión del tratamiento de la distinta información

almacenada en la base de datos.

### 7.0.2. Conclusiones Personales

A lo largo de la estancia en prácticas y el consecuente desarrollo del proyecto, he adquirido una gran cantidad de conocimientos y experiencias. Para empezar, he aprendido mucho sobre el lenguaje Visual Basic. Este lenguaje es el lenguaje básico de programación de Microsoft Access y me ha permitido ampliar mis conocimientos generales, además de poder implementar funcionalidades extra que difícilmente hubiera podido realizar con la interfaz de dicho programa. Por otra parte, integrarse en un equipo me ha permitido observar las diferentes formas de trabajar de una gran empresa.

Al comenzar el proyecto, también observé que cada tecnología nueva que se pretende utilizar requiere un tiempo de formación y práctica para poder usarse de forma correcta. He notado mi evolución en cuanto a conocimientos y soltura a lo largo de la duración de la estancia, y esto me ha permitido comprobar que los estudiantes estamos más preparados para el mundo laboral de lo que a priori se podría pensar.

El tema del proyecto me ha gustado mucho, sobre todo sabiendo que lo que iba a implementar, iba a ser utilizado por usuarios que realmente lo necesitan en cuanto estuviese acabado. Esta experiencia sí que es algo que durante el tiempo de estudio en la universidad se echa en falta. Da mucha satisfacción saber que el proyecto realizado está siendo útil para un conjunto de usuarios. El hecho de haber tenido reuniones con los usuarios, y haber recibido retroalimentación directa también me ha animado mucho a mejorar la aplicación y cumplir los requisitos que esperan.

La posible integración del proyecto con otras partes desarrolladas por otros departamentos (la posible integración con el departamento de Compras mencionado anteriormente) me hizo aprender mucho acerca de cómo acomodar código escrito por diferentes personas entre sí y de cómo distribuir eficientemente cada una de las interfaces de la aplicación. Es sorprendente la cantidad de formas diferentes en las que se puede desarrollar una misma funcionalidad según el programador.

# Bibliografía

- [1] M.L. Liu, Distributed Computing: Principles and Applications, California Polytechnic State University - San Luis Obispo, 2004
- [2] Jerome Gabillaud y Thierry Groussard. Visual Basic 2012 y SQL Server 2012: Domine el desarrollo y el acceso a los datos
- [3] Project Management Institute, Inc. PMBOK Guide and Standards. <http://www.pmi.org/PMBOK-Guide-and-Standards.aspx>.
- [4] Gestión práctica de proyectos con Scrum: Desarrollo de software ágil para el Scrum Master, Antonio Martel
- [5] Seguridad Social.Trabajadores. [http://www.seg-social.es/Internet\\_1/Trabajadores/CotizacionRecaudaci10777/Regimenes/RegimenGeneraldeLaS10957/InformacionGeneral/index.htm](http://www.seg-social.es/Internet_1/Trabajadores/CotizacionRecaudaci10777/Regimenes/RegimenGeneraldeLaS10957/InformacionGeneral/index.htm)
- [6] James Rumbaugh, Ivar Jacobson, Grady Booch. The Unified modeling language reference manual /
- [7] UML and Patterns (Larman) <https://www.utdallas.edu/~chung/SP/applying-uml-and-patterns.pdf>
- [8] H. E. Eriksson and M. Penker. Business Modeling with UML: Business Patterns at Work. J.Wiley, 2000.
- [9] Campos, C., Grangel, R., Verde, V. Apunts d'Enginyeria del Programari de Gestió I (IG16) Col·lecció Sapientia. <http://www.uji.es/bin/publ/edicions/s50.pdf>
- [10] Merche Marqués. Bases de Datos. Colección Sapientia. <http://www.uji.es/bin/publ/edicions/bdatos.pdf>
- [11] Ellen Isaacs i Alan Walendowski. Designing from both sides of the screen. New Riders, 2002. <http://www.uidesigns.com>
- [12] Jenniffer Preece, Yvonne Rogers, i Helen Sharp. Interaction design: beyond human-computer interaction. John Wiley Sons, Inc., 2002. <http://www.id-book.com>
- [13] K. Pohl. Requirements Engineering: Fundamentals, Principles, and Techniques. Springer Publishing Company, Incorporated, 1st edition, 2010.
- [14] G. Booch, J. Rumbaugh, and I. Jacobson. The Unified Modeling Language User Guide, Second Edition. Addison Wesley, 2006.

- [15] R. Elmasri i S. B. Navathe. Fundamentos de sistemas de bases de datos, Addison-Wesley 2002.
- [16] R. Ramakrishnan i J. Gehrke. Sistemas de gestión de bases de datos, McGraw-Hill 2007.
- [17] Clayton Lewis, John Rieman. Task-centered User Interface Design: A practical Introduction. 1993, Sharebook, <http://www.hcibib.org/tcuid>
- [18] Lon Barfield. The User Interface: Concepts and Design. Addison-Wesley, 1994.

# Anexo A

## Manual de Usuario

### A.1. Índice

1. Diseño General
2. Acceso al Pipping Class
3. Validación de Usuario
4. Pantalla Principal
5. Formulario de Consulta
  - 4.1 Tabla de Conexiones
  - 4.2 Generar Informe Clase
  - 4.3 Información General
6. Histórico
7. Formulario de mantenimiento
  - 6.1 Mantenimiento de Servicios
  - 6.2 Mantenimiento de Accesorios
  - 6.3 Mantenimiento de Clases

### A.2. Diseño General

La aplicación PippingClass sirve para consultar de una manera rápida y eficiente la información que antiguamente estaba en el archivo Excel(PippingClass). También permite tanto la gestión del mantenimiento de los datos como la de los usuarios que entrarán en la aplicación.

En la aplicación se mantienen tanto los servicios como las clases, sus accesorios y sus condiciones iniciales, cada uno con sus características y parámetros. La aplicación permite a los diferentes

responsables/usuarios de mantenimiento introducir datos, que posteriormente los usuarios de consulta podrán observar.

El acceso de los usuarios a la aplicación y las tareas que pueden realizar están determinadas por un sistema de roles y de seguridad por medio del login del usuario. Cabe destacar que los datos se guardan en una base de datos SQL-SERVER de manera que la aplicación sea más segura, eficiente y robusta.

### A.3. Acceso al Pipping Class

Para poder acceder a la aplicación PippingClass se deberá entrar en la ruta \\ctopro\aplicaciones\PippingClass\PippingClass y ejecutar el archivo FluidoPilotoDefinitivaFinal.accdr

### A.4. Validación de Usuario

La entrada en la aplicación exige la introducción de un nombre de usuario y contraseña, que determinará el rol que tendremos en la aplicación:



Figura A.1: Login

Los roles existentes son:

- **Consulta:** Este tipo de usuarios solo tendrá acceso a la consulta de los datos de la aplicación y al histórico de cambios realizados tanto por las personas de mantenimiento como por el administrador.
- **Mantenimiento:** Este tipo de usuarios será el encargado de realizar el mantenimiento adecuado de todos los datos de la aplicación, ya sea, de los servicios disponibles, de las clases, de los accesorios asociados a dichas clases y de las condiciones iniciales (ratings). Además este usuario podrá acceder a la consulta y al histórico.
- **Administrador:** El administrador tiene acceso y permiso de edición a todos los datos, incluida la gestión de usuarios. Cabe destacar que si pulsamos el botón 'Cancelar' cerraremos directamente la aplicación.

## A.5. Pantalla Principal

La pantalla principal nos muestra las diferentes funcionalidades de la aplicación.

- La opción **Consulta** permite consultar los datos almacenados en nuestra base de datos a través de unos criterios de búsqueda.
- La opción **Mantenimiento** sirve para el mantenimiento de los datos de la aplicación y es accesible tanto por el usuario de mantenimiento como por el administrador.
- La opción **Histórico** contiene el histórico de todos los cambios realizados en la aplicación.
- La opción **Administrador** contiene la gestión de usuarios de la aplicación.

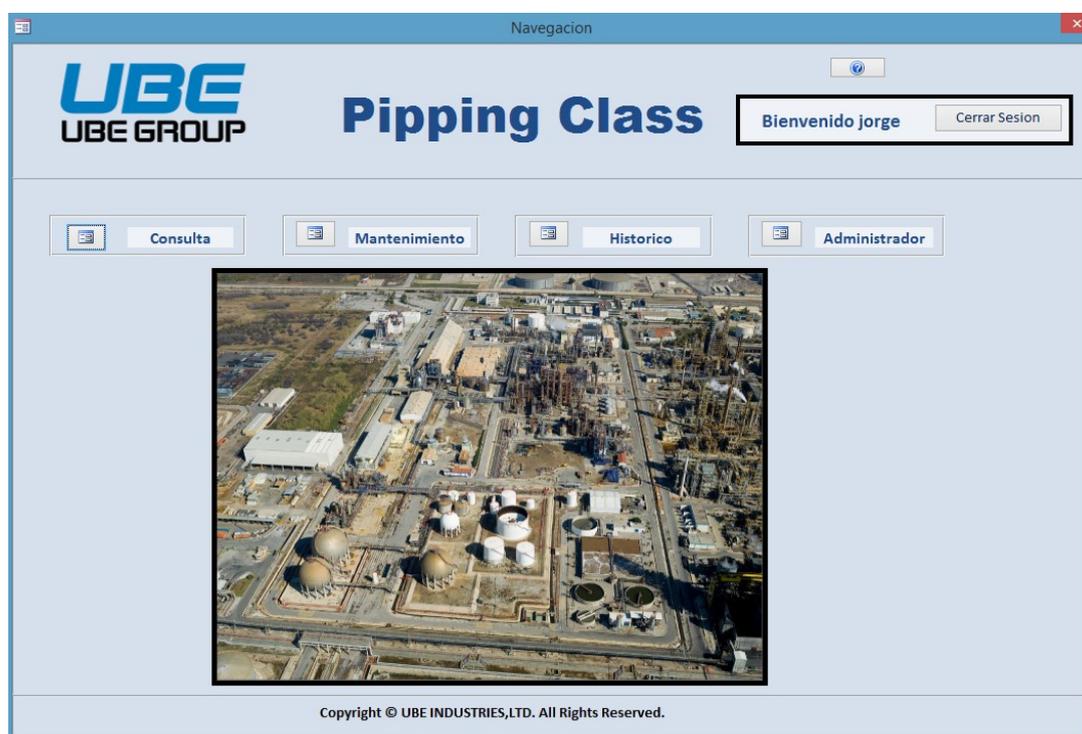


Figura A.2: Pantalla Principal

También tiene un botón de ayuda que establece el link con el presente manual y el botón Cerrar Sesión  con el cual cerramos la sesión de la aplicación.

## A.6. Formulario de Consulta

Desde el formulario de consulta se accede con detalle a todo el tipo de información guardada en la base de datos según los diferentes criterios de búsqueda. A continuación mostramos el aspecto inicial del formulario.

Figura A.3: Formulario de Consulta

Una vez visto el aspecto inicial, pasamos a comentar el funcionamiento inicial del formulario de consulta. En primer lugar se han de rellenar los diferentes criterios de búsqueda para obtener la información inicial de las clases que contienen dicha información. Estos criterios de búsqueda son los siguientes:

Figura A.4: Formulario de Filtrado de Clases

- Servicio: Seleccionamos el servicio que circulará por nuestro proyecto
- Símbolo: Se vincula automáticamente según el servicio seleccionado
- Presión de Diseño: Introducimos la presión de diseño de nuestro proyecto.
- Temperatura de Diseño: Introducimos la presión de diseño de nuestro proyecto

Cabe destacar que para que funcione adecuadamente la búsqueda se ha de rellenar los cuatro criterios de búsqueda mencionados anteriormente. Una vez introducidos estos criterios, pulsaremos el botón  para realizar la búsqueda.

Si en la búsqueda se ha introducido datos correctos, obtendremos una imagen parecida a la siguiente.



Figura A.5: Resultado de una Búsqueda de un Servicio

En ella podemos observar la información inicial resultante de nuestra búsqueda. A continuación seleccionaremos la clase sobre la cual queremos consultar información tanto de los accesorios como de sus conexiones según el material y el libraje obtenidos en la información inicial. Para ello pulsaremos el botón  colocado en la fila correspondiente que deseamos seleccionar. Para comprobar que se ha seleccionado dicha clase habrá que fijarse en los siguientes detalles:

- El título Información Accesorios ha sido modificado según la clase seleccionada. Por ejemplo: **INFORMACION ACCESORIOS CLASE A01**
- Se sombreadrá la línea cuya clase ha sido seleccionada.

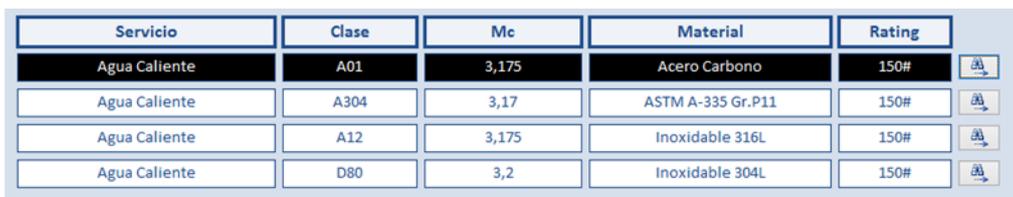


Figura A.6: Resultado Búsqueda

Una vez hecho esto, utilizaremos el filtro de accesorio



Figura A.7: Resultado Búsqueda

para consultar los diferentes accesorios asociados a esta clase. Como resultado tendremos una imagen parecida a esta:

Accesorio	Tipo	Diámetro		Material	Sch	Norma	Extremos	Observaciones
		Desde	Hasta					
tuberia & nipple	tuberia sin soldadura	1/2"	1 1/2"	A-106 Gr.B	40	ASME B 36.10	Planos	
tuberia & nipple	tuberia sin soldadura	2"	6"	A-106 Gr.B	40	ASME B 36.10	Biselados	
tuberia & nipple	tuberia sin soldadura	8"	24"	A-106 Gr.B	20	ASME B 36.10	Biselados	
tuberia & nipple	Nipples	1/2"	1 1/2"	A-106 Gr.B	80	ASME B 36.10	TBE	Long. 100 / 200 / 75
tuberia & nipple	Nipples	1/2"	1 1/2"	A-106 Gr.B	80	ASME B 36.10	TOE	Long. 100 / 200 / 75
tuberia & nipple	Nipples	1/2"	1 1/2"	A-106 Gr.B	80	ASME B 36.10	PBE	Long. 100 / 200 / 75

Figura A.8: Resultado de un Búsqueda de accesorios de una Clase

Cabe destacar que este filtro funciona para todos los tipos de accesorios, pero hay dos en particular que tienen una funcionalidad añadida.

- Si seleccionamos como tipo de accesorio **Junta**, tendremos un botón **Tabla Juntas** en cada línea asociada. Si lo pulsamos obtendremos una tabla con la información de todas las juntas disponibles y en particular la junta seleccionada remarcada.

codigo	claseJunta	tipoJunta	notas	
0	Lámina	Goma nitrilica (NBR)	ertyy	Juntas Homologadas
1-2	Lámina	Fibra de vidrio con elastómero aglomeranteresistente al ácido	ertyy	Juntas Homologadas
1-3	Lámina	Fibra sintética con aglomerante (NBR)		Juntas Homologadas
1-4	Lámina	Fibras de carbon con aglomerante (NBR)	Trabajos de cegado, PPH, etc..	Juntas Homologadas
1-5	Lámina	Lámina de PTFE		Juntas Homologadas
1-6	Lámina	Lámina de PTFE expandido		Juntas Homologadas
1-7	Lámina	Lámina de PTFE modificado		Juntas Homologadas
1-8	Lámina	Lamina de grafito con alma metalica		Juntas Homologadas
1-9	Lámina	Grafito laminado reforzado con lámina perforada AISI 316		Juntas Homologadas
1-10	Lámina	Lámina de Mica y refuerzo en AISI 316		Juntas Homologadas
1-11	Lámina	Lámina de GRAFITO		Juntas Homologadas
2-1	SPM	Espirometalica aro exterior e interior AISI 316 + PTFE, Norma API-601		Juntas Homologadas
2-2	SPM	Espirometalica aro exterior e interior AISI 316 + GRAFITO, Norma API-		Juntas Homologadas
2-3	SPM	Espirometalica AISI 316 + GRAFITO PARA BRIDAS STG		Juntas Homologadas
3-1	RTJ	Anillo metálico octogonal en AISI 304		Juntas Homologadas
3-2	RTJ	Anillo metálico octogonal en AISI 316		Juntas Homologadas
3-3	RTJ	Anillo metálico octogonal en AISI 347		Juntas Homologadas
4	x	Cordon/Cinta de fibra cerámica (con refuerzo de Inconel)	Gases altas temperaturas	Juntas Homologadas
5	x	Metaloplasticas con camisa de AISI-316 corrugado y rellén de grafito.	Uso exclusivo en equipos	Juntas Homologadas
6	x	Junta de aluminio		Juntas Homologadas
7	x	Junta EPDM dureza 88 shore A		Juntas Homologadas
8	metal	RTJ		Juntas Homologadas

Figura A.9: Búsqueda de la tabla de Juntas

Además cómo se puede observar, hay botón **Juntas Homologadas** que nos mostrará qué fabricantes nos pueden proporcionar ese tipo de junta.

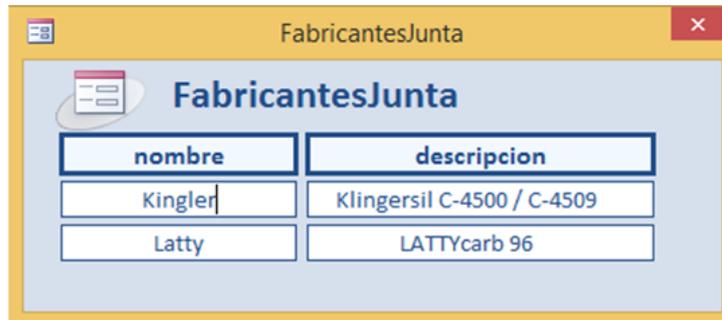


Figura A.10: Búsqueda Tabla Juntas

Si seleccionamos como tipo de accesorio Válvula , obtendremos dos botones asociados a cada línea [Ver Especificacion Valvulas](#) y [Fabricantes Homologados](#) .

- Ver Especificación Válvulas: Muestra con más detalle todos los datos propios o característicos de este tipo de válvula.

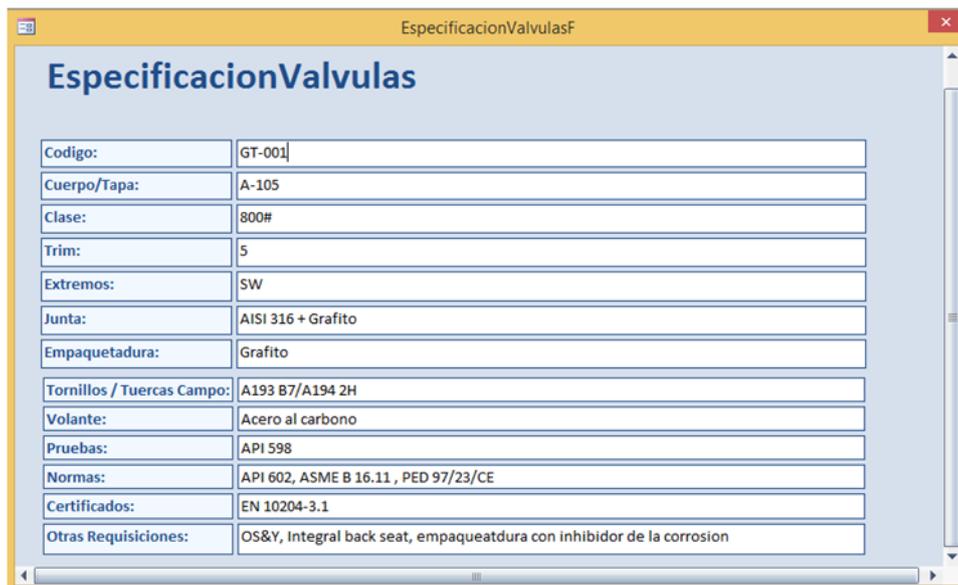


Figura A.11: Especificación de Válvulas

- Fabricantes Homologados: Muestra los diferentes fabricantes asociados a esta válvula.



Figura A.12: Fabricantes de la Válvula

### A.6.1. Tabla de Conexiones

Cada vez que realicemos una búsqueda y seleccionemos una clase, tendremos acceso a una tabla que nos permite consultar las diferentes conexiones de dicha clase pulsando en el botón Ver Tabla Conexiones **Ver Tabla Conexiones** El aspecto de la tabla es el siguiente

**TABLA DE INJERTOS Y CONEXIONES A10**

CLASE:	A11 (150#)		TABLA Y TIPO DE CONEXIONES Y REFUERZOS																											
COLECTOR	ESPESOR	Ø DERIVACION																												
		1/2"	3/4"	1"	1 1/2"	2"	3"	4"	6"	8"	10"	12"	14"	16"	18"	20"	22"	24"	26"	28"	30"	32"	34"	36"						
1/2"		T																									T	TE IGUAL O REDUCIDA.		
3/4"		TRI	T																								W	WELDOLET		
1"		TRI	T	T																							I	INJERTO DIRECTO (PARA REFUERZO VER HaE-100-201 h 51, 52)		
1 1/2"		TRI	T	T	T																						HC	SEMIMANGUITO		
2"		HCR	H	C	S	K	T	T																			HCR	SEMIMANGUITO 3/4" + SWAGE DE 1/2"		
3"		HCR	H	C	S	K	T	T																				TRI	TE IGUAL O REDUCIDA + SWAGE 3/4" x 1/2"	
4"		HCR	H	C	S	K	S	K	W	T	T																			
6"		HCR	H	C	S	K	S	K	W	W	T	T																		
8"		HCR	H	C	S	K	S	K	W	W	T	T	T																	
10"		HCR	H	C	S	K	S	K	W	W	I	I	I	T																
12"		HCR	H	C	S	K	S	K	W	W	I	I	I	I	T															
14"		HCR	H	C	S	K	S	K	W	W	I	I	I	I	I	T														
16"		HCR	H	C	S	K	S	K	W	W	I	I	I	I	I	I														
18"		HCR	H	C	S	K	S	K	W	W	I	I	I	I	I	I	I													
20"		HCR	H	C	S	K	S	K	W	W	I	I	I	I	I	I	I	I												
22"		HCR	H	C	S	K	S	K	W	W	I	I	I	I	I	I	I	I	I											
24"		HCR	H	C	S	K	S	K	W	W	I	I	I	I	I	I	I	I	I	I										

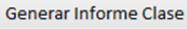
Figura A.13: Tabla de Conexiones

donde :

- La fila X indica los diferentes tamaño/ espesores de la tubería de entrada
- Las diferentes columnas Y indican el tamaño/espesor de las distintas tuberías de salida
- La posición de esa fila X para cada columna Y indica el tipo de derivación a usar para unir correctamente dichas tuberías. Estas derivaciones pueden ser del tipo:

- T= Te igual o reducida
- W= Weldolet
- I= Injerto Directo( Para Refuerzo Ver: HaE-100-201 h 51,52)
- TRI=Te igual o reducida + swage 3/4 x 1/2
- HC= Semimanguito
- SK Sockolet

### A.6.2. Generar Informe Clase

Al igual que el apartado anterior, cada vez que realicemos una búsqueda y seleccionemos una clase, tendremos acceso a un informe de dicha clase con la información tanto de todos los accesorios disponibles como de sus condiciones iniciales. Para poder visualizar este informe deberemos pulsar el botón  Generar Informe Clase

### A.6.3. Información General

En la parte derecha de la pantalla principal de la aplicación (ver Figura A.3), tenemos un menú vertical con diferentes funcionalidades.

- Índice Especificaciones. Nos abre un informe con la información principal asociada a cada clase. Esta información es la siguiente:
  - Nombre de la clase
  - Servicios Asociados a la Clase
  - Material
  - Condiciones Iniciales tanto de temperatura como de presión
  - Rating
- Índice Válvulas. Nos abre un informe con la información principal asociada a cada válvula. Esta información es toda la especificación de dicha válvula.
- Observaciones. Nos abre un informe que nos permite consultar la tabla de las distintas observaciones de los accesorios.
- Tabla Resumen Fluidos vs Clases. Nos abre un informe que nos permite consultar las clases a las que pertenecen los distintos fluidos.
- Transposición de Antiguas Especificaciones. Nos abre un informe que nos permite consultar la transposición de las antiguas especificaciones a las nuevas, por si se quiere consultar una clase en concreto.
- Informe Clases. Nos abre un informe que nos permite consultar toda la información referente a una clase en concreto. Esta información es:
  - Los servicios asociados

- Las condiciones iniciales
- La información de todos los accesorios
- La tabla de conexiones
- Informe Válvulas y Fabricantes Homologados. Nos abre un informe que nos permite consultar toda la información referente a una válvula en concreto. Esta información es:
  - La especificación completa de la válvula
  - Los fabricantes homologados asociados
- Tabla de Juntas. Nos abre un informe que nos permite consultar la información referente a todas las juntas ( clase, tipo y nombre) y sus juntas homologadas.

## A.7. Histórico

Desde el formulario de navegación(ver Figura A.2) se accede al histórico donde se puede ver un listado de los datos que han sido gestionados en nuestra aplicación.

nombreUsuario	tabla	accion	fecha	productoInicial	productoFinal	descripcion
jorge	Elementos	Insertion			cao	
jorge	Elementos	Insertion	17/12/2015 8:57:07		coa	
jorge	Elementos	Insertion	17/12/2015 9:23:38		tipo	Se añadido un cap con diametro minimo 12" diametro maximo 22" y de material hierro
jorge	Elementos	Modificar	17/12/2015 9:48:26		tipo	Se añadido un cap con diametro minimo 12" diametro maximo 22" y de material hierro
jorge	Elementos	Modificar	17/12/2015 10:35:58	90	99	Se añadido un cap con diametro minimo 12" diametro maximo 22" y de material hierro
jorge	Elementos	Modificar	17/12/2015 10:40:29			Se añadido un brida con diametro minimo 2" diametro maximo 33" y de material queeq
jorge	Elementos	Modificar	17/12/2015 10:45:28	Diámetro Mínimo : 12" Material :	Diámetro Mínimo : 12" Materis :	Se ha modificado un cap con diametro minimo 12" diametro maximo 22" y de material f
jorge	Elementos	Borrado	17/12/2015 10:53:13	cao		Se ha borrado un codo con diametro minimo 12 diametro maximo 33 y de material dpe
jorge	Servicio	Insertion	17/12/2015 12:20:49		Nitrato Sodico	Se añadido el servicio Nitrato Sodico con el simbolo No2
jorge	Servicio	Modificar	17/12/2015 12:21:02	Simbolo : No2	Simbolo : N	Se ha modificado el servicio Nitrato Sodico con simbolo No2 por el servicio Nitrato
jorge	Servicio	Borrado	17/12/2015 12:21:10	Agua Caliente		Se ha borrado el servicio Agua Caliente con simbolo HW
jorge	Servicio	Borrado	17/12/2015 12:24:23	ewr		Se ha borrado el servicio ewr con simbolo egeq
jorge	Servicio	Modificar	17/12/2015 13:24:02	Simbolo : fdf	Simbolo : yt	Se ha modificado el servicio few con simbolo fdf por el servicio few con simbolo yt
jorge	Servicio	Insertion	17/12/2015 13:24:40		tres	Se añadido el servicio tres con el simbolo uy
jorge	Servicio	Borrado	17/12/2015 13:24:50	tres		Se ha borrado el servicio tres con simbolo uy
jorge	Servicio	Insertion	18/12/2015 8:39:03		Acido Nitrico	Se añadido el servicio Acido Nitrico con el simbolo N
jorge	Servicio	Modificar	18/12/2015 8:39:20	Simbolo : N	Simbolo : No2	Se ha modificado el servicio Acido Nitrico con simbolo N por el servicio Acido Nitr
jorge	Servicio	Borrado	18/12/2015 8:39:30	Acido Nitrico		Se ha borrado el servicio Acido Nitrico con simbolo N
jorge	Servicio	Insertion	18/12/2015 9:21:36		Nitrico	Se añadido el servicio Nitrico con el simbolo No2
jorge	Elementos	Insertion	21/12/2015 10:36:04		tyo	Se añadido un brida con diametro minimo 12" diametro maximo 22" y de material AS
jorge	Servicio	Insertion	21/12/2015 11:11:43		a2	Se añadido el servicio a2 con el simbolo a2
jorge	Servicio	Insertion	21/12/2015 11:11:55		a3	Se añadido el servicio a3 con el simbolo a3
jorge	Servicio	Insertion	21/12/2015 11:12:52		a4	Se añadido el servicio a4 con el simbolo a4
jorge	Servicio	Insertion	21/12/2015 11:13:03		a5	Se añadido el servicio a5 con el simbolo a5
jorge	Servicio	Borrado	21/12/2015 11:13:24	a5		Se ha borrado el servicio a5 con simbolo a5

Figura A.14: Tabla con el histórico de cambios en la aplicación

La tabla Histórico contiene los siguientes datos:

- NombreUsuario: Es el nombre del usuario que ha realizado el cambio
- Tabla: Indica la tabla donde se ha realizado el cambio
- Acción: Indica el tipo de cambio realizado( inserción, modificación o borrado)
- Fecha: Momento en el cual se produjo el cambio
- ProductoInicial: Indica la información inicial del dato a cambiar. Si el producto inicial es nulo significa que el tipo de acción es de inserción y por tanto antes de añadir ese dato, anteriormente no había nada.

- ProductoFinal: Indica la información del dato una vez se ha producido el cambio. Si el producto final aparece vacío, significa que el tipo de acción es de borrado y que por tanto no existe información final debido a que ha sido eliminada de la base de datos.
- Descripción: Es una pequeña descripción que nos indica que tipo de cambio ha sido realizado.

## A.8. Formulario de Mantenimiento

Desde el formulario de mantenimiento (ver Figura A.15), se pueden gestionar los diferentes datos maestros de la aplicación PippingClass. Este mantenimiento se realizará tanto de los servicios disponibles como de las clases y los accesorios asociados, y el encargado realizar esta funciones será el usuario de mantenimiento.

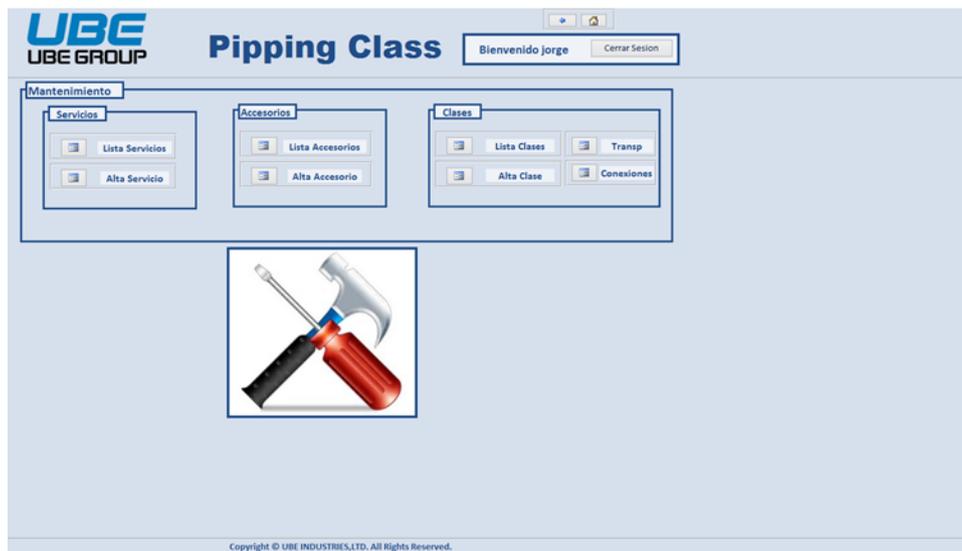


Figura A.15: Formulario de Mantenimiento

Si observamos detenidamente el aspecto de la ventana de mantenimiento, podemos observar que el mantenimiento se puede realizar sobre tres tipos de datos maestros (servicios, accesorios y clases). Estos podrán ser dados de alta directamente o listados y, a partir de esta opción, tanto modificarlos como borrarlos de la base de datos.

### A.8.1. Mantenimiento de Servicios

Pulsando en el botón Alta Servicio del formulario de mantenimiento accedemos al formulario para dar de alta un servicio nuevo.

En este formulario hay dos campos distintos a insertar : el nombre del servicio y su símbolo. Si pulsamos el botón Registrar  añadiremos el nuevo servicio adecuada-

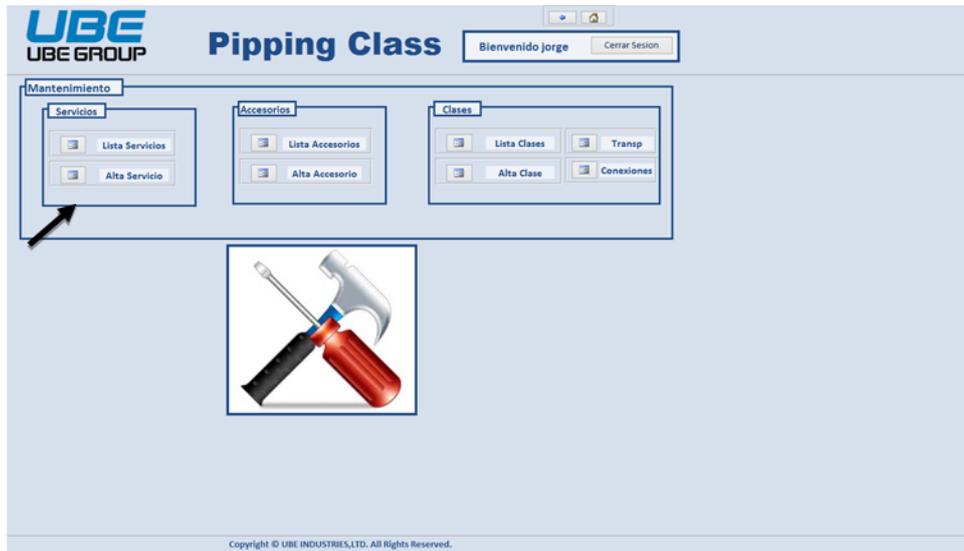


Figura A.16: Acceso al alta de servicios desde el formulario de mantenimiento

mente siempre que no exista ese servicio. Si pulsamos el botón **Cancelar** resetearemos los datos introducidos en el formulario. Si queremos salir del formulario porque hemos entrado erróneamente simplemente debemos pulsar el botón **X**.

Figura A.17: Alta Servicio

Los campos del Servicio son:

- Servicio: Nombre del servicio. No pueden existir de servicios iguales, es decir, que tengan el mismo nombre.
- Símbolo: Símbolo asociado a cada servicio.

Pulsando el botón Lista Servicios del formulario mantenimiento accedemos al formulario con la lista de servicios disponibles actuales.

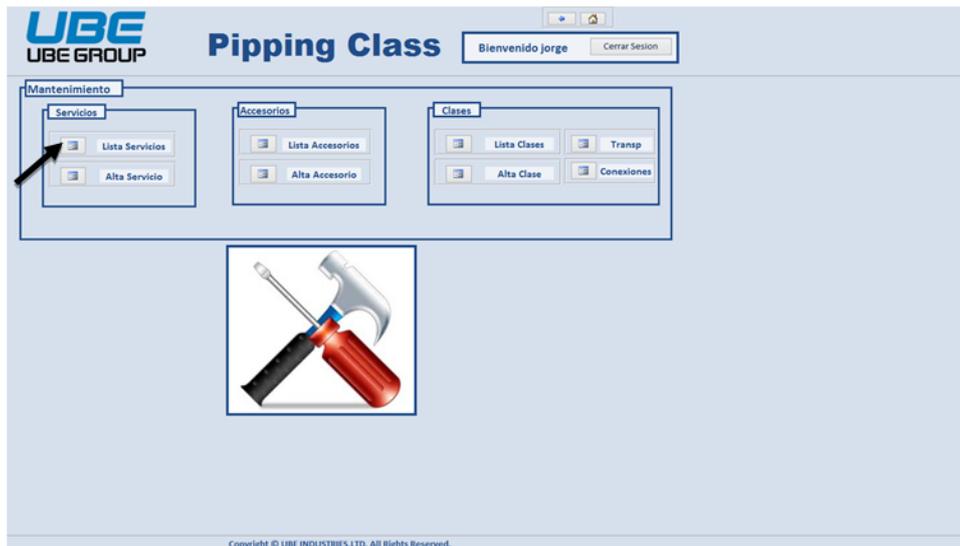


Figura A.18: Acceso a la lista de servicios desde el formulario de mantenimiento

A parte de poder consultar la lista de servicios disponibles, en este formulario tenemos diversas opciones de funcionamiento. Los botones que encontramos en las diferentes líneas nos permiten:

- **Botón edición**  : Abrir el formulario de edición con los datos del tipo de servicio seleccionado en la tabla
- **Botón borrado**  : Borrar el servicio seleccionado en la tabla y por tanto en la base de datos. Esta acción sólo podrá realizarse si previamente se borra dicho servicio en las asociaciones existentes en los diferentes clases.

### Botones de Gestión y Navegación

En la parte superior del formulario de mantenimiento de la lista de servicios existen los siguientes botones de gestión y navegación:

- **Atrás**  : Vuelve a la página anterior. En este caso es el formulario de mantenimiento.
- **Home**  : Entra en la página principal de navegación.
- **Crear Servicio**  : Abre el formulario de alta servicio vacío para introducir un nuevo tipo de servicio. Realiza la misma funcionalidad que alta servicio pero se consigue acceder a este formulario sin necesidad de estar anteriormente en el formulario de mantenimiento.

servicio	simbolo		
Agua Caliente	HW		
Agua Condensada	CW		
werrw	werwer		
Acido Sulfurico	S		
Nitrico	No2		
Acido	Ac		
Agua Contraincendios	AI		
Agua Clorada	AC		
Agua Refrigeración	ART		
Ah	Ah		
Ad	Ad		

Figura A.19: Lista Servicios

### A.8.2. Mantenimiento de Accesorios

Pulsando en el botón Alta Accesorio de la pestaña de mantenimiento accedemos al formulario para dar de alta un accesorio nuevo.

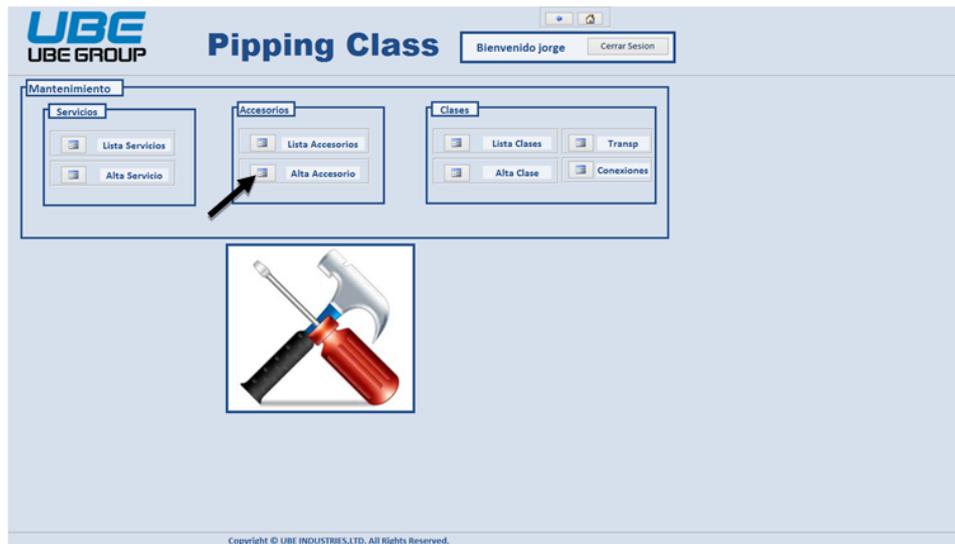


Figura A.20: Acceso al alta de accesorios desde el formulario de mantenimiento

Al igual el caso de Alta Servicio, si pulsamos el botón Registrar  añadiremos el nuevo accesorio adecuadamente siempre que no exista previamente. Si pulsamos el botón Cancelar  resetearemos los datos introducidos en el formulario. Si queremos salir del formulario por que hemos entrado erróneamente simplemente debemos pulsar el botón

Figura A.21: Formulario Alta Accesorio

El formulario de Alta Accesorio contiene los siguientes campos a rellenar:

- Accesorio : Tipo de accesorio
- Tipo: Nombre del nuevo accesorio a añadir en la base de datos
- Desde: Diámetro mínimo del accesorio
- Hasta: Diámetro máximo del accesorio
- Material: Material del accesorio
- Norma: Norma del accesorio
- Extremos: Tipo de extremos del accesorio
- Rating
- Schedule
- Observaciones

Si el accesorio a añadir es de tipo válvula o de tipo de junta tendremos una función añadida.

- Si el accesorio es de tipo válvula, tendremos que indicar la especificación de la válvula para poder dar de alta este accesorio correctamente. Para ello habrá que pulsar el botón  También se observa que aparece el código de la válvula, pero este no hace falta añadirlo en este formulario ya que se importará correctamente si se ha realizado la especificación de válvula de una forma adecuada.

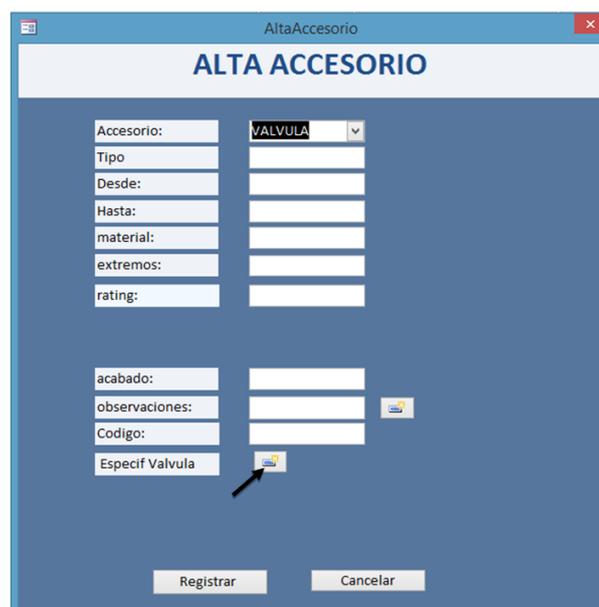


Figura A.22: Formulario Alta Accesorio

En la especificación de la válvula habrá que rellenar los campos descrito a continuación y pulsar el botón save  para que los cambios se guarden correctamente.

- Código
- Cuerpo/Tapa
- Clase
- Trim
- Extremos
- Junta
- Empaquetadura
- Tornillos/Tuercas Cuerpo
- Volante
- Pruebas
- Normas
- Certificados
- Otras Requisiciones

Figura A.23: Formulario Especificación Válvula

- Si el accesorio es de tipo junta, al igual que en el caso anterior tendremos que indicar la especificación de la junta para poder dar de alta este accesorio correctamente. Para ello habrá que pulsar el botón  También se observa que aparece el código de la junta, pero este no hace falta añadirlo en este formulario ya que se importará correctamente si se ha realizado la especificación de la junta de una forma adecuada.

Figura A.24: Formulario Alta Accesorio

La especificación de la junta se podrá hacer de dos formas diferentes:

Figura A.25: Formulario Especificación Junta

1. **Crear Una nueva Especificación** Se abrirá el formulario para añadir la especificación de la junta desde cero.

Figura A.26: Formulario Especificación Junta

En este formulario se añadirán :

Código

Clase de la Junta

Tipo de Junta

Se deberá pulsar el botón save  para asociar correctamente esta especificación a la Junta.

2. **Asociar Especificación** Asignar una especificación ya existente a este junta que se quiere añadir

Figura A.27: Asociar Especificación Junta

En este formulario simplemente habrá que seleccionar el tipo junta a través del cual asignaremos la especificación de válvula ya existente. Al igual que el caso anterior se deberá pulsar el botón save  para asociar correctamente esta especificación a la Junta.

Pulsando el botón Lista Accesorios de la pestaña de mantenimiento accedemos al formulario con la lista de accesorios disponibles actuales.

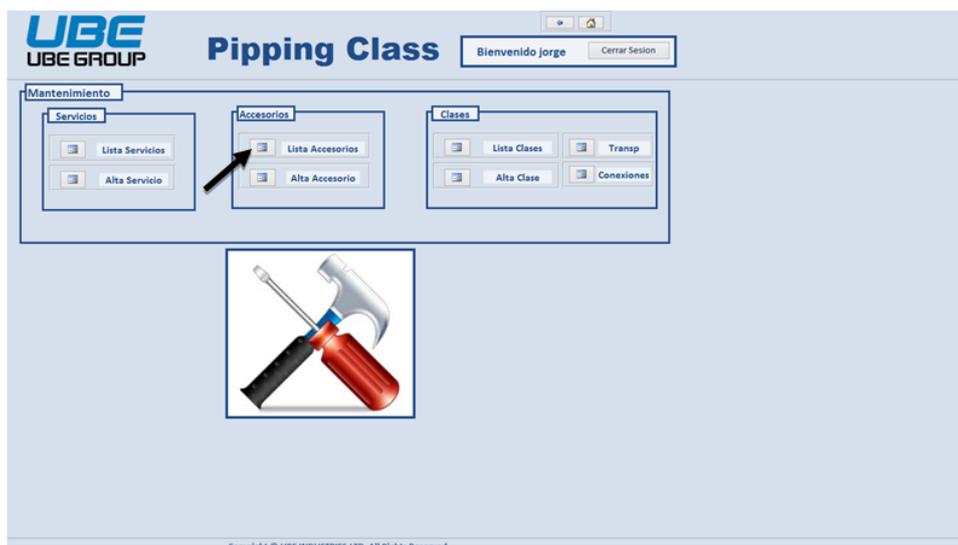


Figura A.28: Acceso a la lista de accesorios desde el formulario de mantenimiento

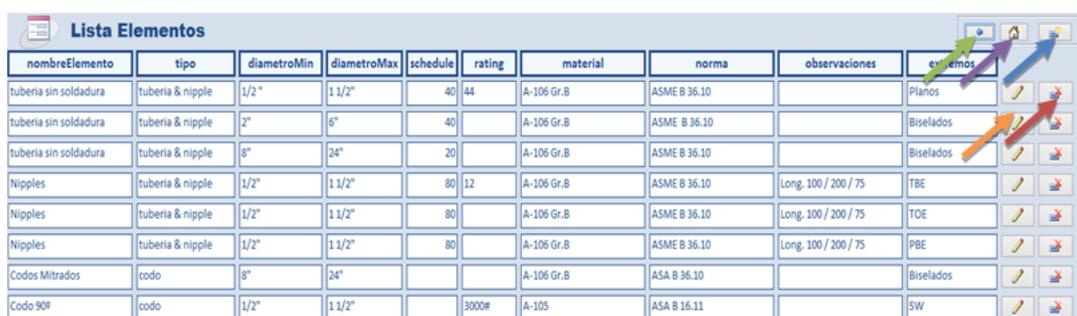
A parte de poder consultar la lista de accesorios disponibles, en este formulario tenemos diversas opciones de funcionamiento. Los botones que encontramos en las diferentes líneas nos permiten:

- **Botón edición**  : Abrir el formulario de edición con los datos del tipo de accesorio seleccionado en la tabla
- **Botón borrado**  : Borrar el accesorio seleccionado en la tabla y por tanto en la base de datos. Esta acción sólo podrá realizarse si previamente se borra dicho accesorio en las asociaciones existentes en los diferentes clases.

## Botones de Gestión y Navegación

En la parte superior del formulario de lista de accesorios existen los siguientes botones de gestión y navegación:

- **Atrás**  : Vuelve a la página anterior. En este caso es el formulario de mantenimiento.
- **Home**  : Entra en la página principal de navegación.
- **Crear Servicio**  : Abre el formulario de alta accesorio vacío para introducir un nuevo tipo de accesorio. Realiza la misma funcionalidad que alta accesorio pero se consigue acceder a este formulario sin necesidad de estar anteriormente en el formulario de mantenimiento.



nombreElemento	tipo	diametroMin	diametroMax	schedule	rating	material	norma	observaciones	especificamos
tuberia sin soldadura	tuberia & nipple	1/2"	1 1/2"	40	44	A-106 Gr.B	ASME B 36.10		Planos
tuberia sin soldadura	tuberia & nipple	2"	6"	40		A-106 Gr.B	ASME B 36.10		Biselados
tuberia sin soldadura	tuberia & nipple	8"	24"	20		A-106 Gr.B	ASME B 36.10		Biselados
Nipples	tuberia & nipple	1/2"	1 1/2"	80	12	A-106 Gr.B	ASME B 36.10	Long. 100 / 200 / 75	TBE
Nipples	tuberia & nipple	1/2"	1 1/2"	80		A-106 Gr.B	ASME B 36.10	Long. 100 / 200 / 75	TOE
Nipples	tuberia & nipple	1/2"	1 1/2"	80		A-106 Gr.B	ASME B 36.10	Long. 100 / 200 / 75	PBE
Codos Mitrados	codo	8"	24"			A-106 Gr.B	ASA B 36.10		Biselados
Codo 90º	codo	1/2"	1 1/2"		3000#	A-105	ASA B 16.11		SW

Figura A.29: Editar Accesorio

Si editamos un accesorio general (cualquiera menos válvula y junta) simplemente tendremos que modificar el campo o campos a modificar y pulsar el botón de modificar. Si editamos una válvula tendremos varias funcionalidades extra a recalcar:

- **Inserción/Edición Especificación Válvula.** Se podrán modificar los datos de la especificación de dicha válvula. Si previamente se ha borrado la especificación de la válvula, se abrirá el formulario de inserción de una nueva especificación.
- **Borrar Especificación Válvula** Se borrará la especificación de la válvula seleccionada.
- **Lista Fabricantes** Se abrirá el formulario para asociar fabricantes a este tipo de válvula.

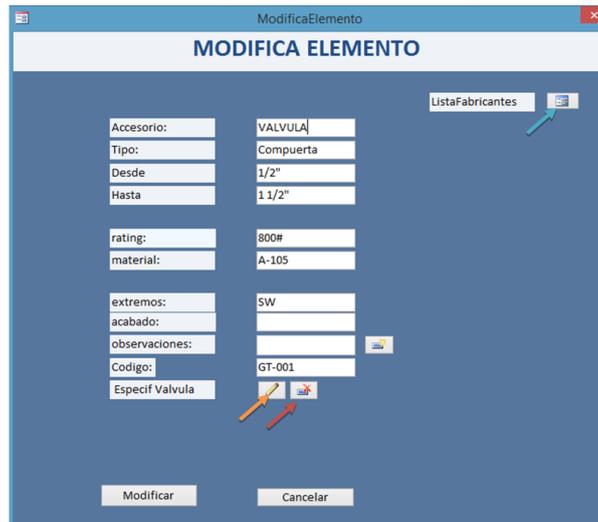


Figura A.30: Editar Accesorio

Si seleccionamos la lista de fabricantes se nos abrirá un formulario como el de la Figura A.31.



Figura A.31: Gestión Fabricantes Válvula

En la parte izquierda podemos consultar los fabricantes disponibles y en la parte derecha los fabricantes asociados a esta válvula cuyo código podemos observar en la parte central.

Simplemente si queremos añadir un fabricante deberemos pulsar el botón 'añadir' y si queremos eliminarlo deberemos pulsar el botón 'borrar'

Al igual que el caso anterior para que se guarden los datos deberemos pulsar el botón save  En el caso de que queramos añadir un fabricante no existente, habrá que pulsar el botón  que nos abrirá un formulario para gestionar los fabricantes.

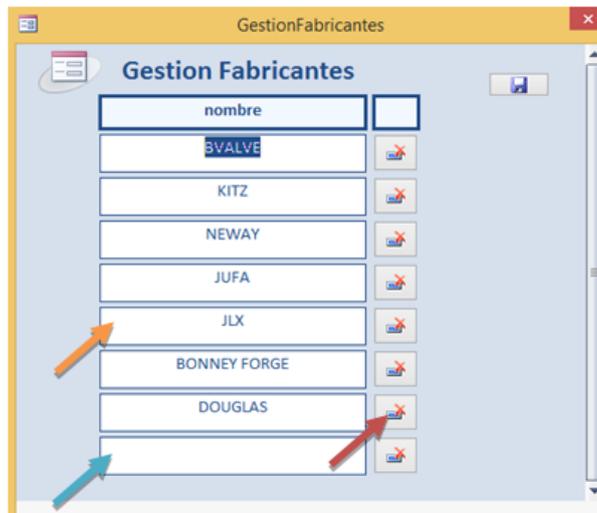


Figura A.32: Gestión Fabricantes

Simplemente se podrá:

- **Añadir** uno nuevo introduciéndolo por teclado
- **Modificarlo** substituyendo su valor
- **Borrarlo** pulsando el botón

Si editamos una junta tendremos varias funcionalidades extra a recalcar:

- **Inserción/Edición Especificación Junta.** Se podrán modificar los datos de la especificación de dicha junta. Si previamente se ha borrado la especificación de la junta, se abrirá el formulario de inserción de una nueva especificación.
- **Borrar Especificación Válvula.** Se borrará la especificación de la junta asociada a dicha junta.
- **Lista Fabricantes** Se abrirá el formulario para asociar fabricantes a este tipo de junta.
- **Tabla Juntas** Se abrirá el formulario de la tabla de juntas para su gestión.

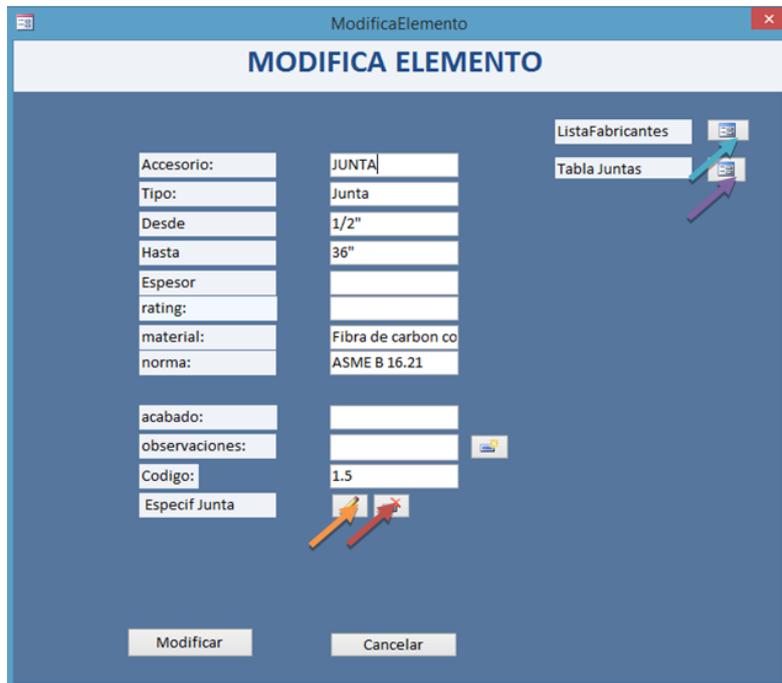


Figura A.33: Modifica Elemento

Si seleccionamos la lista de fabricantes se nos abrirá un formulario como este.



En la parte izquierda podemos consultar los fabricantes disponibles y en la parte derecha los fabricantes asociados a esta válvula cuyo código podemos observar en la parte central.

Simplemente si queremos añadir un fabricante deberemos pulsar el botón añadir y si queremos eliminarlo deberemos pulsar el botón borrar

Al igual que el caso anterior para que se guarden los datos deberemos pulsar el botón save  En el caso de que queramos añadir un fabricante no existente, habrá que pulsar el botón  que nos abrirá un formulario para gestionar los fabricantes.

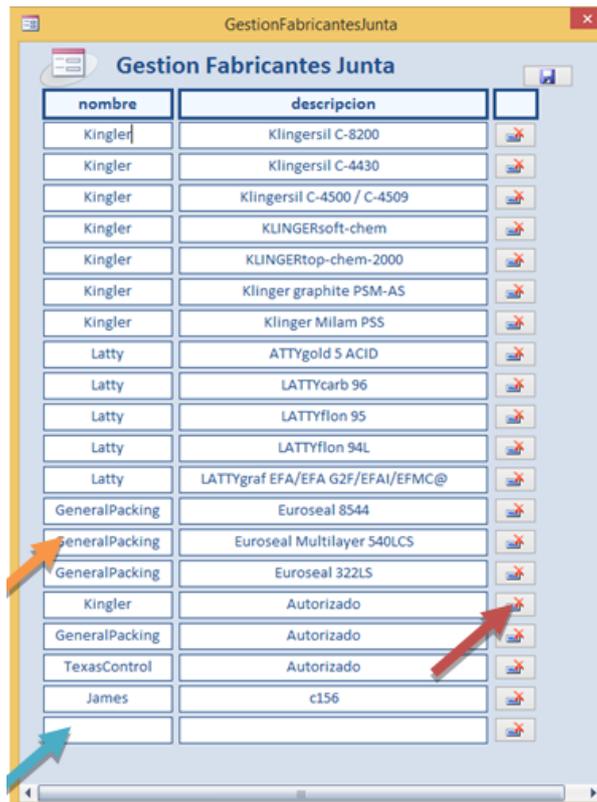


Figura A.35: Gestión Fabricantes Junta

Simplemente se podrá:

- **Añadir** uno nuevo introduciéndolo por teclado
- **Modificarlo** substituyendo su valor
- **Borrarlo** pulsando el botón

Si seleccionamos la tabla de juntas se nos abrirá un formulario como este donde podremos **eliminar** la especificación de alguna junta en el caso que se necesario. Simplemente habrá que pulsar el botón .

codigo	claseJunta	tipoJunta	notas
1-1	Lámina	Goma nitrilica (NBR)	ertyy
1-2	Lámina	Fibra de vidrio con elastómero aglomeranteresistente al ácido	ertyy
1-3	Lámina	Fibra sintética con aglomerante (NBR)	
1-4	Lámina	Fibras de carbon con aglomerante (NBR)	Trabajos de cegado, PPH, etc...
1-5	Lámina	Lámina de PTFE	
1-6	Lámina	Lámina de PTFE expandido	
1-7	Lámina	Lámina de PTFE modificado	
1-8	Lámina	Lamina de grafito con alma metalica	
1-9	Lámina	Grafito laminado reforzado con lámina perforada AISI 316	
1-10	Lámina	Lámina de Mica y refuerzo en AISI 316	
1-11	Lámina	Lámina de GRAFITO	
2-1	SPM	Espirometalica aro exterior e interior AISI 316 + PTFE, Norma API-601	
2-2	SPM	Espirometalica aro exterior e interior AISI 316 + GRAFITO, Norma API-	
2-3	SPM	Espirometalica AISI 316 + GRAFITO PARA BRIDAS STG	
3-1	RTJ	Anillo metálico octogonal en AISI 304	
3-2	RTJ	Anillo metálico octogonal en AISI 316	
3-3	RTJ	Anillo metálico octogonal en AISI 347	
4	x	Cordon/Cinta de fibra cerámica (con refuerzo de Inconel)	Gases altas temperaturas
5	x	Metaloplasticas con camisa de AISI-316 corrugado y rellén de grafito.	Uso exclusivo en equipos
6	x	Junta de aluminio	
7	x	Junta EPDM dureza 88 shore A	
8	metal	RTJ	

Figura A.36: Tabla Juntas

Cabe destacar que existe una funcionalidad que funciona igual para todo todos los tipos de accesorios, y es el caso de añadir observaciones Si se desea añadir una o varias observaciones se pulsará el botón  colocado al lado del campo observaciones.

observaciones:  

Figura A.37: Observaciones

codigo	Descripcion	Añadir
1	Algo	Añadir
2	Nada	Añadir
3	Limit the use of these elements to strictly necessary cases	Añadir
4	datafada	Añadir
5	oo	Añadir
23	datafada	Añadir
22	1/16 "R"	Añadir

Figura A.38: Gestión Observaciones Accesorio

En la parte izquierda podemos consultar observaciones disponibles y en la parte derecha las observaciones asociadas a este accesorio.

Simplemente si queremos añadir un fabricante deberemos pulsar el botón añadir y si queremos eliminarlo deberemos pulsar el botón borrar

Al igual que el caso anterior para que se guarden los datos deberemos pulsar el botón save



En el caso de que queramos añadir una observación no existente, habrá que pulsar el botón  que nos abrirá un formulario para gestionar los fabricantes.

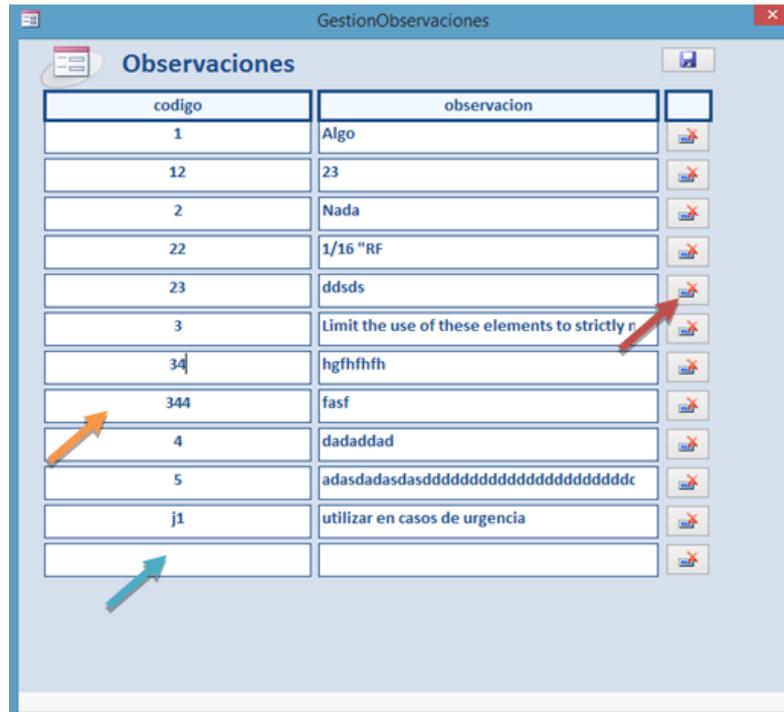


Figura A.39: Gestión Observaciones

Simplemente se podrá:

- **Añadir** uno nuevo introduciéndolo por teclado
- **Modificarlo** substituyendo su valor
- **Borrarlo** pulsando el botón 

### A.8.3. Mantenimiento de Clases

Pulsando en el botón Alta Clase de la pestaña de mantenimiento accedemos al formulario para dar de alta una clase nueva.

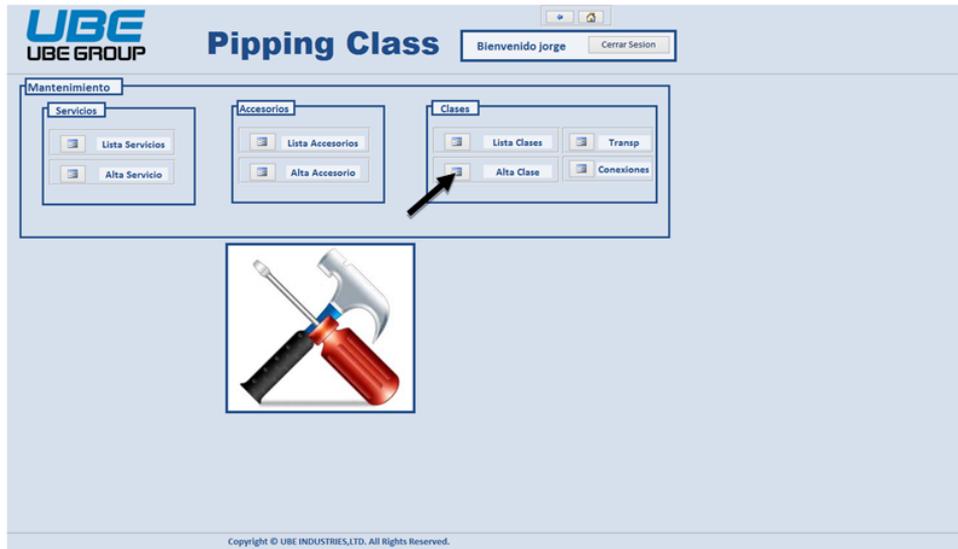


Figura A.40: Alta Clase

Para dar de alta una clase correctamente, habrá que seguir los siguientes pasos:

- Introduce el nombre de la clase y el material asociado a esta. Si el nombre de la clase ya existe se le informará adecuadamente y deberá introducir un nombre distinto

- A continuación se seleccionarán los servicios que se deseen añadir a dicha clase. Simplemente hay que indicar el margen de corrosión del servicio y pulsar el botón de añadir.

Servicio	Símbolo	Añadir
Agua Caliente	HW	Añadir
Agua Condensada	CW	Añadir
Vertido	water	Añadir
Aspiración	S	Añadir

- Cómo se puede apreciar en la imagen anterior, en la parte izquierda podremos consultar los servicios disponibles de la base de datos. Y que se añadirán a la clase nueva mediante el botón 'añadir' Siempre teniendo en cuenta que se ha introducido el margen de corrosión adecuado

Figura A.41: Alta Clase

Cabe destacar que si el servicio que se desea introducir no existe, solamente pulsando en el botón  Se nos abrirá el formulario de creación de nuevo servicio. En la parte

derecha podemos consultar la información inicial de todos los servicios asociados a esta nueva clase.

Servicio	mC	Material	
Agua Caliente	3,2	A11	Borrar
Agua Condensada	3,2	A11	Borrar

Figura A.42: Información Inicial

En el caso que hayamos añadido un servicio erróneo, solo tendremos que pulsar el botón borrar **Borrar** para eliminarlo de la clase.

- Una vez introducido todos los servicios, tendremos que añadir los accesorios que contendrá esta clase. El funcionamiento es bastante similar al de inserción de servicios.

Filtro Accesorios Disponibles  
TipoAccesorio: CODO

Nombre	Tipo	Desde	Hasta	Sch	rating	material	norma	extremos	observaciones	
CODO	Codos Mitrados	8"	24"			A-106 Gr.B	ASME B 36.10	Biselados		Añadir
CODO	Codo 90º	1/2"	1 1/2"		3000#	A-105	ASME B 16.11	SW		Añadir
CODO	Codo 90º	2"	6"		40	A-234 Gr.WPB	ASME B 16.9	Biselados		Añadir

Filtro Accesorio Clase  
TipoAccesorio: CODO

nombreElemento	tipo	diametroMin	diametroMax	Sch	rating	material	norma	extremos	observaciones	
Codos Mitrados	CODO	8"	24"			A-106 Gr.B	ASME B 36.10	Biselados		Borrar

Figura A.43: Accesorios de la Clase

- Tendremos un filtro de accesorios disponibles mediante el cual seleccionaremos el tipo de accesorio a añadir y simplemente deberemos pulsar el botón añadir **Añadir** para asignarlo adecuadamente

- También tendremos un filtro de accesorios clase que nos mostrará todos los accesorios asignados a la clase según el tipo de accesorio por el cual hayamos filtrado anteriormente.

- Al igual que el caso anterior hay 2 opciones que nos facilitan el mantenimiento de la clase. La primera es el botón borrar **Borrar** que nos permite revertir el cambio en el caso de que hayamos asignado erróneamente un accesorio a la clase. La segunda es el botón que nos permite añadir directamente un accesorio si no se encuentra en los elementos disponibles.

- - Por último asignaremos las condicionales iniciales tanto de temperatura y presión a la clase.

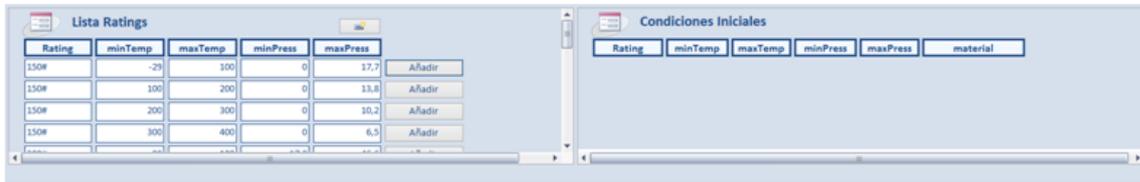


Figura A.44: Ratings de la Clase

El funcionamiento es exactamente igual que los dos casos anteriores. Si se quiere añadir un condición pulsamos el botón añadir **Añadir** . Si se quiere eliminar una condición pulsamos el botón borrar **Borrar** . Y se quiere añadir una nueva condición inicial por que no existe la que deseamos, pulsamos el botón  el cual nos abrirá el formulario de alta de rating.

Para guardar la información añadida correctamente tendremos el botón guardar colocado en la parte superior derecha junta a la barra de navegación.



Pulsando en el botón Lista Clases del formulario de mantenimiento accedemos al formulario con la lista de clases disponibles actuales.

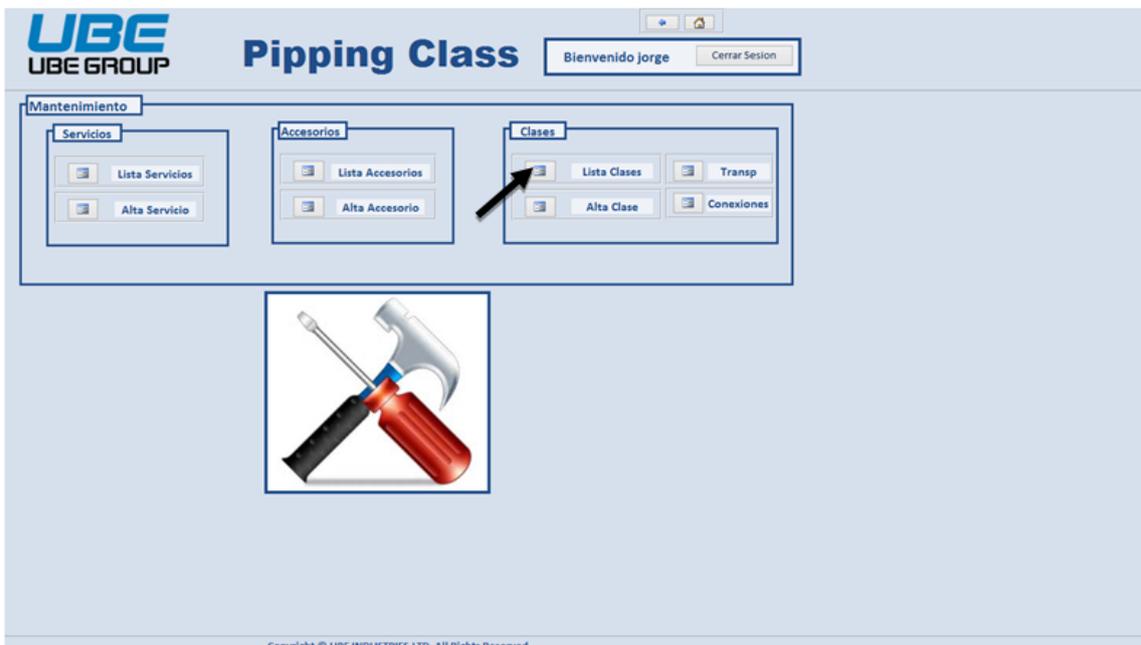


Figura A.45: Acceso a la lista de clases desde el formulario de mantenimiento

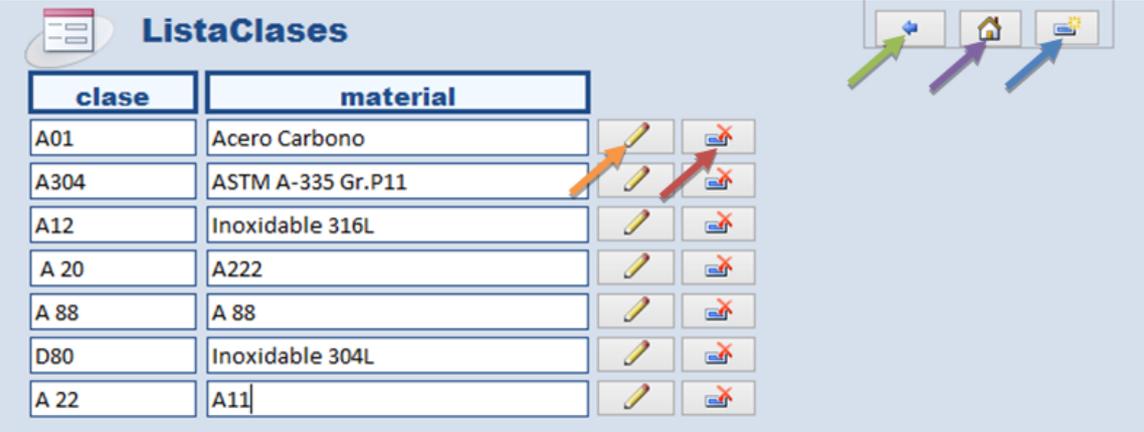
A parte de poder consultar la lista de clases disponibles, en este formulario tenemos diversas opciones de funcionamiento. Los botones que encontramos en las diferentes líneas nos permiten:

- **Botón edición**  : Abrir el formulario de edición con los datos del tipo de clase seleccionado en la tabla
- **Botón borrado**  : Borrar la clase seleccionada en la tabla y por tanto en la base de datos.

### Botones de Gestión y Navegación

En la parte superior del formulario de lista de accesorios existen los siguientes botones de gestión y navegación:

- **Atrás**  : Vuelve a la página anterior. En este caso es el formulario de mantenimiento.
- **Home**  : Entra en la página principal de navegación.
- **Crear Servicio**  : Abre el formulario de alta clase vacío para introducir un nuevo tipo de clase. Realiza la misma funcionalidad que alta clase pero se consigue acceder a este formulario sin necesidad de estar anteriormente en el formulario de mantenimiento.



clase	material		
A01	Acero Carbono		
A304	ASTM A-335 Gr.P11		
A12	Inoxidable 316L		
A 20	A222		
A 88	A 88		
D80	Inoxidable 304L		
A 22	A11		

Figura A.46: Lista de Clases

Pulsando en el botón Transp del formulario de mantenimiento accedemos al formulario con la lista de las transposiciones de clases actuales.

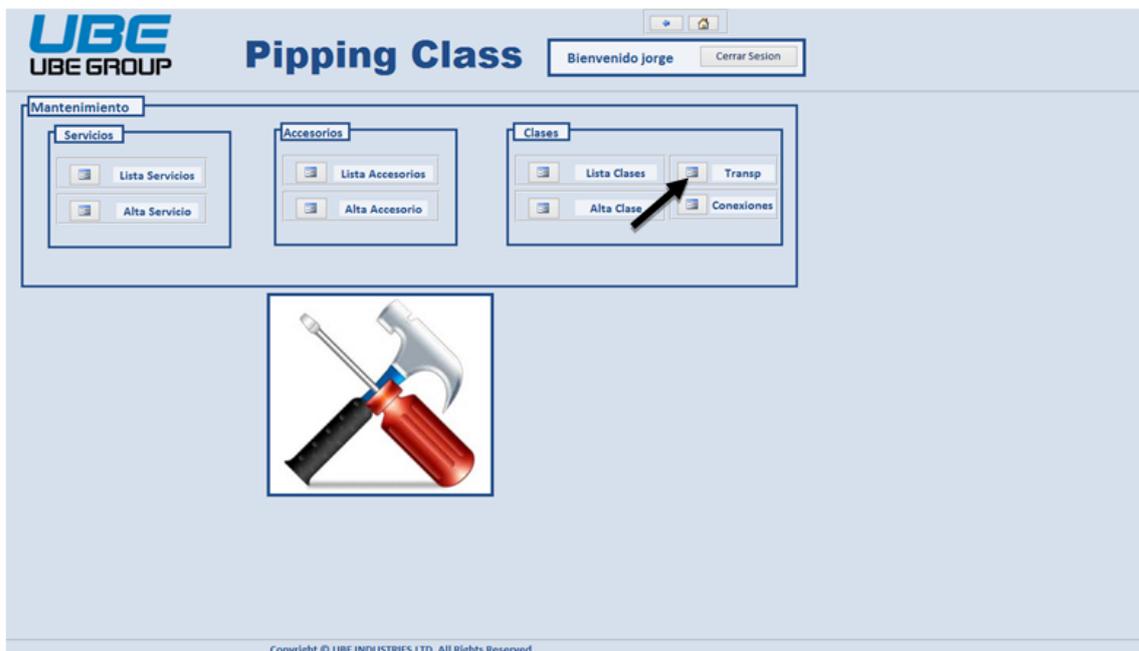


Figura A.47: Acceso a la transposición de clases desde el formulario de mantenimiento

A parte de poder consultar la lista de transposiciones de clases existentes, en este formulario tenemos diversas opciones de funcionamiento. Los botones que encontramos en las diferentes líneas nos permiten:

- **Botón borrado**  : Borrar la clase seleccionada en la tabla y por tanto en la base de datos.

### Botones de Gestión y Navegación

En la parte superior del formulario existen los siguientes botones de gestión y navegación:

- **Crear Nueva Transp**  : Abre el formulario de alta transposición vacío para introducir un nuevo tipo de transposición de clase.

Simbolo	Servicio	Clase				Propuesta Definitiva
		Uche	Nylon	Hdl	PCD/PDL	
AR	Agua Refrigeración	A01B	A2s	A3a	A01B	A10
RC	Raw Chips	A21	A22	A22	A22	A2k

Figura A.48: Transposición Antiguas Especificaciones

El aspecto del formulario de alta transposición es el siguiente:

**ALTA TRANSPOSICION CLASE**

Servicio:

Clase:

uche:

hdl:

nylon:

pcd:

Definitiva:

Figura A.49: Alta Transposición

Pulsando en el botón Conexiones de la pestaña de mantenimiento accedemos al formulario que nos permitirá editar las fotos de la tabla de injertos y de conexiones de cada una de las clases.

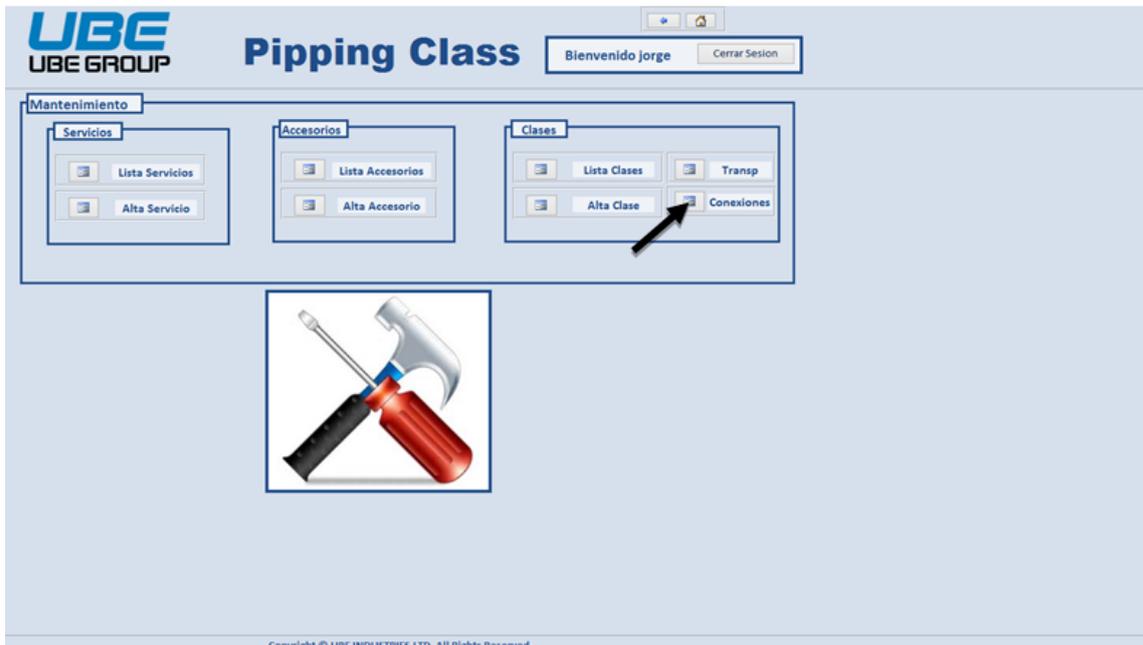


Figura A.50: Acceso a la conexiones desde el formulario de mantenimiento

A parte de poder consultar la tabla de injertos y de conexiones, en este formulario tenemos diversas opciones de funcionamiento. Los botones que encontramos en las diferentes líneas nos permiten:

- **Botón Ruta**  : Permite establecer la ruta de la imagen de la tabla de injertos y conexiones.
- **Botón Guardado**  : Permite guardar el cambio realizado anteriormente.

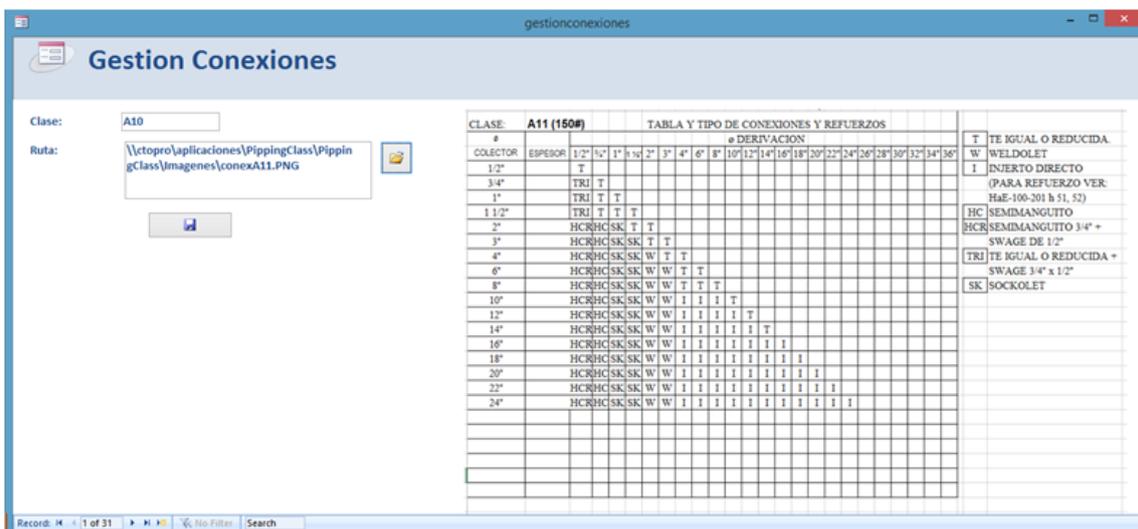


Figura A.51: Gestión Conexiones

Cabe destacar que facilitar el manejo de acceso a las distintas clases, se dispone de una barra de navegación que nos permite tanto filtrar como navegar desde una clase a otra.

