



GRADO EN INGENIERÍA INFORMÁTICA

TRABAJO DE FINAL DE GRADO

Módulo Gestión Maquinaria

Autor:
Daniel ESTREMIANA LÓPEZ

Supervisor:
Sergi VILAR DOMENECH
Tutor académico:
Dolores MARÍA LLIDÓ ESCRIVÁ

Fecha de lectura: 12 de julio de 2023
Curso académico 2022/2023

Resumen

El presente documento detalla la memoria del trabajo de final de grado realizado por el alumno Daniel Estremiana López de la Universitat Jaume I de Castellón durante la estancia en prácticas en la empresa Innova Advanced Consulting S.L. Esta memoria contiene el proceso de creación de un proyecto que tiene como objetivo el desarrollo de un módulo para la gestión de maquinaria.

Para el desarrollo de este proyecto se ha utilizado la herramienta de Microsoft, Business Central, ya que es la que mejor se adapta a las necesidades del producto final.

Este módulo es capaz de gestionar los datos de máquinas, clientes, y contratos, así como de crear y gestionar partes de avería. Además, también es capaz de automatizar el proceso de facturación de los contratos.

Palabras clave

Gestión de alquiler de maquinaria, facturación, Microsoft Business Central, ERP

Keywords

Machine rental management, invoicing, Microsoft Business Central, ERP

Índice general

1. Introducción	9
1.1. Contexto y motivación del proyecto	9
1.2. Objetivo y alcance del proyecto	9
1.3. Objetivo y alcance del producto	10
1.3.1. Alcance funcional	10
1.3.2. Alcance organizacional	10
1.3.3. Alcance informático	11
1.4. Descripción del proyecto	11
1.4.1. Tecnologías	12
1.5. Estructura de la memoria	13
2. Planificación del proyecto	15
2.1. Metodología	15
2.2. Planificación temporal del proyecto	15
2.3. Seguimiento del proyecto	16
2.4. Costes	18
2.4.1. Recursos software	18
2.4.2. Recursos hardware	19
2.4.3. Recursos humanos	20

2.4.4. Resumen costes	20
2.5. Riesgos	21
3. Análisis del sistema	23
3.1. Definición de requisitos	23
3.1.1. Diagrama de casos de uso	23
3.1.2. Requisitos de datos	32
3.2. Análisis de requisitos	34
3.2.1. Diagrama de clases	34
3.2.2. Diagramas de actividad	35
4. Diseño del sistema	37
4.1. Diseño de la arquitectura del sistema/producto	37
4.2. Diseño de la base de datos	37
4.3. Diseño de las interfaces	39
5. Implementación y pruebas	43
5.1. Estructura del código	43
5.2. Descripción técnica de la implementación	45
5.3. Pruebas	49
6. Conclusiones	51

Índice de figuras

2.1. Diagrama de la estructura de descomposición del trabajo	16
2.2. Representación visual del Diagrama de Gantt sobre la planificación	17
2.3. Representación visual del Diagrama de Gantt sobre el seguimiento del proyecto	18
3.1. Diagrama de casos de uso del sistema	24
3.2. Diagrama de clases del sistema	35
3.3. Diagrama de actividades del caso de uso CU_05 Gestionar prefacturas	36
4.1. Representación visual de la conexión de los sistemas informáticos de BC[5]	38
4.2. Diseño físico de la base de datos	39
4.3. Menú de navegación del producto final	39
4.4. Página de lista de contratos del producto final	40
4.5. Página de ficha de contrato del producto final	40
4.6. Página de ficha de contrato del producto final	41
5.1. Estructura de ficheros y carpetas del proyecto	44
5.2. Código de la tabla Contract	46
5.3. Código de la página ContractCard	47
5.4. Código de un campo de la página ContractCard	47
5.5. Estructura de los archivos necesarios para implementar el minimapa	49
5.6. Ejecución de las pruebas en Github Actions	50

Índice de tablas

2.1. Resumen total costes software	19
2.2. Resumen total costes hardware	20
2.3. Coste de recursos humanos	20
2.4. Resumen total costes	21
3.1. Actores del sistema	24
3.2. Documentación del caso de uso CU_01	25
3.3. Documentación del caso de uso CU_02	26
3.4. Documentación del caso de uso CU_03	27
3.5. Documentación del caso de uso CU_04	28
3.6. Documentación del caso de uso CU_05	29
3.7. Documentación del caso de uso CU_06	30
3.8. Documentación del caso de uso CU_07	30
3.9. Documentación del caso de uso CU_08	30
3.10. Documentación del caso de uso CU_09	31
3.11. Documentación del caso de uso CU_10	31
3.12. Documentación del caso de uso CU_11	31
3.13. Documentación del requisito de datos RD01	32
3.14. Documentación del requisito de datos RD02	33

3.15. Documentación del requisito de datos RD03	33
3.16. Documentación del requisito de datos RD04	34
3.17. Documentación del requisito de datos RD05	34

Capítulo 1

Introducción

1.1. Contexto y motivación del proyecto

En este documento se presenta el proyecto que se ha desarrollado en Innova Advanced Consulting¹, una empresa especializada en consultoría y soluciones tecnológicas ubicada en Castellón de la Plana. La empresa cuenta con una amplia plantilla y con más de 19 años de experiencia en el sector. Esta cuenta con dos departamentos de desarrollo. Uno se centra en el desarrollo de módulos para el ERP (Enterprise Resource Planning) Dynamics 365 Business Central (D365 BC)² y el otro se centra en el desarrollo del ERP Dynamics 365 AX (Axapta)³. La mayor diferencia entre ambos ERPs reside en el tamaño de su target. BC se centra en pequeñas y medianas empresas, mientras que AX se centra en empresas de más volumen. Principalmente, los servicios que ofrece Innova son contratados a través de partners, que son quienes están en contacto con el cliente final (aquellos que utilizarán el módulo desarrollado).

En el presente proyecto se va a generar un módulo para el ERP BC de Microsoft, que gestione el alquiler y mantenimiento de maquinaria. El sistema a desarrollar deberá poder gestionar el alquiler de dichas máquinas, su reparación en caso de avería, así como la facturación de los contratos que se realicen con los clientes. Además, también se deberá poder gestionar la ubicación de las máquinas en tiempo real.

1.2. Objetivo y alcance del proyecto

El principal objetivo de este proyecto es desarrollar un módulo, adaptándolo a las necesidades de un cliente, “ElectroTech”, en este caso para la gestión de mantenimientos preventivos y correctivos de maquinaria. Además, también se quiere crear una documentación técnica del software para los futuros usuarios del sistema.

¹<https://www.innovaconsulting.es>

²<https://dynamics.microsoft.com/es-es/business-central/overview/>

³<https://dynamics.microsoft.com/es-es/ax/>

El proyecto comenzará con un estudio del arte de los ERPs y de BC. Tras este, se realizará el desarrollo para la gestión de los contratos con clientes, partes, facturación y ubicación. Finalmente, el proyecto concluirá con la documentación del sistema.

Por otro lado, no se incluirá la gestión de los datos de las máquinas, su compra o stock, o la de los clientes de la empresa. Tampoco abarca el sistema de autenticación y gestión de los usuarios del sistema.

1.3. Objetivo y alcance del producto

El módulo que se ha desarrollado debe ser capaz de llevar la gestión de los datos de los diferentes contratos y partes que se puedan generar a lo largo de estos. Es por esto que también debe incluir un sistema de partes, en caso de que se necesite cobertura adicional. Sin embargo, no incluye aspectos como la producción de las máquinas, o el marketing de la empresa.

1.3.1. Alcance funcional

Entre las principales funciones que el sistema deberá ser capaz de llevar a cabo se encuentran:

- Generar contratos de alquiler y de mantenimiento.
- Crear una planificación de facturación para los contratos de mantenimiento.
- Generar partes (pueden ser de mantenimiento, avería, etc.) a partir de las máquinas estipuladas en los contratos.
- Generar prefacturas (no definitivas) a partir de los contratos de alquiler, planificación de facturación (contratos de mantenimiento), y albaranes generados a partir de los partes (cada apartado se factura de manera diferente) y se agruparán dependiendo del cliente (no agrupar, agrupar por contrato, agrupar todo).
- Registrar las pre-facturas para generar las facturas finales.
- Ver la ubicación actual de cada máquina.

Los contratos de mantenimiento generarán una planificación de facturación, que es la que se incluirá en las prefacturas. Por otro lado, los de alquiler se incluirán, junto con los posibles partes derivados de estos, directamente en las prefacturas. Dichos documentos previos serán los que, una vez revisados, pasarán a formar la factura final.

1.3.2. Alcance organizacional

El alcance organizacional abarcará únicamente a los trabajadores de “ElectroTech”, que serán los encargados de realizar todas las gestiones necesarias.

1.3.3. Alcance informático

El sistema desarrollado se integrará en BC para parecer parte del ERP en su estado original. A pesar de esto, el nuevo módulo accederá a módulos ya existentes:

- Clientes: a pesar de modificar un pequeño aspecto sobre este módulo, el sistema accederá para utilizar los datos de los clientes existentes.
- Productos: para poder crear contratos sobre las máquinas, el sistema debe acceder a los datos existentes de las máquinas de la empresa.
- Facturas: como se explicará más adelante, a pesar de haberse creado el proceso de facturación, este hará uso del sistema de facturas existente para poder ajustarse más uniformemente a los procesos de contabilidad.

Además de esto, el sistema se conectará con un servicio remoto para mostrar en un mapa la ubicación de las máquinas.

1.4. Descripción del proyecto

En la actualidad, BC es una solución ERP ampliamente utilizada por las organizaciones para gestionar sus procesos empresariales. BC proporciona funcionalidades integrales para áreas como finanzas, ventas, compras, inventario y más, lo que lo convierte en una herramienta versátil y eficiente para la gestión empresarial.

Sin embargo, a pesar de las numerosas funcionalidades que ofrece BC, no cuenta con un módulo de gestión de maquinaria que se ajuste a las necesidades de todos los clientes. Por esto, se debe suplir vacío funcional que puede ser una limitación para empresas que operan en industrias relacionadas con el alquiler y mantenimiento de maquinaria, como “ElectroTech”.

Durante el desarrollo del proyecto, se buscará una integración efectiva con los módulos y funcionalidades existentes de BC. Esto implica aprovechar las capacidades de gestión financiera, de inventario, de ventas y otras áreas relevantes de BC para garantizar una operación fluida y coherente en toda la aplicación.

A medida que se desarrolle el módulo, se llevarán a cabo pruebas exhaustivas para garantizar la calidad y la funcionalidad correcta de las nuevas características. Esto implicará la creación de casos de prueba y la realización de pruebas de integración para asegurar que el módulo cumple con los requisitos establecidos.

Este proyecto será implementado por un analista de datos, quien se encargará de ponerse en contacto con el cliente, estudiar su caso y sus necesidades, y diseñar una solución correspondiente. Por otro lado, también contará con un desarrollador de software, quien se dedicará a integrar la solución anteriormente mencionada en BC, y de crear un manual de usuario.

1.4.1. Tecnologías

Herramientas

Dynamics 365 Business Central El ERP utilizado para el proyecto será Dynamics 365 Business Central, el ERP de Microsoft. Es una solución de gestión empresarial diseñada para pequeñas y medianas empresas. Proporciona una visión completa de los procesos de negocio, incluyendo finanzas, ventas, compras, inventario y producción. Se basa en una arquitectura cliente-servidor, donde el servidor alberga la base de datos centralizada y el motor de procesamiento, mientras que los clientes se conectan al servidor para acceder y operar en la aplicación. Esta arquitectura permite una gestión centralizada de los datos y facilita el acceso concurrente de múltiples usuarios desde diferentes ubicaciones.

Visual Studio Code⁴: es un editor de código fuente ampliamente utilizado que proporciona un entorno de desarrollo integrado (IDE) para el desarrollo de extensiones de BC utilizando el lenguaje AL. Ofrece una variedad de herramientas y extensiones para facilitar el desarrollo, la depuración y la administración de proyectos de BC. Es altamente configurable y admite características como el resaltado de sintaxis, la finalización automática de código y la integración con sistemas de control de versiones.

GitHub: para el control de versiones se utilizará GitHub⁵. Es una plataforma web de alojamiento y colaboración para proyectos de desarrollo de software. Permite a los desarrolladores compartir y trabajar en código fuente, realizar un seguimiento de cambios, colaborar en equipo, realizar revisiones de código y gestionar versiones de proyectos de manera eficiente. Además, para trabajar con GitHub de una manera más intuitiva y visual, se utilizará GitHub Desktop, una aplicación de escritorio que permite clonar, crear, confirmar y sincronizar repositorios, así como administrar ramas y fusiones, todo sin tener que utilizar la línea de comandos. También se utilizará GitHub Actions⁶, que es un servicio de integración y entrega continua proporcionado por GitHub. Permite automatizar y gestionar el flujo de trabajo de desarrollo de software, desde la construcción y prueba hasta la implementación y entrega de aplicaciones. En este caso, se utilizará para probar constantemente que el código pase correctamente una batería de pruebas.

API Google Maps: una herramienta que proporciona Google para integrar mapas interactivos en aplicaciones web y móviles. Permite mostrar mapas, ubicaciones, direcciones, rutas y otra información geográfica en tus aplicaciones.

Lenguajes

Application Language (AL) BC utiliza el lenguaje de programación AL para la personalización y el desarrollo de extensiones. AL es un lenguaje declarativo y orientado a eventos que permite la creación de objetos, páginas, informes y otros elementos personalizados dentro de BC. Además, se utilizarán herramientas como Visual Studio Code para el desarrollo y la depuración de las extensiones.

⁴<https://code.visualstudio.com>

⁵<https://github.com>

⁶<https://github.com/features/actions>

JavaScript (JS) Es un lenguaje de programación de alto nivel, interpretado y orientado a objetos. Es ampliamente utilizado para desarrollar aplicaciones web interactivas y dinámicas. JS se ejecuta en el lado del cliente, es decir, en el navegador web del usuario, lo que permite agregar interactividad y funcionalidad a las páginas web[2]. Este se utilizará para implementar conectarse con la API de Google Maps e implementarla en el lenguaje AL.

1.5. Estructura de la memoria

El resto de la memoria se organizará de la siguiente manera:

- **Planificación del proyecto:** primero, se especificará la planificación previa que se ha seguido para el desarrollo del proyecto. Entonces, se explicará cómo ha ido avanzando, y si se ha mantenido esta planificación inicial, así como los costes que ha supuesto este desarrollo para la empresa.
- **Análisis del producto:** en este apartado se indicarán las necesidades del sistema. Esto incluirá la especificación de los requisitos con los que deberá contar el producto final, así como los casos de uso de los posibles usuarios finales necesarios para cumplir con estos requisitos.
- **Diseño del producto:** a continuación se explicará el diseño que se ha utilizado para desarrollar los diferentes apartados. En estos se incluye cómo se relacionarán todos los componentes del sistema, y el diseño de las estructuras de datos, y las interfaces que las relacionarán.
- **Implementación y pruebas:** en este apartado se incluirá, de manera general, el número de elementos desarrollados por código, así como otros apartados referentes al desarrollo más “práctico” del proyecto. Además, también se especificarán las pruebas que se han creado, y cómo se mantiene la integridad del código y del sistema.
- **Conclusiones:** por último, se reflexionará acerca de cuál era el proyecto planteado, y las diferencias con el producto final. Además, también se incluirá en esta reflexión los conocimientos adquiridos tanto a nivel profesional como personal.

Capítulo 2

Planificación del proyecto

2.1. Metodología

Debido a que este proyecto contará con una especificación inicial que se estima constante durante todo el proceso de desarrollo, se ha decidido utilizar una metodología predictiva. Esta metodología es especialmente útil en entornos estables en los que se mantiene la definición a lo largo del proyecto[1].

Esta consta de varias fases:

- Inicio del proyecto: se creará un primer esbozo de lo que será el producto final.
- Planificación: se estimará una planificación temporal para el proyecto.
- Análisis de requisitos: Se especificarán las funcionalidades finales del producto.
- Diseño: se diseñará toda la estructura a implementar.
- Desarrollo y despliegue: se implementará toda la solución y se entregará al cliente final.

2.2. Planificación temporal del proyecto

Para organizar las tareas necesarias para desarrollar el proyecto, se ha confeccionado el diagrama de la Figura 2.1, que incluye las fases anteriormente mencionadas.

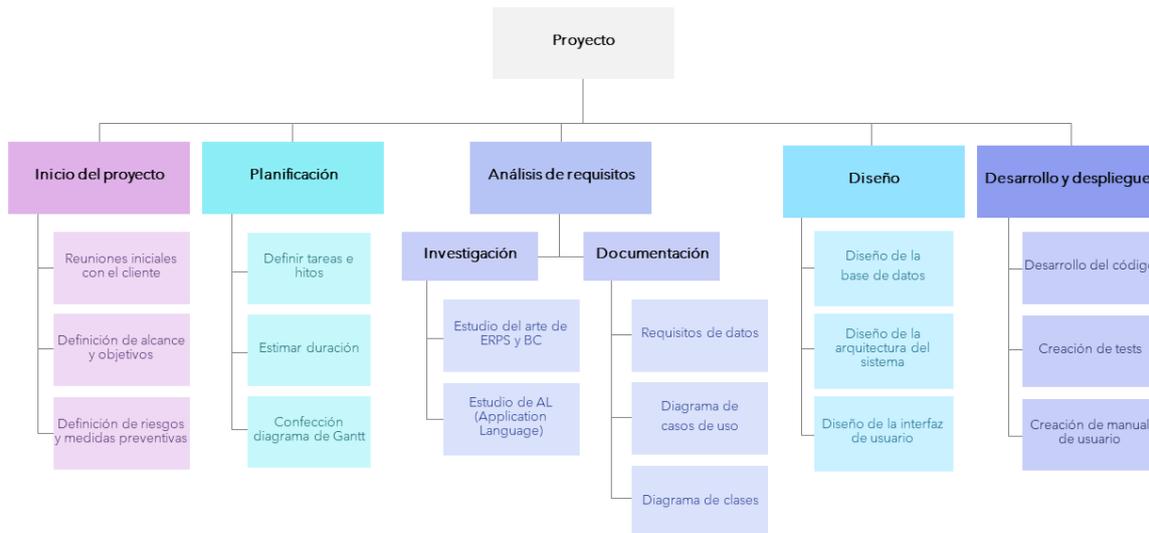


Figura 2.1: Diagrama de la estructura de descomposición del trabajo

Para la estimación temporal del proyecto, se han obviado las dos primeras fases del proyecto, y simplemente se han incluido las del análisis, diseño y desarrollo. Esta distribución temporal se puede ver de manera esquematizada en la Figura 2.2. A pesar de haber sido planteadas como dos fases diferentes, el diseño y el desarrollo se han solapado de cierta manera, ya que cada apartado se diseñaba y desarrollaba de manera seguida. En este diagrama, podemos ver la fecha inicial del proyecto, el 06/03/2023 y la fecha estimada de finalización, el 23/05/23. Esta estimación se ha realizado teniendo en cuenta que el proyecto durará un total de 300 horas y que se dedicarán 6 horas diarias al proyecto, 5 días a la semana a excepción de durante mayo, mes en el que la empresa implementará el formato de semana laboral de 4 días.

2.3. Seguimiento del proyecto

A pesar de que a mitad de proyecto pueden surgir un gran número de imprevistos que pueden acabar cambiando el proceso de trabajo, el ritmo de este se ha mantenido bastante acorde al planificado. Como se puede ver en la Figura 2.3, la estructura de trabajo ha sido prácticamente la planificada. Las únicas diferencias significativas son la inclusión de un apartado extra, en el que se desarrolló un apartado para poder acceder a todas las funcionalidades de manera más simple, y la de un histórico para poder diferenciar entre los procesos en curso y los ya finalizados. Sin embargo, no se ha podido realizar un manual de usuario como se había previsto en un inicio. A pesar de estas modificaciones, el tiempo final dedicado al módulo no ha variado. Además, también se puede observar el porcentaje de completación, que indica el progreso de cada tarea.

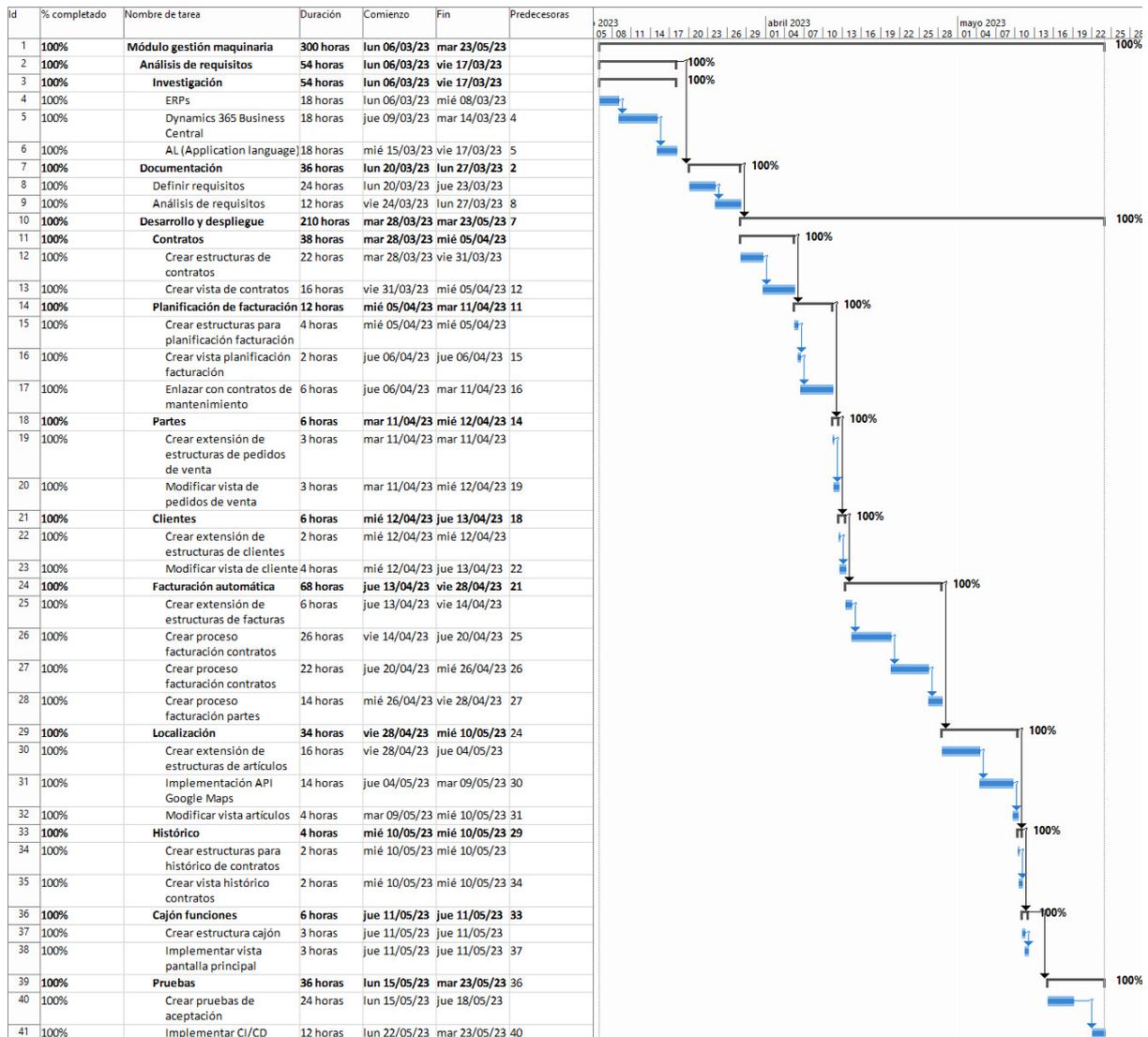


Figura 2.3: Representación visual del Diagrama de Gantt sobre el seguimiento del proyecto

2.4. Costes

Para este proyecto se realizó una estimación relativa al coste que supondría para la empresa. A continuación se muestra un desglose sobre este dividido en tres categorías.

2.4.1. Recursos software

Para este proyecto solo se ha tenido en cuenta el coste del desarrollo, por lo que únicamente se ha incluido el coste de aquellos programas utilizados para ello. En este caso tenemos:

- Dynamics 365 Business Central: Para poder desarrollar utilizando toda la funcionalidad de este ERP, será necesario contar con una licencia activa. Estas tienen un coste de 65,50€ al mes.
- Visual Studio Code: puesto que se trata de software de código abierto, se utilizará la versión gratuita.
- GitHub: se utilizará la versión de GitHub Teams, ya que es la que más se adapta a las necesidades del producto. Este tiene un coste de 3,65€ al mes.
- API Google Maps: este servicio ofrece un crédito de 182,47€ mensuales para su uso. Debido a que se estima que no se realizará un gran número de llamadas a este servicio, y que no se superará el crédito mencionado, no será necesario pagar nada.

Como se puede ver en la Tabla 2.1, debido a que se estima que el proyecto durará un total de 3 meses, se ha calculado el coste total en este periodo de tiempo. Por lo tanto, el precio aumenta a 207,45€.

Software	Precio/Mes (€)	Precio/3 Meses (€)
D365 BC	65,50	196,50
VSC	0	0
GitHub	3,65	10,95
API Google Maps	0	0
		207,45

Tabla 2.1: Resumen total costes software

2.4.2. Recursos hardware

Este proyecto se ha desarrollado en una estación de trabajo que se compone de:

- Ordenador portátil: tiene un coste de 500€ y una vida útil de aproximadamente 4 años.
- Monitor extra: tiene un coste de 80€ y una vida útil de aproximadamente 6,5 años.
- Teclado: tiene un coste de 10€ y una vida útil de aproximadamente 3,5 años.
- Ratón: tiene un coste de 5€ y una vida útil de aproximadamente 3,5 años.

Para calcular el precio que supondrá el uso de estos elementos, debemos adaptarlo al tiempo de desarrollo del proyecto, es decir, 3 meses. Por esto, como se ve en la Tabla 2.2 podemos afirmar que el coste del hardware asciende a **35,40€**.

Software	Precio (€)	Vida útil (años)	Total 3 meses (€)
PC	500	4	31,25
Monitor	80	6,5	3,08
Teclado	10	3,5	0,71
Ratón	5	3,5	0,36
			35,40

Tabla 2.2: Resumen total costes hardware

2.4.3. Recursos humanos

Por último, para el punto clave del proyecto, se ha contado con los siguientes roles:

- Analista de datos: es la persona que se ha encargado de investigar las necesidades del cliente y de diseñar la solución para solventarlas. Como hemos podido ver anteriormente, ha invertido un total de 90 horas.
- Desarrollador de software: por otra parte, también ha sido necesaria una persona dedicada a implementar dicha solución. Como hemos podido ver anteriormente, ha invertido un total de 210 horas.

Para calcular el coste asociado a estos recursos humanos, se tomaron en cuenta los salarios medios correspondientes a los perfiles mencionados en el ámbito del ERP BC[6][7]. Utilizando esta información, se ha podido determinar el coste total de los recursos humanos dedicados al proyecto, como podemos ver en la Tabla 2.3.

Rol	Precio por hora (€)	Total de horas	Total (€)
Analista de datos	17,95	90	1.615,50
Desarrollador de software	16,92	210	3.553,20
			5.168,70

Tabla 2.3: Coste de recursos humanos

Además del coste directo de los recursos humanos, es importante considerar los costes adicionales asociados con la contratación y los gastos indirectos. Estos costes suelen variar entre un 20 % y un 30 % para la contratación y un 20 % para los gastos indirectos. Por lo tanto, tenemos que el coste total de los recursos humanos es:

$$\text{Coste total RRHH} = 5,168,70 + 25\% + 20\% = 7.494,62\text{€}$$

2.4.4. Resumen costes

Teniendo en cuenta los tres apartados anteriormente mencionados, podemos concluir que el proyecto tendrá un coste aproximado de 7.737,47€, como se puede ver en la Tabla 2.4

Recursos	Subtotal (€)
Software	207,45
Hardware	35,40
Humanos	7.494,62
	7.737,47

Tabla 2.4: Resumen total costes

2.5. Riesgos

Como posibles riesgos para la elaboración de este proyecto se tienen los siguientes:

- R01 - Falta de experiencia: la poca experiencia en el mundo de los ERPs, al igual que con el lenguaje y entorno a utilizar, puede afectar al desempeño. Importancia alta.
- R02 - Falta de tiempo: el hecho de que se vaya a compaginar este desarrollo con otros proyectos, puede causar que la calidad del producto final no sea suficiente. Importancia media.
- R03 - Estimación optimista: puede que la estimación temporal para el proyecto haya sido demasiado optimista y acabe faltando tiempo de desarrollo. Importancia media.

Para evitar que estos inconvenientes acaben afectando negativamente al resultado final, se tomarán las siguientes medidas:

- R01: Se reservará una parte suficiente del tiempo del proyecto a investigar y adquirir experiencia en el mundo de los ERPs, así como en el lenguaje y entorno a utilizar.
- R02: Se realizará una planificación detallada y realista, teniendo en cuenta la carga de trabajo actual y los recursos disponibles. Se establecerán plazos realistas para evitar la falta de tiempo.
- R03: Se llevará a cabo una revisión de la estimación temporal inicial y se realizará un seguimiento regular del progreso del proyecto. Si se identifica una falta de tiempo, se evaluará ajustar la planificación, o reevaluar los requisitos y prioridades.

Capítulo 3

Análisis del sistema

3.1. Definición de requisitos

Para poder identificar las necesidades del sistema, se ha llevado a cabo un estudio de los requisitos funcionales. Estos requisitos se han obtenido a partir de la especificación de los objetivos del sistema, así como de las funcionalidades que se le ofrecerán a los trabajadores. En esta sección se describirá la definición de estos requisitos.

3.1.1. Diagrama de casos de uso

El diagrama de casos de uso es una herramienta que nos permite identificar los actores del sistema, así como las funcionalidades que se les ofrecerán. En la figura 3.1 se muestra el diagrama de casos de uso del sistema.

Actores

Los actores del sistema son las entidades externas que interactúan con el mismo. En este caso, el sistema tendrá un único actor principal, el trabajador, y un actor secundario, el servicio de Google Maps. Si nos fijamos en la Tabla 3.1, podemos ver un resumen de los actores del sistema, así como una breve descripción de los mismos.

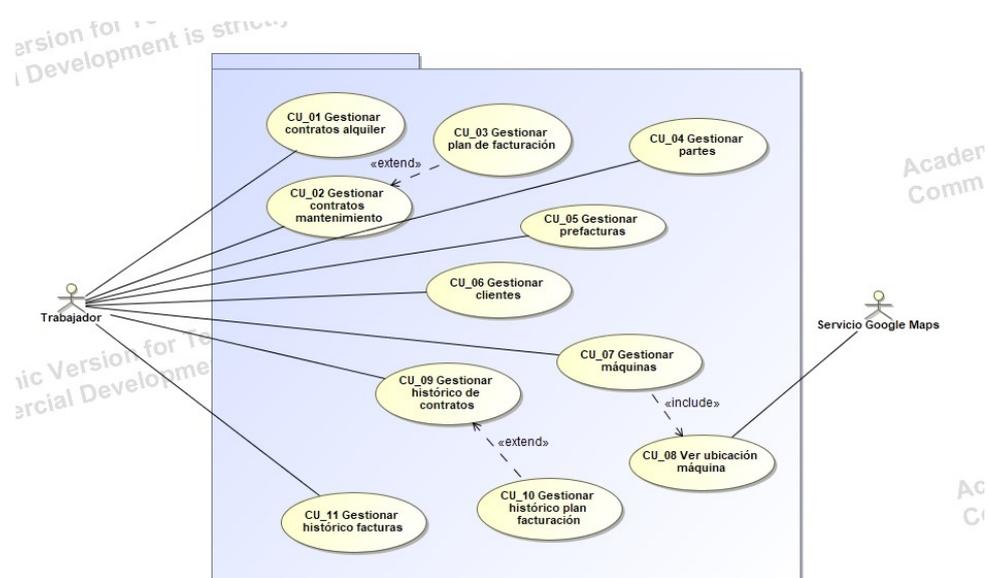


Figura 3.1: Diagrama de casos de uso del sistema

Identificador	Actor	Descripción
AP01	Trabajador	El trabajador es el actor principal del sistema. Tendrá acceso a toda la funcionalidad del sistema, y será el encargado de gestionar los contratos, partes, facturas, etc.
AS01	Google Maps	Google Maps es un servicio externo que se utilizará para obtener la localización de las máquinas basándose en el cliente que las tenga en uso.

Tabla 3.1: Actores del sistema

Especificación casos de uso

A continuación, se detallan los casos de uso del sistema, así como una breve descripción de los mismos.

Identificador	CU_01
Nombre:	Gestionar contratos alquiler
Actor principal:	Trabajador
Descripción:	Este caso de uso representa la funcionalidad que permite al trabajador crear nuevos contratos de alquiler, modificar sus datos, o eliminarlos.
Secuencia normal:	<p>Se desea crear un contrato de alquiler.</p> <ol style="list-style-type: none"> 1. El trabajador accede a la lista de contratos seleccionando la opción “Contratos máquinas”. 2. El trabajador selecciona la opción de “nuevo”. 3. El trabajador rellena todos los datos del contrato (cliente, cliente de facturación, tipo, transportista, día de facturación y periodo de facturación). 4. El trabajador añade las líneas de contrato y rellena sus datos (máquina, fechas de entrega prevista, tipo de facturación y descuento). 5. El sistema asigna un identificador al contrato y almacena los datos introducidos en la base de datos.
Secuencia normal:	<p>Se desea modificar un contrato.</p> <ol style="list-style-type: none"> 1. El trabajador accede a la lista de contratos seleccionando la opción “Contratos máquinas”. 2. El trabajador selecciona el contrato que desea modificar. 3. El trabajador modifica los datos del contrato. 4. El sistema almacena los datos introducidos en la base de datos.
Secuencia normal:	<p>Se desea eliminar un contrato.</p> <ol style="list-style-type: none"> 1. El trabajador accede a la lista de contratos seleccionando la opción “Contratos máquinas”. 2. El trabajador selecciona el contrato que desea eliminar. 3. El trabajador selecciona la opción de “eliminar”. 4. El sistema elimina el contrato de la base de datos.
Excepciones:	<ol style="list-style-type: none"> 1. El trabajador intenta añadir una máquina que ya está en uso. 2. El trabajador intenta añadir fechas no válidas (fecha de fin entrega anterior a recogida, etc.). 3. El trabajador intenta añadir un contrato con datos no válidos (cliente no existente, etc.).

Tabla 3.2: Documentación del caso de uso CU_01

Identificador	CU_02
Nombre:	Gestionar contratos mantenimiento
Actor principal:	Trabajador
Descripción:	Este caso de uso representa la funcionalidad que permite al trabajador crear nuevos contratos de mantenimiento, modificar sus datos, o eliminarlos.
Secuencia normal:	<p>Se desea crear un contrato de mantenimiento.</p> <ol style="list-style-type: none"> 1. El trabajador accede a la lista de contratos desde el menú principal. 2. El trabajador selecciona la opción de “Nuevo”. 3. El trabajador rellena todos los datos del contrato (cliente, tipo, facturación...). 4. El trabajador añade las máquinas que se incluyen en el contrato y rellena sus datos. 5. El sistema almacena los datos introducidos en la base de datos.
Secuencia alternativa:	<p>Se desea modificar un contrato.</p> <ol style="list-style-type: none"> 1. El trabajador accede a la lista de contratos desde el menú principal. 2. El trabajador selecciona el contrato que desea modificar. 3. El trabajador modifica los datos del contrato. 4. El sistema almacena los datos introducidos en la base de datos.
Secuencia alternativa:	<p>Se desea eliminar un contrato.</p> <ol style="list-style-type: none"> 1. El trabajador accede a la lista de contratos desde el menú principal. 2. El trabajador selecciona el contrato que desea eliminar. 3. El trabajador selecciona la opción de “Eliminar”. 4. El sistema elimina el contrato de la base de datos.
Excepciones:	<ol style="list-style-type: none"> 1. El trabajador intenta añadir una máquina que ya está en uso. 2. El trabajador intenta añadir fechas no válidas (fecha de fin entrega anterior a recogida, etc.). 3. El trabajador intenta añadir un contrato con datos no válidos (cliente no existente, etc.).

Tabla 3.3: Documentación del caso de uso CU_02

Identificador	CU_03
Nombre:	Gestionar plan de facturación
Actor principal:	Trabajador
Descripción:	Este caso de uso representa la funcionalidad que permite al trabajador crear o ver la planificación de facturación para los contratos de mantenimiento y marcarla como facturable.
Secuencia normal:	<p>El trabajador desea crear la planificación de facturación para un contrato de mantenimiento.</p> <ol style="list-style-type: none"> 1. El trabajador accede a la lista de contratos desde el menú principal. 2. El trabajador selecciona el contrato de mantenimiento para el que desea crear la planificación. 3. El trabajador selecciona la opción de “Crear planificación de facturación”. 4. El sistema crea la planificación de facturación para el contrato de mantenimiento y la almacena en la base de datos.
Secuencia alternativa:	<p>El trabajador desea ver la planificación de facturación para un contrato de mantenimiento.</p> <ol style="list-style-type: none"> 1. El trabajador accede a la lista de contratos desde el menú principal. 2. El trabajador selecciona el contrato de mantenimiento para el que desea ver la planificación. 3. El trabajador selecciona la opción de “Ver planificación de facturación”.
Secuencia alternativa:	<p>El trabajador desea marcar la planificación de facturación para un contrato de mantenimiento como facturable.</p> <ol style="list-style-type: none"> 1. Partiendo desde la secuencia anterior, el trabajador selecciona el plan de facturación que desea marcar como facturable. 2. El trabajador marca el campo “Facturable”. 3. El sistema almacena los datos modificados en la base de datos.
Excepciones:	<ol style="list-style-type: none"> 1. El trabajador intenta crear la planificación para un contrato de alquiler. 2. El trabajador intenta crear la planificación para un contrato de mantenimiento que ya tiene una planificación. 3. El trabajador intenta crear la planificación para un contrato de mantenimiento que no está marcado como facturable. 4. El trabajador intenta crear la planificación para un contrato de mantenimiento que no tiene máquinas asociadas facturables. 5. El trabajador intenta crear la planificación para un contrato de mantenimiento que no tiene máquinas asociadas facturables en el periodo de facturación.

Tabla 3.4: Documentación del caso de uso CU_03

Identificador	CU_04
Nombre:	Gestionar partes
Actor principal:	Trabajador
Descripción:	Este caso de uso representa la funcionalidad que permite al trabajador crear nuevos partes a partir de los contratos de alquiler, modificar sus datos, eliminarlos y registrarlos.
Secuencia normal:	Se desea crear un parte. 1. El trabajador accede a la lista de partes desde el menú principal. 2. El trabajador selecciona la opción de “Nuevo”. 3. El trabajador rellena todos los datos del parte (contrato asociado, tipo, recursos...). 4. El sistema almacena los datos introducidos en la base de datos.
Secuencia alternativa:	Se desea modificar un parte no registrado. 1. El trabajador accede a la lista de partes desde el menú principal. 2. El trabajador selecciona el parte que desea modificar. 3. El trabajador modifica los datos del parte. 4. El sistema almacena los datos introducidos en la base de datos.
Secuencia alternativa:	Se desea eliminar un parte no registrado. 1. El trabajador accede a la lista de partes desde el menú principal. 2. El trabajador selecciona el parte que desea eliminar. 3. El trabajador selecciona la opción de “Eliminar”. 4. El sistema elimina el parte de la base de datos.
Secuencia alternativa:	Se desea registrar un parte. 1. El trabajador accede a la lista de partes desde el menú principal. 2. El trabajador selecciona el parte que desea registrar. 3. El trabajador selecciona la opción de “Registrar”. 4. El sistema registra el parte en la base de datos.
Excepciones:	1. El trabajador intenta crear un parte con datos no válidos (contrato no existente, etc.). 2. El trabajador intenta modificar un parte con datos no válidos (contrato no existente, etc.). 3. El trabajador intenta modificar un parte que ya está registrado. 4. El trabajador intenta eliminar un parte que ya está registrado. 5. El trabajador intenta registrar un parte que ya está registrado.

Tabla 3.5: Documentación del caso de uso CU_04

Identificador	CU_05
Nombre:	Gestionar prefacturas
Actor principal:	Trabajador
Descripción:	Este caso de uso representa la funcionalidad que permite al trabajador crear nuevas prefacturas a partir de los contratos de mantenimiento, de alquiler, y de partes, modificar sus datos, eliminarlas o registrarlas.
Secuencia normal:	<p>Se desea crear una prefactura.</p> <ol style="list-style-type: none"> 1. El trabajador accede a la opción de “Facturación automática” desde el menú principal. 2. El trabajador incluye los filtros para la prefactura (tipos de contrato, partes, fechas, etc.). 3. El trabajador selecciona la opción de “Crear prefactura”. 4. El sistema crea la o las prefacturas correspondientes, dependiendo del tipo de emisión de facturas de los clientes, y las almacena en la base de datos.
Secuencia alternativa:	<p>Se desea modificar una prefactura no registrada.</p> <ol style="list-style-type: none"> 1. El trabajador accede a la lista de prefacturas desde el menú principal. 2. El trabajador selecciona la prefactura que desea modificar. 3. El trabajador modifica los datos de la prefactura (cantidad de elementos a facturar). 4. El sistema almacena los datos introducidos en la base de datos.
Secuencia alternativa:	<p>Se desea eliminar una prefactura no registrada.</p> <ol style="list-style-type: none"> 1. El trabajador accede a la lista de prefacturas desde el menú principal. 2. El trabajador selecciona la prefactura que desea eliminar. 3. El trabajador selecciona la opción de “Eliminar”. 4. El sistema elimina la prefactura de la base de datos.
Secuencia alternativa:	<p>Se desea registrar una prefactura.</p> <ol style="list-style-type: none"> 1. El trabajador accede a la lista de prefacturas desde el menú principal. 2. El trabajador selecciona la prefactura que desea registrar. 3. El trabajador selecciona la opción de “Registrar”. 4. El sistema registra la prefactura en la base de datos y lleva a cabo la facturación de los contratos y partes asociados.
Excepciones:	<ol style="list-style-type: none"> 1. El trabajador intenta crear una prefactura con datos no válidos (no se crea ninguna prefactura). 2. El trabajador intenta modificar una prefactura con datos no válidos. 3. El trabajador intenta registrar una prefactura con datos no válidos.

Tabla 3.6: Documentación del caso de uso CU_05

Identificador	CU_06
Nombre:	Gestionar clientes
Actor principal:	Trabajador
Descripción:	Este caso de uso representa la funcionalidad que permite al trabajador ver los datos de los clientes y modificarlos.
Secuencia normal:	Se desea modificar la emisión de facturas de un cliente. 1. El trabajador accede a la lista de clientes desde el menú principal. 2. El trabajador selecciona el cliente que desea modificar. 3. El trabajador modifica el campo de “Emisión de facturas”. 4. El sistema almacena los datos introducidos en la base de datos.

Tabla 3.7: Documentación del caso de uso CU_06

Identificador	CU_07
Nombre:	Gestionar máquinas
Actor principal:	Trabajador
Descripción:	Este caso de uso representa la funcionalidad que permite al trabajador ver el listado de máquinas y sus datos.
Secuencia normal:	Se desea ver los datos de una máquina. 1. El trabajador accede a la lista de máquinas desde el menú principal. 2. El sistema muestra el listado de máquinas. 3. El trabajador selecciona la máquina que desea ver. 4. El sistema muestra los datos de la máquina.

Tabla 3.8: Documentación del caso de uso CU_07

Identificador	CU_08
Nombre:	Ver ubicación máquina
Actor principal:	Trabajador
Descripción:	Este caso de uso representa la funcionalidad que permite al trabajador ver la ubicación de una máquina.
Secuencia normal:	Se desea ver la ubicación de una máquina en uso. 1. El trabajador accede a la lista de máquinas desde el menú principal. 2. El trabajador selecciona la máquina que desea ver. 3. El sistema muestra la ubicación de la máquina.
Excepciones:	1. El trabajador intenta ver la ubicación de una máquina que no está en uso. 2. El trabajador intenta ver la ubicación de una máquina cuyo arrendatario no tiene dirección registrada. 3. Ocurre un error con la API de donde se obtienen los datos de la ubicación (Google Maps API).

Tabla 3.9: Documentación del caso de uso CU_08

Identificador	CU_09
Nombre:	Gestionar histórico de contratos
Actor principal:	Trabajador
Descripción:	Este caso de uso representa la funcionalidad que permite al trabajador ver el histórico de contratos.
Secuencia normal:	Se desea ver el histórico de contratos. 1. El trabajador accede a la lista de histórico de contratos desde el menú principal. 2. El sistema muestra el listado de histórico de contratos.

Tabla 3.10: Documentación del caso de uso CU_09

Identificador	CU_10
Nombre:	Gestionar histórico plan facturación
Actor principal:	Trabajador
Descripción:	Este caso de uso representa la funcionalidad que permite al trabajador ver el histórico de planes de facturación de un contrato.
Secuencia normal:	Se desea ver el histórico de planes de facturación de un contrato de mantenimiento. 1. El trabajador accede a la lista de histórico de contratos desde el menú principal. 2. El trabajador selecciona el contrato del que desea ver el histórico de planes de facturación. 3. El trabajador selecciona la opción “Ver histórico de planes de facturación”. 4. El sistema muestra el listado de histórico de planes de facturación.
Excepciones:	1. El trabajador intenta ver el histórico de planes de facturación de un contrato que no tiene planes de facturación.

Tabla 3.11: Documentación del caso de uso CU_10

Identificador	CU_11
Nombre:	Gestionar histórico facturas
Actor principal:	Trabajador
Descripción:	Este caso de uso representa la funcionalidad que permite al trabajador ver el histórico de facturas.
Secuencia normal:	Se desea ver el histórico de facturas. 1. El trabajador accede a la lista de histórico de facturas desde el menú principal. 2. El sistema muestra el listado de histórico de facturas.

Tabla 3.12: Documentación del caso de uso CU_11

3.1.2. Requisitos de datos

En esta sección se detallan los requisitos de datos que trabajador deberá introducir en el sistema para poder realizar las tareas descritas en los casos de uso anteriores. Debido a la naturaleza de la base de datos, y a las necesidades del módulo, los requisitos de datos se dividen en dos grupos:

- Nuevo: requisitos de datos que son completamente nuevos para el sistema.
- Extensión: requisitos de datos que ya existen en el sistema y que se amplían para dar soporte a las nuevas funcionalidades.

Identificador	RD01
Nombre:	Línea de contrato
Tipo:	Nuevo
Descripción:	Requisito que representa los datos sobre el arrendamiento de una máquina.
Casos de uso:	CU_01, CU_02, CU_05, CU_07, CU_08
Datos específicos:	Nº producto, estado, fecha de entrega, fecha de recogida, fecha fin contrato, tipo facturación, precio, descuento, horas entrega, horas recogida, facturable.
Comentarios:	Nº producto: se seleccionará de una lista de productos disponibles. Estado: se especificará si la máquina está pendiente de entrega, entregada, o recogida. Tipo facturación: se especificará si el arrendamiento de la máquina se facturará de manera diaria o mensual. Precio: se especificará de manera automática en función del producto seleccionado. Descuento: se especificará el descuento que se quiere aplicar al precio de la máquina. Horas entrega: se especificará el número de horas que tiene la máquina en el momento de la entrega. Horas recogida: se especificará el número de horas que tiene la máquina en el momento de la recogida. Facturable: se especificará si el arrendamiento de la máquina está listo para ser facturado.

Tabla 3.13: Documentación del requisito de datos RD01

Identificador	RD02
Nombre:	Contrato
Tipo:	Nuevo
Descripción:	Requisito que representa los datos sobre un contrato.
Casos de uso:	CU_01, CU_02, CU_03, CU_04, CU_05, CU_08, CU_09, CU_10
Datos específicos:	Nº contrato, cliente, tipo, cliente facturación, transportista, fecha facturación, periodo facturación, facturable.
Comentarios:	<p>Nº contrato: se especificará de manera automática al crear un nuevo contrato.</p> <p>Cliente: se seleccionará de una lista de clientes.</p> <p>Tipo: se especificará si el contrato es de alquiler o de mantenimiento.</p> <p>Cliente facturación: se seleccionará de una lista de clientes. Puede ser diferente del cliente o el mismo.</p> <p>Transportista: se seleccionará de una lista de transportistas.</p> <p>Fecha facturación: se especificará qué día del mes se realizará la facturación.</p> <p>Periodo facturación: se especificará cada cuánto tiempo se realizará la facturación.</p> <p>Facturable: se especificará si el contrato está listo para ser facturado.</p>

Tabla 3.14: Documentación del requisito de datos RD02

Identificador	RD03
Nombre:	Plan facturación
Tipo:	Nuevo
Descripción:	Requisito que representa los datos sobre un plan de facturación generado a partir de un contrato.
Casos de uso:	CU_03, CU_05, CU_10
Datos específicos:	Nº contrato, cliente, fecha facturación, precio, nº prefactura, nº factura, facturable.
Comentarios:	<p>Nº contrato: especifica el contrato al que hace referencia.</p> <p>Cliente: especifica el cliente del contrato al que hace referencia.</p> <p>Fecha facturación: especifica la fecha en la que se realizará la facturación.</p> <p>Precio: especifica el precio total de la facturación.</p> <p>Nº prefactura: especifica el número de prefactura cuando se genere.</p> <p>Nº factura: especifica el número de factura cuando se genere.</p> <p>Facturable: especifica si el plan de facturación está listo para ser facturado.</p>

Tabla 3.15: Documentación del requisito de datos RD03

Identificador	RD04
Nombre:	Parte
Tipo:	Extensión
Descripción:	Requisito que representa los datos sobre partes.
Casos de uso:	CU_04, CU_05
Datos específicos:	Avería, nº contrato, nº máquina.
Comentarios:	Avería: se especificará si el parte es de avería o no. Nº contrato: se especificará el contrato al que hace referencia. Nº máquina: se especificará la máquina del contrato al que hace referencia.

Tabla 3.16: Documentación del requisito de datos RD04

Identificador	RD05
Nombre:	Cliente
Tipo:	Extensión
Descripción:	Requisito que representa los datos sobre clientes.
Casos de uso:	CU_05, CU_06
Datos específicos:	Emisión de facturas.
Comentarios:	Emisión de facturas: se especificará si el cliente quiere que se agrupen sus facturas por contrato, por cliente, o que no se agrupen.

Tabla 3.17: Documentación del requisito de datos RD05

3.2. Análisis de requisitos

Para conseguir alcanzar los requisitos anteriores, se ha desarrollado el análisis de una solución que cumpla con los requisitos funcionales del sistema. Para ello, se han utilizado herramientas como el diagrama de clases, o de actividad.

3.2.1. Diagrama de clases

El diagrama de clases es una herramienta que permite representar las clases que forman parte de un sistema y las relaciones entre ellas. En la Figura 3.2 se muestra el diagrama de clases del sistema. No se han especificado los atributos y métodos que contendrá cada clase debido a que se indagará con más precisión más adelante, cuando se explique el contenido de la base de datos.

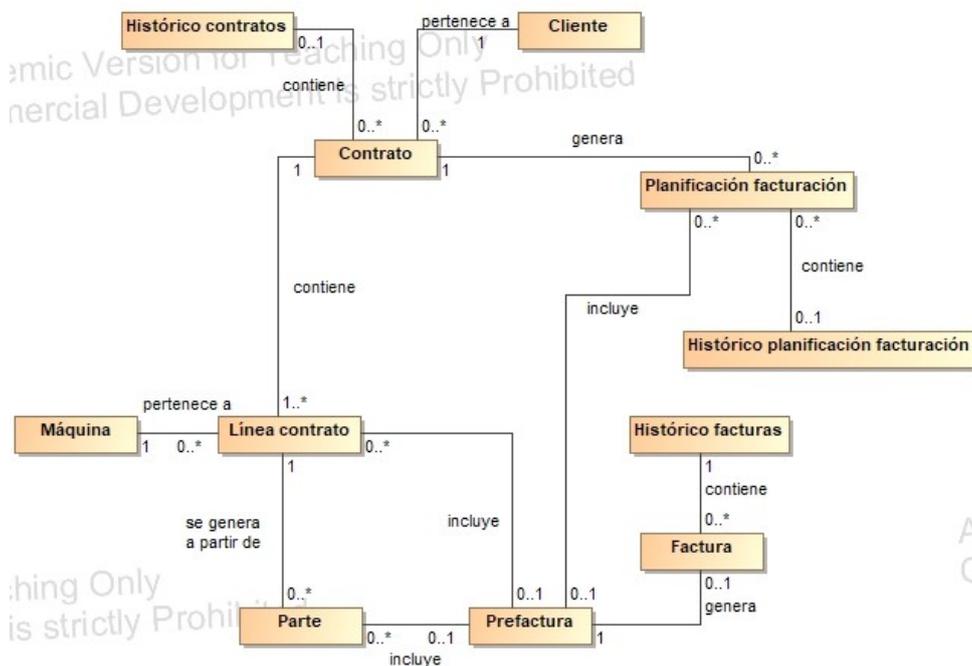


Figura 3.2: Diagrama de clases del sistema

3.2.2. Diagramas de actividad

Los diagramas de actividad son una herramienta que permite representar el flujo de actividades de un sistema. En este caso, el único caso de uso que requiere de una lógica compleja es el caso de uso CU_05 Gestionar prefacturas, por lo que se ha realizado el diagrama de actividad de dicho caso de uso.

En la Figura 3.3 podemos ver como el flujo comienza aplicando los filtros para los datos que utilizaremos para conformar las prefacturas. Estos pueden ser tipo de documento a utilizar como fuente (líneas de contratos de alquiler y planes de facturación de contratos de mantenimiento), si se quieren incluir los partes y fechas de inicio y de fin. A continuación, se genera una prefactura y se añade el primer elemento de los datos de entrada. Entonces, se comprueba si siguen quedando elementos por añadir. En caso de que no fuera el último, se comprueba si son del mismo cliente, y si no lo son, se genera una nueva prefactura y se añade el elemento. En caso de que sean del mismo cliente, se comprueba si son del mismo contrato, y si no lo son, se genera una nueva prefactura y se añade el elemento. En caso de que sean del mismo contrato, se recurrirá al tipo de emisión de facturas del cliente para decidir si se genera una nueva prefactura o no.

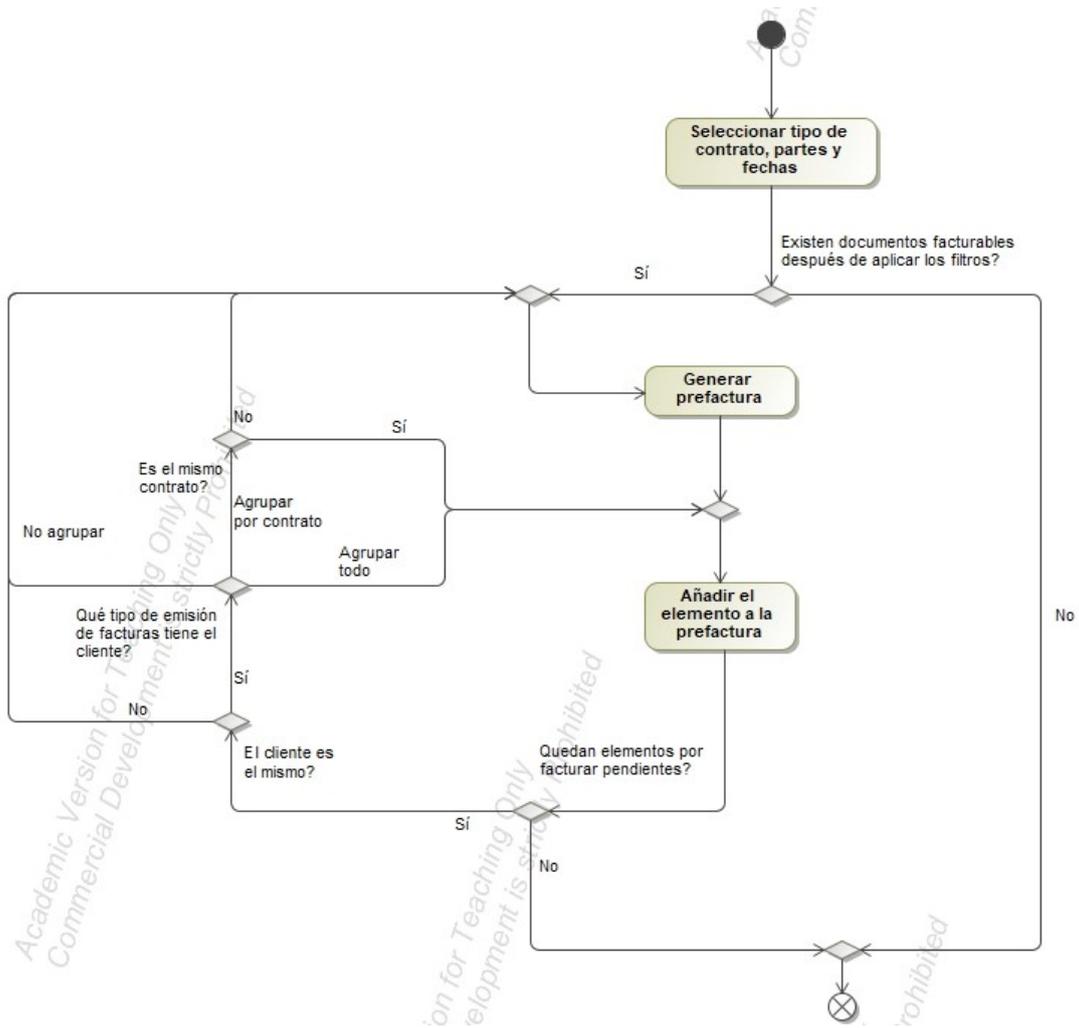


Figura 3.3: Diagrama de actividades del caso de uso CU_05 Gestionar prefacturas

Capítulo 4

Diseño del sistema

4.1. Diseño de la arquitectura del sistema/producto

La arquitectura de un sistema es la estructura o estructuras del sistema, que comprende los elementos del software, los componentes externos, las propiedades externamente visibles de esos elementos y componentes y las relaciones entre ellos.

En este caso, el sistema se compone de dos partes, una parte de cliente web y una parte de servidor. La parte de cliente web es la que se encarga de mostrar la interfaz de usuario y de enviar las peticiones al servidor. La parte de servidor es la que se encarga de procesar las peticiones y de devolver los resultados al cliente.

Detrás del cliente web, hay un servidor de aplicaciones que aloja la instancia de BC. El servidor de aplicaciones es administrado por Microsoft y se encarga de ejecutar y procesar los cambios generados por los usuarios.

Además, también se utiliza una base de datos para almacenar la información y los cambios realizados en la aplicación. Microsoft administra y mantiene la base de datos en la nube, por lo que no se requiere que los usuarios interactúen directamente con esta.

Estas conexiones se pueden ver de manera más sencilla en la Figura 4.1. En esta también se incluyen otros aspectos como Los Servicios Web SOAP (Simple Object Access Protocol) y OData (Open Data Protocol), que se han omitido, ya que no son de importancia para el proyecto.

4.2. Diseño de la base de datos

Debido a que el sistema se ha desarrollado en el ERP BC, la base de datos ya viene definida por defecto. Sin embargo, ha sido necesario desarrollar nuevas tablas para almacenar los datos de los nuevos requisitos.

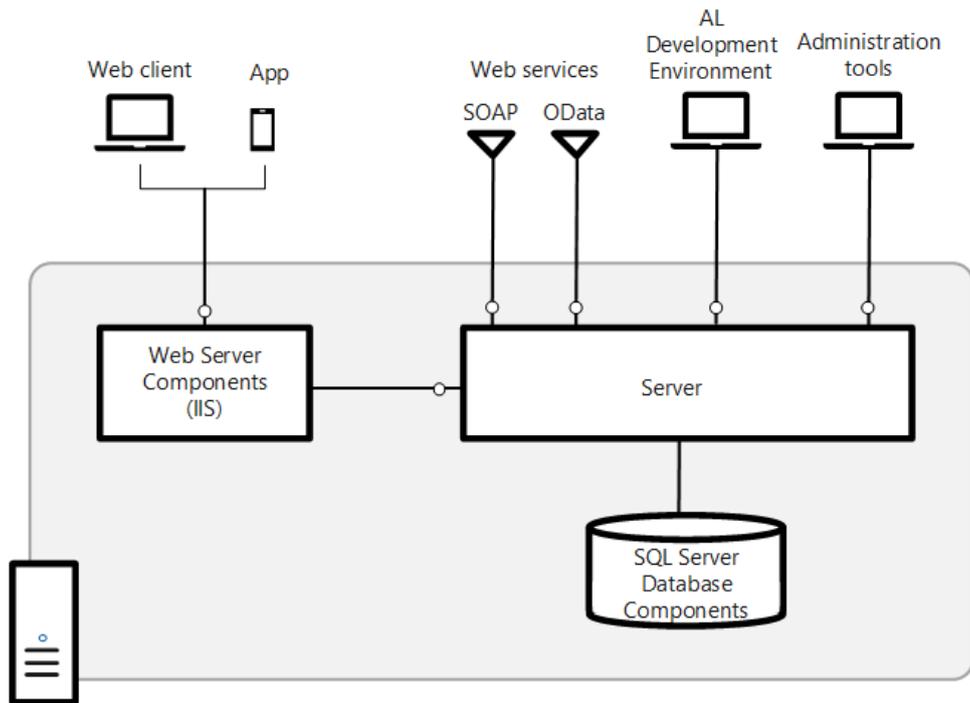


Figura 4.1: Representación visual de la conexión de los sistemas informáticos de BC[5]

En la Figura 4.2 se puede ver el diagrama de la base de datos desarrollada. Como se puede ver, se puede diferenciar entre cuatro tipos de tablas:

- Nuevo: tablas creadas para almacenar los datos de los nuevos requisitos.
- Extensión: campos añadidos a tablas ya existentes. Estas tablas no incluyen los campos ya existentes, ya que dificultarían la lectura del diagrama.
- Existente: tablas ya existentes en el sistema. Tampoco se incluyen los campos de estas tablas por el mismo motivo que en las tablas de extensión. Se incluyen únicamente para mostrar las claves foráneas.
- Histórico: únicamente se incluyen para representar dónde se almacenan los datos históricos, y mostrar que contienen los mismos campos que las nuevas tablas.

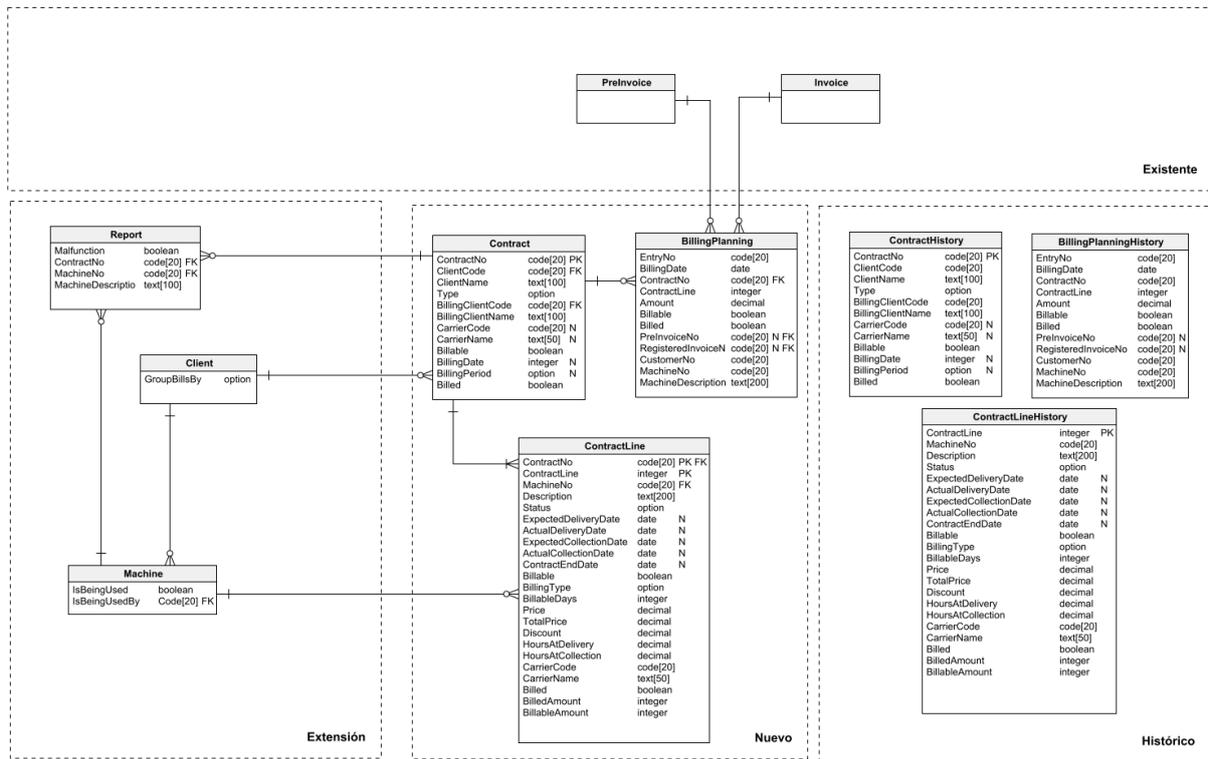


Figura 4.2: Diseño físico de la base de datos

4.3. Diseño de las interfaces

Como en el caso anterior, al estar utilizando un ERP ya desarrollado, las interfaces vienen definidas por defecto. A pesar de que esto puede parecer una desventaja, al quitarle personalización y estilo a los desarrollares, se consigue que la interfaz sea más intuitiva y coherente para los usuarios.

El único cambio que puede apreciarse entre módulos creados por equipos diferentes es la manera en la que se organizará la información. En este apartado se mostrará cómo se ha realizado esta organización.

Para comenzar, desde la página de inicio, junto al menú de navegación, se ha añadido un botón que permite acceder a todas las funcionalidades desarrolladas, como se puede ver en la Figura 4.3.

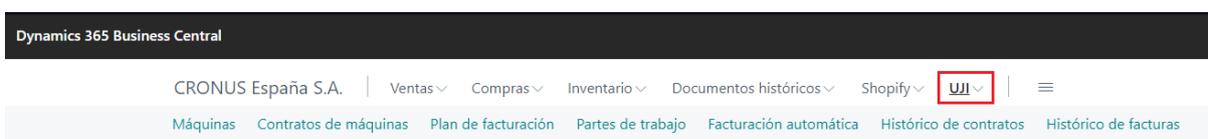


Figura 4.3: Menú de navegación del producto final

Además de esto, la estructura que se suele utilizar en el ERP a la hora de mostrar cualquier elemento suele ser, una lista con todos los elementos, y una ficha con los detalles del elemento seleccionado. En la Figura 4.4 se puede ver un ejemplo de esta estructura en el que se muestra la lista de contratos abiertos actualmente y al seleccionar uno de estos se puede ver la ficha con todos sus datos, como se puede ver en la Figura 4.5.

Nº Contrato	Código Cliente	Nombre Cliente	Tipo	Código Cliente Fact.	Nombre Cliente Fact.	Código Transportista	Nombre Transportista	Factura...	Día de facturación	Periodo de facturación
CT00042	01445544	Progressive Home Furnishings	Alquiler	01445544	Progressive Home Furnishings	DHL	DHL Systems, Inc.	<input type="checkbox"/>	1	Mensual
CT00043	01905893	Candoy Canada Inc.	Mantenimiento	01905893	Candoy Canada Inc.	-	-	<input type="checkbox"/>	3	Mensual

Figura 4.4: Página de lista de contratos del producto final

Tarjeta Contrato | Fecha de trabajo: 23/01/2025

CT00042

Acciones

General

Código Cliente: 01445544 | Nombre Cliente: Progressive Home Furnishings | Tipo: Alquiler | Código Facturación: 01445544 | Nombre Facturación: Progressive Home Furnishings

Código Transportista: DHL | Nombre Transportista: DHL Systems, Inc. | Facturable: | Día Facturación: 1 | Período Facturación: Mensual

Líneas de Contrato

Id	Descripción	Estado	Cód. Transportista	Nombre Transportista	Fecha Entrega Prevista	Fecha Entrega Real	Fecha Recogida Prevista	Fecha Recogida Real	Fecha Fin Contrato	Fact.	Tipo Facturación	Días Facturables	Precio
70066	MaquinaB	Entregada	DHL	DHL Systems, Inc.	10/01/2023	10/01/2023				<input type="checkbox"/>	Diario	8	2000,00

Figura 4.5: Página de ficha de contrato del producto final

Por último, también se ha incluido en la interfaz un minimapa mostrando la ubicación de las máquinas, como se puede ver en la Figura 4.6. Este minimapa se ha decidido incluir dentro de la ficha de la máquina, y no en la lista de estas. De esta manera se ahorran recursos, al tener que actualizarlo únicamente cuando se acceda a la información de una máquina específica.

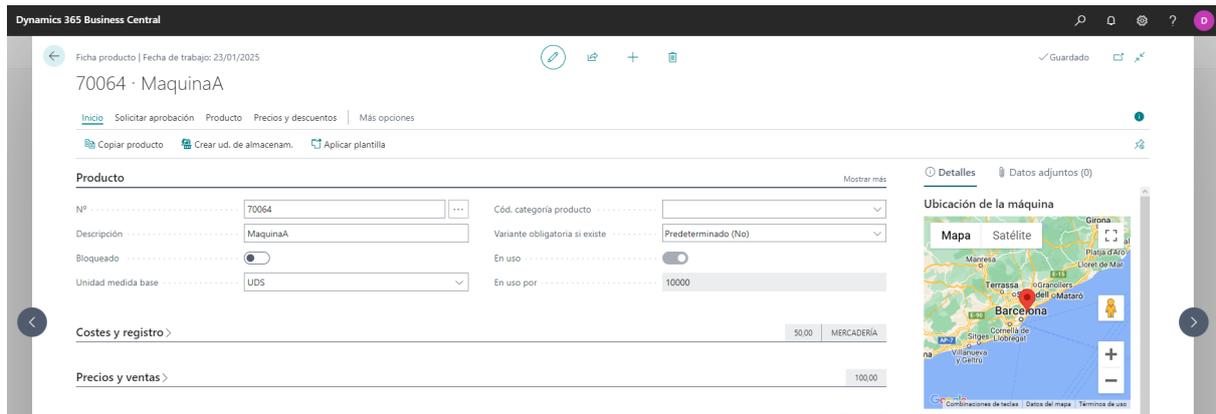


Figura 4.6: Página de ficha de contrato del producto final

Capítulo 5

Implementación y pruebas

5.1. Estructura del código

En el desarrollo del código se ha seguido la estructura que se utiliza en el ERP. Esta estructura consiste en separar el código en diferentes objetos, aunque solo se han utilizado los siguientes:

- **Table:** objeto que representa una tabla de la base de datos. Contiene los campos de la tabla y algunas funcionalidades menores.
- **Table Extension:** objeto que representa una extensión de una tabla. Contiene los campos añadidos a la tabla original.
- **Page:** objeto que representa una página de la interfaz. Obtienen los datos de una tabla y los muestran en la interfaz.
- **Page Extension:** objeto que representa una extensión de una página. Se utiliza para modificar añadiendo o eliminando elementos a una página.
- **Codeunit:** objeto que contiene funcionalidades que se pueden utilizar en cualquier parte del código. A pesar de que otros objetos también pueden contener funcionalidades, se recomienda utilizar los codeunits para que el código sea más legible. También se utilizan para incluir los tests.
- **Report:** objeto que representa un informe. Se utiliza para generar documentos con los datos de las tablas.
- **Control Add-in:** objeto que representa un control de la interfaz. Se utiliza para mostrar elementos que no están incluidos en el ERP.
- **Permission Set:** objeto que representa un conjunto de permisos. Se utiliza para definir los permisos de los usuarios.

El proyecto se ha dividido en dos subdirectorios principales, uno para el código y otro para los tests. Esto se debe a que es necesario crear un objeto de tipo codeunit para cada test, y estos objetos no se pueden incluir en el mismo proyecto que el código principal. Como se puede ver en la Figura 5.1, dentro de cada carpeta, se incluyen ciertos elementos necesarios para el funcionamiento del ERP:

- .alpackages: carpeta que contiene todas las estructuras contenidas en el ERP en ese momento.
- launch.json: archivo que contiene la configuración necesaria para ejecutar el código.
- Translations: carpeta que contiene los archivos .xlf (eXtensible Localization Format) que se utilizan para traducir el código.
- app.json: archivo que contiene la configuración del proyecto.

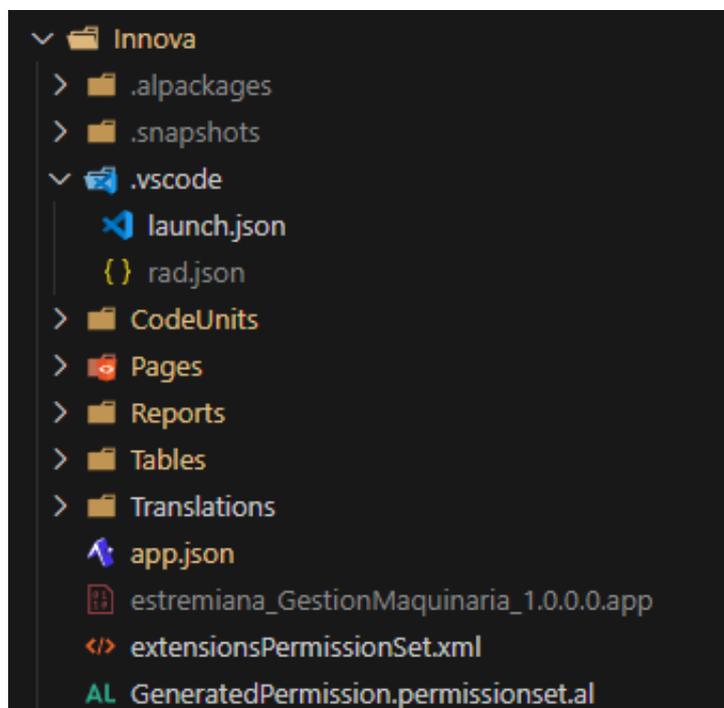


Figura 5.1: Estructura de ficheros y carpetas del proyecto

Además, también se ha decidido crear una carpeta para cada tipo de objeto, haciendo así más sencillo encontrar el objeto que se quiere modificar.

5.2. Descripción técnica de la implementación

Tablas

Una vez con la estructura del proyecto creada, se comenzó con la implementación de las estructuras básicas: las tablas. Es importante mencionar que para todos los componentes desarrollados se ha utilizado la siguiente nomenclatura para objetos nuevos y extensiones, correspondientemente:

```
<ObjectNameSuffix>.<FullNameName>.al  
<ObjectNameSuffix>.<FullNameName>Ext.al
```

Con respecto al idioma utilizado, se ha utilizado el inglés para todos los objetos, ya que es el idioma que se utiliza en el ERP. Sin embargo, para cada elemento que contuviera un nombre en inglés (normalmente utilizado con el atributo “caption”) se ha incluido una traducción al español, como se puede ver en la Figura 5.2. En esta podemos ver como el objeto Contract se ha traducido a Contrato. Estas traducciones se han incluido de manera automática en un fichero .xlf, creado al compilar el proyecto.

Como también se ha podido ver en la Figura 5.2, en lugar de crear tablas directamente con comandos SQL, en BC se utiliza el lenguaje de definición de objetos de AL para definir las tablas y sus propiedades.

Páginas

Más adelante, conforme se tenía la estructura de datos, se fueron creando las páginas para introducir, mostrar, o modificar los datos. En la Figura 5.3 se puede ver el código de la página de la lista de contratos. En este código se puede ver la estructura general de un objeto de este tipo, siendo el apartado “group” el que contiene los campos de la tabla, “part” el que contiene las líneas del contrato, y el apartado “actions” el que contiene los botones de la página. Además, también se incluye cierta funcionalidad necesaria al iniciar la página, o al eliminar un elemento de la lista.

Además, si nos fijamos en como se estructuran los campos de estas páginas, como en la Figura 5.4, se puede ver un “ToolTip” o descripción de cada campo, que también se ha traducido al español, y otras propiedades. Además, también se incluyen “triggers” que se ejecutan para cada acción que se realiza en la página. En este caso, se especificará el cliente de facturación automáticamente al seleccionar al cliente.

```

1  table 50101 "Contract"
2  {
3      DataClassification = ToBeClassified;
4      Caption = 'Contract', Comment = 'ESP="Contrato"';
5      LookupPageId = "ContractHeader";
6      DrillDownPageId = "ContractHeader";
7
8      fields
9      {
10         25 references
11         field(1; ContractNo; Code[20])
12         {
13             DataClassification = ToBeClassified;
14             Caption = 'Id', Comment = 'ESP="Id"';
15         }
16         13 references
17         > field(2; ClientCode; Code[20]) ...
18         2 references
19         > field(3; ClientName; Text[100]) ...
20         71 references
21         > field(4; Type; Option) ...
22         5 references
23         > field(5; BillingClientCode; Code[20]) ...
24         2 references
25         > field(6; BillingClientName; Text[100]) ...
26         9 references
27         > field(7; CarrierCode; Code[20]) ...
28         2 references
29         > field(8; CarrierName; Text[50]) ...
30         5 references
31         > field(9; Billable; Boolean) ...
32         8 references
33         > field(10; BillingDate; Integer) ...
34         16 references
35         > field(11; BillingPeriod; Option) ...
36         0 references
37         > field(12; Billed; Boolean) ...
38     }
39
40     keys
41     {
42         - reference
43         key(Key1; ContractNo)
44         {
45             Clustered = true;
46         }
47     }
48 }

```

Figura 5.2: Código de la tabla Contract

```

1  page 50102 "ContractCard"
2  {
3      ApplicationArea = All;
4      Caption = 'Contract Card', Comment = 'ESP="Tarjeta Contrato"';
5      PageType = Card;
6      SourceTable = Contract;
7
8      layout
9      {
10         0 references
11         area(content)
12         {
13             0 references
14             group(General) ...
15             0 references
16             part(ContractLines; ContractLine) ...
17         }
18     }
19
20     actions
21     {
22         0 references
23         area(Processing)
24         {
25             0 references
26             action(CreateBillingPlan) ...
27             0 references
28             action(ShowBillingPlan) ...
29         }
30     }
31
32     trigger OnInit() ...
33
34     trigger OnDeleteRecord(): Boolean ...
35 }

```

Figura 5.3: Código de la página ContractCard

```

15     field(ClientCode; Rec.ClientCode)
16     {
17         Tooltip = 'Specifies the client code', Comment = 'ESP="Especifica el código del cliente"';
18         Caption = 'Client Code', Comment = 'ESP="Código Cliente"';
19         ShowMandatory = true;
20         NotBlank = true;
21         trigger OnValidate()
22         begin
23             if Rec.BillingClientCode = '' then
24                 Rec.BillingClientCode := Rec.ClientCode;
25                 CurrPage.Update();
26             end;
27     }
28

```

Figura 5.4: Código de un campo de la página ContractCard

Codeunits y Reports

Una vez se tenían las tablas y las páginas, se comenzó con la implementación de las funcionalidades. Para ello, se crearon codeunits que contenían las funcionalidades necesarias para cada proceso.

Una de las partes más complejas ha sido la generación de las prefacturas contenida en el *CU_05 Gestionar prefacturas*. Como ya se ha explicado anteriormente, las prefacturas se generan a partir de las líneas de los contratos de alquiler, la planificación de facturación de los contratos de mantenimiento y de los partes.

Esta problemática se planteó de dos maneras diferentes. Crear un sistema de facturación completamente nuevo independiente del estándar de BC, o adaptarlo al ya existente. Finalmente, se decidió adaptar el sistema de facturación ya existente para así poder aprovechar el resto de procesos internos que este incluye.

Puesto que BC ya incluye la funcionalidad de crear a prefacturas a partir de los partes, el “único” reto era crear las prefacturas a partir de los otros dos elementos, y juntarlas en una única prefactura. El reto que esto suponía era principalmente adaptar los nuevos datos creados (líneas de contrato y planificación de facturación) al formato de los partes, para así poder utilizar el mismo proceso de creación de prefacturas.

Finalmente, se llevó a cabo analizando en detalle la funcionalidad que transformaba los partes en prefacturas, y se adaptó para que también funcionase con las nuevas líneas de contrato y planificación de facturación.

Normalmente, los reports se utilizan para generar informes de datos específicos. En este caso se utilizó para poder seleccionar los datos que se querían incluir en las prefacturas, siguiendo el proceso mencionado anteriormente.

ControlAddIns

Por último, para implementar un minimapa, necesario para el *CU_08 Ver ubicación de máquina*, en el que poder ver la posición de los equipos, se utilizó un controlAddIn. Este controlAddIn se encarga de mostrar un mapa utilizando la API de Google Maps¹ en la ubicación del arrendatario de la máquina en cuestión.

Como se puede ver en la Figura 5.5, se ha tenido que dividir el trabajo en 5 objetos diferentes:

- **ItemMap.PageExt.al**: Este objeto se encarga de enlazar la máquina con el mapa, y de crear espacio en la interfaz para este.
- **MachineMapFactBoxPart.Page.al**: Se encarga de procesar la dirección del arrendatario y llamar al controlAddIn.

¹<https://developers.google.com/maps>

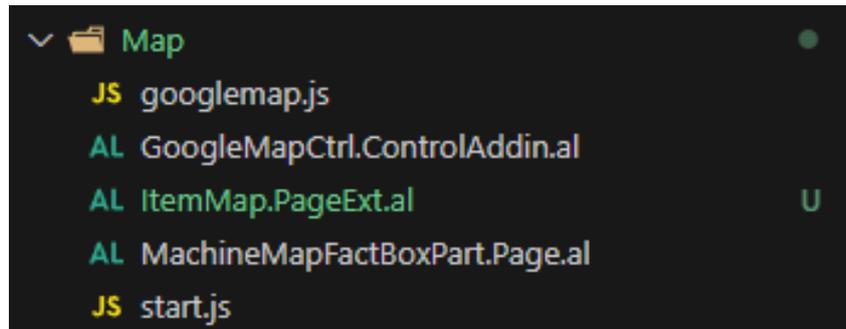


Figura 5.5: Estructura de los archivos necesarios para implementar el minimapa

- **GoogleMapCtrl.ControlAddin.al:** Se encarga de llamar al script que inicializa el mapa y al que lo actualiza con la nueva dirección.
- **start.js:** Se encarga de inicializar la API y de crear la instancia del mapa.
- **googlemap.js:** Se encarga mostrar la nueva dirección en el mapa.

5.3. Pruebas

BC es un ERP que se actualiza constantemente y el cambio de una versión a otra puede suponer cambios en la estructura de los objetos. Por ello, es muy importante realizar pruebas para asegurar que el código sigue funcionando correctamente.

Para este proyecto se ha realizado una serie de pruebas de aceptación para asegurar que el flujo de trabajo sigue funcionando correctamente. Se valoró realizar pruebas unitarias, pero debido a la complejidad de las pruebas, se decidió descartar esta opción. Por esto, se ha creado un objeto del tipo Codeunit para cada caso, en el que se cubrirá toda la funcionalidad del mismo.

En cada prueba se ha utilizado el patrón Given-When-Then[3], que consiste en dividir la prueba en tres partes:

- **Given:** Se especifica el estado inicial de la prueba.
- **When:** Se especifica la acción que se va a realizar.
- **Then:** Se especifica el resultado esperado.

Además, el objetivo de cada prueba es testear un proceso concreto, como crear un contrato de alquiler, teniendo en cuenta posibles escenarios como incluir datos erróneos, o que simplemente no se cumplan las condiciones necesarias para realizar la acción.

Por último, todas estas pruebas se han incluido en un entorno de Continuous Integration & Continuous Delivery (CI/CD) para que se ejecuten automáticamente. Para ello se ha utilizado

GitHub Actions, que permite, mediante flujos de trabajo, automatizar tareas como la compilación del código, la ejecución de las pruebas, o la publicación de la extensión en el marketplace de BC.

Para este proyecto, se ha decidido hacer uso de AL-Go. AL-Go es un repositorio de GitHub creado por Microsoft que contiene un conjunto de plantillas y acciones que se pueden utilizar para llevar a cabo los procesos anteriormente mencionados[4]. Estas plantillas se encargan de crear un entorno en la nube, compilar el código, ejecutar los tests, y comprobar que la extensión es completamente funcional.

En la Figura 5.6 se puede ver un ejemplo de esta ejecución a raíz de un *push* a la rama.

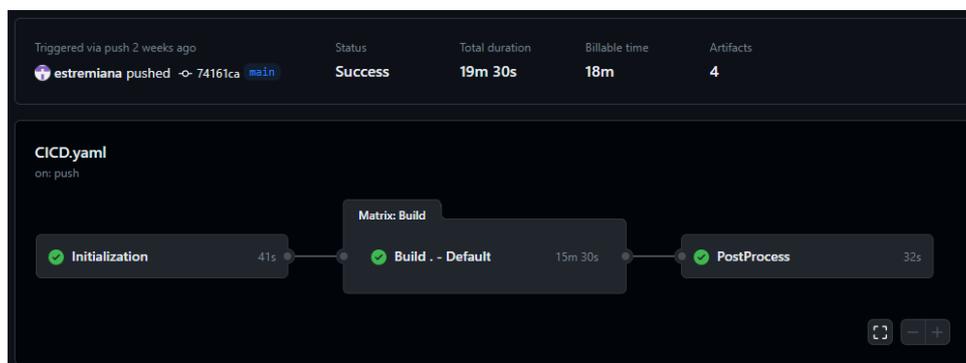


Figura 5.6: Ejecución de las pruebas en Github Actions

Capítulo 6

Conclusiones

A nivel técnico, pienso que el proyecto se ha realizado con un grado de compleción bastante satisfactorio. Es verdad que no se llegaron a cumplir todos los objetivos iniciales propuestos, ya que se tenía intención de incluir un manual de usuario, pero a cambio de esto se han incluido otros apartados que se han estimado más relevantes, como son el cajón de funciones, o los diferentes históricos.

En el ámbito profesional, este proyecto me ha abierto muchas puertas en diferentes sentidos. Por un lado, me ha permitido conocer de primera mano cómo funciona una empresa, y cómo se trabaja en ella. Por otro lado, me ha permitido conocer el mundo de los ERP, y en concreto el de BC, que es un ERP muy utilizado en el mundo empresarial. Además, me ha permitido conocer el mundo de la consultoría, cómo es trabajar con clientes, y cómo se trabaja en equipo.

Personalmente, este ha sido uno de los proyectos más importantes que he realizado hasta ahora. No solo por la complejidad del proyecto, sino también por la importancia que tiene para mí. Este proyecto ha supuesto un gran reto personal, ya que me ha obligado a salir de mi zona de confort, la universidad, y adentrarme en una empresa llena de gente desconocida. Dicho esto, me gustaría añadir que en cierta manera, este documento representa el cierre de mi etapa universitaria, y el comienzo de mi vida profesional.

Bibliografía

- [1] CEUPE. Metodologías de dirección de proyectos. <https://www.ceupe.com/blog/metodologias-de-direccion-de-proyectos.html>.
- [2] Maria Coppola. Qué es javascript, para qué sirve y cómo funciona. <https://github.com/microsoft/AL-Go>, 08 2022.
- [3] Marin Fowler. bliki: Givenwhenthen. <https://martinfowler.com/bliki/GivenWhenThen.html>, 08 2013.
- [4] Freddy Kristiansen. Al-go for github. <https://github.com/microsoft/AL-Go>, 11 2021.
- [5] Microsoft. Component and system topology - business central. <https://learn.microsoft.com/en-us/dynamics365/business-central/dev-itpro/deployment/product-and-architecture-overview>, 01 2023.
- [6] talent.com. Salario para analista business central en españa - salario medio. <https://es.talent.com/salary?job=analista+business+central>.
- [7] talent.com. Salario para desarrollador business central en españa - salario medio. <https://es.talent.com/salary?job=desarrollador+business+central>.