



GRADO EN INGENIERÍA INFORMÁTICA

TRABAJO FINAL DE GRADO

**Solución informática para ERP:
Self Setup**

Autor:
Mihai MANEA

Supervisor:
Carlos Roberto
RODRIGUEZ MORENO

Tutor académico:
José Luis LLOPIS
BORRÁS

Fecha de lectura:
16 de Septiembre de 2019
Curso académico 2018/2019

Resumen

En este documento se detalla la memoria del trabajo final de grado realizado por Mihai Manea durante su estancia en prácticas del grado de Ingeniería Informática de la Universitat Jaume I.

El objetivo principal del proyecto consiste en la creación de una aplicación que también se conoce por el nombre de solución informática basada en el sistema de planificación de recursos empresariales conocido como Microsoft Dynamics 365 Business Central. A partir de ahora se hará referencia a ella con el nombre de solución o solución informática.

La solución, llamada Self Setup, se crea para afrontar las dificultades que conlleva la configuración de una nueva empresa. Mediante un asistente de configuración se llevan a cabo diversos procesos que actúan como guía para utilizar la solución.

De tal manera la integración de la solución permite una amplia y más compatible manera de interactuar con los usuarios finales. Además no se necesitan conocimientos previos ya que es muy intuitiva.

Palabras clave

Microsoft Dynamics Business Central 365, Solución ERP, Aplicación Basada en la Web, Lenguaje de Programación AL, Azure.

Keywords

Microsoft Dynamics Business Central 365, ERP Solution, Web-Based Application, AL Programming Language, Azure.

Agradecimientos

A mis padres por apoyarme en todos los momentos difíciles.

A mi tutor, José Luis Llopis Borrás por apoyarme académicamente y guiarme durante toda la estancia en prácticas.

A mis amigos por estar presentes en mi vida.

Índice general

1. Introducción	9
1.1. Contexto y motivación del proyecto	9
1.2. Objetivos del proyecto	9
1.3. Restricciones	10
1.3.1. Temporales	10
1.3.2. Económicas	11
1.3.3. Recursos humanos	11
1.3.4. Generales	11
1.4. Gestión de riesgos	12
1.4.1. Identificación de riesgos	12
1.4.2. Análisis de riesgos	13
1.5. Estructura de la memoria	15
2. Descripción del proyecto	17
2.1. Descripción General	17
2.2. Herramientas y Tecnologías	17
2.2.1. Microsoft Dynamics 365 Business Central	17
2.2.2. Microsoft Visual Studio	18

2.2.3.	Microsoft Visual Studio Code	19
2.2.4.	Visual Studio .NET	20
2.2.5.	MagicDraw UML Editor	20
3.	Planificación del proyecto	21
3.1.	Metodología tradicional	21
3.2.	Planificación	22
3.2.1.	Primer contacto y creación de estructura de datos	22
3.2.2.	Diseño de las páginas	23
3.2.3.	La funcionalidad de los elementos	23
3.2.4.	La forma de entrega de la solución informática	24
3.3.	Gestión del Alcance	24
3.3.1.	Alcance Funcional	24
3.3.2.	Alcance Organizativo	25
3.3.3.	Alcance Informático	25
3.4.	Estructura de Descomposición del Trabajo (EDT)	26
3.5.	Gestión del tiempo	27
3.6.	Gestión de recursos humanos	27
3.7.	Estimación de recursos y costes del proyecto	28
3.8.	Gestión del Seguimiento	31
3.8.1.	Escenario 1: 45 % del proyecto completado	31
3.8.2.	Escenario 2: 90 % del proyecto completado	32
4.	Análisis y diseño del sistema	35

4.1. Análisis del sistema	35
4.1.1. Investigación y extracción de requisitos	35
4.1.2. Diagrama de casos de uso	38
4.1.3. Historias de usuario	46
4.2. Diseño de la arquitectura del sistema	51
4.3. Diseño de la interfaz	53
4.4. Diagrama de clases	57
5. Implementación y pruebas	59
5.1. Detalles de implementación	59
5.2. Verificación y validación	64
6. Conclusiones	65
6.1. Ámbito formativo	65
6.2. Ámbito profesional	65
6.3. Ámbito personal	66
A. Diagrama de Gantt	69

Capítulo 1

Introducción

1.1. Contexto y motivación del proyecto

La empresa en la cual se ha desarrollado la estancia en prácticas se llama Innova Advanced Consulting[1] y está situada en la ciudad de Castellón de la Plana. Lleva más de 15 años trabajando con el *Enterprise Resource Planning (ERP)*[2] de Microsoft. Este ERP, llamado *Microsoft Dynamics 365 Business Central (BC)* es un programa que se encarga de la gestión de los principales procesos de negocio.

La forma de prestación de los servicios de Innova se traduce como el desarrollo e implantación de extensiones para el ERP que solucionan diferentes necesidades para las empresas que tienen un nivel de gestión de sus recursos informatizados. A las extensiones se les llama, en éste contexto, *solución informática* o simplemente *solución*.

A día de hoy la empresa tiene una pequeña colección de soluciones desarrolladas. Un detalle importante es que las soluciones están desarrolladas también para las otras versiones del ERP como por ejemplo, *Microsoft Dynamics 365 Finance and Operations* y *Microsoft Dynamics 365 Navision*.

El proyecto está basado en la versión BC que es la última disponible y de allí surge la motivación del proyecto, es decir implantar la solución en el nuevo entorno para respaldar la necesidad de negocio para los nuevos usuarios/clientes del *software*.

1.2. Objetivos del proyecto

El principal objetivo del proyecto es la resolución de problemas de gestión de procesos de negocio a nivel informático que aparecen cuando un cliente quiere informatizar su

empresa de tal manera que sea más fácil, óptima, sencilla y rápida.

A la hora de la creación de una empresa virtual en BC se necesita configurar una serie de parámetros para que la gestión funcione correctamente. Se necesitan conocimientos para determinar qué parámetros son los que se tienen que configurar y aquí es donde entra en juego Self Setup.

El objetivo principal de Self Setup es darle la oportunidad al cliente de realizar la configuración necesaria por su cuenta mediante un asistente y sin la necesidad de ningún consultor. El asistente tiene disponibles una serie de pantallas de ayuda y configuración para poder hacer de guía durante todo el proceso.

Los resultados de esta solución se marcan por cada pantalla de configuración del asistente y por lo tanto se espera que al final del proceso todos los datos recogidos se integren en BC.

1.3. Restricciones

1.3.1. Temporales

Un factor importante a la hora de definir las restricciones es el tiempo. Al principio de cualquier proyecto es necesario saber cuanto va a durar para que se tenga un control temporal sobre el mismo.

Si no hay un control del tiempo no se podrán saber ni las fechas provisionales de la finalización del proyecto, ni cuanto desarrollo se va a dedicar a las diferentes áreas, ni los márgenes de error (si los hubiera), etc. Por lo tanto es necesario calcular el tiempo aproximado de la duración total para cualquier proyecto sea cual sea su tamaño.

No es recomendable basarse mucho sobre los datos que son decisivos a la hora de influir en el tiempo total del proyecto sin tener estudios o estimaciones ya que existe una gran probabilidad de que se cree un panorama desfavorable. Los datos con una influencia de decisión podrían ser, por ejemplo, cambios de última hora en el plan contable español debido a un decreto de urgencia.

Esto podría causar una desventaja ya que se tienen que cambiar partes de la solución para que se considere legalmente apta.

Si es que este inconveniente llega a pasar, ocurrirán pérdidas ya sean de dinero o aún peor, del proyecto en sí. Sin embargo, no es el caso en este proyecto dado que no hay una fecha de finalización establecida y por lo tanto se dispone de una flexibilidad mucho más grande que en proyectos con una fecha de finalización a cumplir.

1.3.2. Económicas

Self Setup forma parte del conjunto de soluciones que actualmente tiene la empresa Innova Consulting. El objetivo más valioso en cualquier empresa es el de recibir unos beneficios conformes a sus productos.

Aparte del precio del pack para la solución Self Setup que dependerá del departamento de marketing, no se encuentran otro tipo de restricciones económicas que perjudiquen directa o indirectamente el producto.

1.3.3. Recursos humanos

La plantilla de desarrollo que está trabajando actualmente sobre el proyecto está formada por dos personas. Como es una solución bastante compleja una de las restricciones a considerar es que solo hay dos personas que trabajan en el proyecto.

Además, teniendo en cuenta que el espacio de las oficinas de la empresa esta limitado y no hay más posibilidades de nuevas incorporaciones, indica que existe una consecuencia de la restricción de personal. Aquí no se tienen en cuenta los recursos humanos de los otros departamentos.

1.3.4. Generales

Otro tipo de restricción que presenta actualmente la solución pertenece al ámbito territorial. Self Setup trabaja actualmente con plantillas definidas para planes contables, cuentas, divisas, etc.

En cada país el Plan General de Contabilidad es diferente y lleva tiempo hacer un desarrollo que sirva para más de un país. Las plantillas creadas hasta el momento se ajustan solamente a nivel nacional de España.

Por lo tanto, no se podrá, hasta que no se saque una nueva versión, utilizar por clientes que estén fuera del rango del alcance de la solución.

La integridad es otra de las restricciones que crean el vínculo que tiene la solución con Business Central ya que está hecha solamente para la plataforma en sí. No se podrá instalar en otro entorno que no sea el adecuado.

1.4. Gestión de riesgos

La gestión de riesgos es una de las fases más importantes en un proyecto. Una apropiada implementación, planificación, documentación y monitorización de los riesgos nos va a ayudar a garantizar el éxito del proyecto.

La identificación de riesgos detectada en los apartados anteriores va a suponer la preparación de unas tareas de prevención para así tener con claridad cuáles son las resoluciones de cada uno de los riesgos de manera individual. Además de esto se podrán hacer cambios y mantener actualizados aquellos riesgos que cambien o, en el peor caso, que pasen.

La gestión de riesgos está, actualmente, gestionada por los desarrolladores ya que tienen un conocimiento alto sobre los posibles situaciones en las cuales se podría producir una contingencia.

1.4.1. Identificación de riesgos

Este apartado pretende dar a conocer los diferentes tipos de riesgos que se crean durante la realización del proyecto. Se definirán, para cada riesgo en particular, una descripción para entender el contexto y el tipo teniendo en cuenta la gravedad de la cual procede. La tabla 1.1 muestra los detalles de cada uno.

ID	Descripción del riesgo	Tipo del riesgo
R01	Cambios en los requisitos	Riesgo del producto
R02	Falta de experiencia en tareas de implementación	Riesgo del producto
R03	Falta de conocimiento en pruebas unitarias	Riesgo del producto
R04	Falta de comunicación entre los desarrolladores del proyecto	Riesgo del producto
R05	Cambios relacionados con la legislación	Riesgo del proyecto
R06	Falta de conocimiento en planificación	Riesgo del proyecto

Tabla 1.1: Resumen identificación de riesgos

1.4.2. Análisis de riesgos

En este apartado se va a realizar un análisis más detallado de los riesgos anteriormente identificados. Se van a señalar la importancia, una descripción y qué es lo que los origina.

ID	Riesgo
R01	<p>Importancia: alta.</p> <p>Descripción: representan la idea que se tiene al principio de la solución. Sirven de guía para todo el proceso de desarrollo.</p> <p>Origen: cambio de última hora para añadir/quitar funcionalidad.</p>
R02	<p>Importancia: media-alta.</p> <p>Descripción: hace referencia a los conocimientos adquiridos previamente para poder desarrollar bien el producto.</p> <p>Origen: que el alumno este o no preparado.</p>
R03	<p>Importancia: media</p> <p>Descripción: no se tiene experiencia necesaria en tal campo para poder ser productivo.</p> <p>Origen: depende del tiempo de preparación previo.</p>

R04	<p>Importancia: alta</p> <p>Descripción: es un punto muy sensible si afecta a más personas y no tan grave si afecta solo a dos, por ejemplo. De todos modos, se centra en la comunicación de aquellas tareas que dependen unas de otras y se necesita la interacción para que el producto esté en condiciones favorables.</p> <p>Origen: compatibilidad profesional baja, diferentes tipos de caracteres, personalidad, etc.</p>
R05	<p>Importancia: alta</p> <p>Descripción: tiene que ver con algunos cambios de carácter legislativo que impone el estado. Nuevas leyes implantadas que impidan el funcionamiento correcto del producto.</p> <p>Origen: creación de nuevas leyes desfavorables.</p>
R06	<p>Importancia: media</p> <p>Descripción: no se conocen las buenas prácticas de la planificación.</p> <p>Origen: mala comprensión de las técnicas de planificación.</p>

Tabla 1.2: Resumen análisis de riesgos

1.5. Estructura de la memoria

En el **Capítulo 1** de esta memoria se describen los objetivos tanto principales como aquellos secundarios, y la motivación para desarrollar el proyecto. Por otra parte, se presenta la empresa como entidad además de hacer una descripción acerca de la sede y entender el contexto en el cual actúa.

En el **Capítulo 2** se presentarán de una manera más detallada aquellas tecnologías y herramientas usadas a la hora de realizar el proyecto y también el punto desde donde se parte con el desarrollo.

En el **Capítulo 3** se presentarán varias partes que tienen relación con la planificación del proyecto. La metodología escogida al empezar, la manera de trabajar al principio y al final del desarrollo y por último la estimación de los costes y recursos.

En el **Capítulo 4** se presentará la parte de análisis que básicamente consta en la recogida de los requisitos. También forman parte el diseño de aquellos componentes del proyecto como por ejemplo el diagrama de casos de uso o interfaces.

En el **Capítulo 5** se presentará la implementación de la fase de desarrollo, así como las pruebas a realizar para validar y verificar el contenido.

En el **Capítulo 6** se presentará la conclusión final objetiva y subjetiva relacionada con el proyecto. Además, se dan unos puntos de vista profesionales vistos en la estancia de prácticas.

Capítulo 2

Descripción del proyecto

2.1. Descripción General

El proyecto consiste en el desarrollo de una solución informática para un determinado *software*, en este caso el *Enterprise Resource Planning (ERP)* para *Microsoft Dynamics 365 Business Central*. Con la creación de ciertas soluciones como por ejemplo *Self Setup* entraremos en más detalle a continuación. La idea surgió debido a que era necesario instalar soluciones donde el cliente tenía la necesidad. Por lo tanto, se creaban una serie de tareas que no se podían ejecutar desde la empresa.

Cuando un cliente solicitaba que se le implantase cualquier solución en su ERP, un consultor de la empresa tenía que desplazarse a la dirección correspondiente, reunirse con el cliente y por último pasar los requisitos al equipo de desarrollo el cual se dedicaría a la implementación.

Como el proceso era lento el equipo del departamento de *NAVISION (NAV)* propuso una alternativa al problema, crear una solución llamada *Self Setup*. Esta solución completaría casi todo el trabajo del consultor y parte del trabajo de los desarrolladores.

2.2. Herramientas y Tecnologías

2.2.1. Microsoft Dynamics 365 Business Central

Business Central[3] forma parte de la familia de software de sistemas de planificación de recursos empresariales creado por Microsoft. Más concretamente es el sucesor de Microsoft Dynamics NAV y que basa parte de su desarrollo en el código base del mismo.

Está relacionado con el sistema operativo nativo, Windows, que proporciona fiabilidad ya que es un producto muy popular en el mundo empresarial.

Una de las principales ventajas del ERP es que trabaja con los datos mediante la nube. Esto conlleva a que todo esté relacionado entre diferentes cuentas y además accesible. Asimismo, mantiene la posibilidad de recuperar los datos, copias de seguridad de datos, sin infraestructura que mantener.

Hay otra forma de adquirir el acceso al producto como, por ejemplo, *on-premise*, donde los datos se almacenan localmente, es decir que la sincronización está deshabilitada. También existe el despliegue híbrido que es como una mezcla entre el primero y segundo. Parte de los datos se almacenan en el cliente y parte en la nube.

La empresa posee varias licencias de uso relacionadas con cada departamento, ya que son productos diferentes. Con relación a la solución Self Setup se que trabaja sobre la versión en la nube se ha procedido a la entrega de las credenciales, al alumno de prácticas necesarias para poder entrar en el portal.

El objetivo del anterior proceso es facilitar con los datos y herramientas mínimos al alumno para que pueda desempeñar las actividades que se le asignan a lo largo del proyecto.

Referente a la solución que se va a implantar, se ha elegido la opción basada en la nube debido a que era necesario hacer ciertas pruebas para ver el comportamiento general. Además, como es una solución global se va a publicar en la tienda oficial de Microsoft y será accesible en diferentes partes del mundo.

El rol que tiene Business Central referente a la solución es aquel de albergar implementaciones como esta y permitir ser utilizadas por diferentes usuarios dependiendo de su licencia. Al mismo tiempo el *partner* [2] (falta citar bien) desarrollador de cualquier solución queda como el creador sin tener ningún tipo de deuda para Microsoft.

2.2.2. Microsoft Visual Studio

La herramienta que se encarga de gestionar una gran parte del proyecto y de todas las soluciones, así como vistas, controladores, modelos, generación de paquetes, etc. recibe el nombre de *Microsoft Visual Studio* [4].

Visual Studio es un entorno de desarrollo integrado de Microsoft creado para facilitar las tareas del programador y diseñado para crear aplicaciones web. Incluye un editor de texto que soporta diferentes lenguajes de programación tales como C++, C Sharp y también gestión de versiones.

Es posible adquirirlo para diferentes versiones ya sean de comunidad, profesional o

para empresas cada una teniendo sus ventajas. La versión utilizada en la estancia en practicas es la empresarial ya que tiene integrados los servicios de *Microsoft Azure*[5] con su correspondiente cuenta de usuario.

Estos servicios sirven para poder subir las soluciones a la plataforma *AppSource*[6] de Microsoft que se considera como una tienda virtual.

En relación con el proyecto la solución se publica cuando se tiene un avance significativo respetando la versión anterior estable del paquete. Este principio se sigue en todos los demás departamentos. Pero en nuestro caso se aplicaría el paso anterior después de publicar oficialmente la versión inicial.

Aún no se sabe si se publicará una versión Beta de la solución. Sin embargo, dependiendo de la evolución de esta el director ejecutivo tendrá la última decisión.

En cuanto al responsable de la tarea para confirmar que la solución tiene un comportamiento óptimo para ser publicada será el supervisor. El alumno solamente hará desarrollos y pruebas, pero no tendrá la responsabilidad de tomar decisiones a cargo de la empresa en este ámbito.

Esto no quiere decir que no se le da la libertad a la hora de implantar diferentes perspectivas en relación con la implantación, desarrollo, opinión, mejoras sobre la solución, etc.

2.2.3. Microsoft Visual Studio Code

Visual Studio Code[7] pertenece a la familia de editores de texto desarrollados por Microsoft para los sistemas operativos *Windows*, *Linux* y *Mac*. Es un programa que incluye diferentes herramientas que dan soporte a la depuración, un modo de énfasis de sintaxis, control de versiones como por ejemplo *Git*[8] y otras extensiones.

La licencia es de tipo gratuito para un uso comercial o para aquellos usos privados.

Debido a estas ventajas se ha elegido el programa para hacer el desarrollo de las partes que componen la solución en su totalidad. El lenguaje del desarrollo en la cual se ha basado esta parte del proyecto se llama “AL” que es propio de *Microsoft* y además tiene relación con su predecesor C/AL.

En el proyecto se hace un uso diario de la herramienta ya que, al formar parte del ecosistema de *Microsoft*, se integra perfectamente. Además, tiene también extensiones para los servicios de *Azure* como por ejemplo *Timetracker*[9], lo que indica una comodidad considerable.

2.2.4. Visual Studio .NET

Visual Studio .NET[10] es un kit de desarrollo de software más comúnmente conocido en inglés como SDK (software development kit)[11] que se define como una agrupación de diversas herramientas para crear aplicaciones, interfaces gráficas, formularios, servicios y aplicaciones web, para un determinado sistema operativo.

El Sistema sobre el cual trabaja el kit es el mismo desarrollado por Microsoft, llamado Windows.

2.2.5. MagicDraw UML Editor

Magic Draw[12] es un programa de modelado de sistemas diseñado por diferentes tipos de analistas con conocimiento en diferentes áreas. Su función es hacer más ameno el diseño y análisis de sistemas de información o bases de datos orientados a objetos.

Cuenta con los medios necesarios para el desarrollo de una gran cantidad de diagramas independientemente de la metodología utilizada. Dentro de lo que es la interfaz de usuario, los controles son muy intuitivos y no hay ninguna necesidad de requerir formación previa para poder entender su funcionamiento.

Contiene una versión de trabajo en equipo que brinda la posibilidad de trabajar sobre el mismo modelo al mismo tiempo, acceso remoto y una manera de no interferir con otros usuarios.

Los puntos más significativos son:

- La creación de una interfaz amigable e intuitiva para poder acceder a todos los controles en un solo clic que también puede ser configurable.
- La creación de diferentes diagramas que pueden ser obtenidos rápidamente de otros modelos como por ejemplo de *JAVA*, *CSharp*, *C++*, etc
- Tener la posibilidad de trabajar sobre el mismo modelo para poder agilizar el proceso de desarrollo
- Enlazar elementos en diferentes modelos o documentos fuera del modelo actual. Una personalización acerca de la navegación del modelo para las necesidades específicas.

En lo referente al proyecto el programa ha sido usado para crear los diagramas de casos de uso que describen la interacción de los sistemas con la solución Self Setup.

Capítulo 3

Planificación del proyecto

3.1. Metodología tradicional

El proyecto se ha desarrollado siguiendo una metodología tradicional. Había una propuesta del cambio de la metodología por una ágil pero en ese momento el departamento estaba pasando por un cambio.

Se decidió seguir usando la misma hasta acabar el proyecto Self Setup por las siguientes razones:

- El tipo de metodología utilizada durante todos los proyectos anteriores ha sido una tradicional y por lo tanto los desarrolladores se podían basar en una experiencia ya adquirida.
- La introducción a una nueva metodología, en ese momento, suponía la dedicación a dar el paso del cambio de una a otra, lo que llevaría tiempo.

El último proyecto en el cual se ha utilizado una metodología de este tipo ha sido Self Setup. Al finalizar el proyecto se decidirá qué modelo de procesos se elegirá para el futuro y que tenga un beneficio de acuerdo a la cultura de la empresa.

Algunos de los inconvenientes del modelo tradicional, también llamado modelo en cascada, de acuerdo al proyecto se van a describir a continuación:

- **Los pasos raramente siguen un ciclo secuencial:** este problema se ha podido comprobar cuando en el desarrollo del asistente se ha tenido que añadir, después de completar las fases necesarias, código adicional porque era necesaria una nueva funcionalidad que no se había detectado previamente.

Si el proyecto tiene un desarrollo lento o es de proporciones bastante grandes tal tipo de inconvenientes siempre van a perjudicar el tiempo total. Por esa misma razón se decidió que se debería cambiar la metodología de trabajo, es decir, elegir otra que no sea la tradicional para los siguientes proyectos .

- **Es difícil establecer los requisitos desde el principio:** hace referencia a que si en las fases finales del proyecto se detecta algún otro requisitos es difícil volver atrás y añadir cierta funcionalidad.
- **Resultados no visibles hasta el final de las últimas etapas:** Si solo se pueden ver los resultados del trabajo al final, los errores tampoco se podrán detectar hasta ese instante. Como se puede notar los problemas aumentan con este tipo de modelo.

No es una metodología muy recomendable comparada con otras más nuevas.

3.2. Planificación

En cuanto a la planificación del proyecto cabe decir que se centra en cumplir los objetivos definidos de una manera uniforme. Debido a que algunas tareas han surgido posteriores a la planificación, esta no se ha podido seguir en su totalidad.

El motivo viene a ser uno de carácter temporal porque al mismo tiempo que transcurría el desarrollo de este proyecto, otros proyectos secundarios se estaban monitorizando.

Ya que tenían una importancia considerable y no se les podía suspender, se tomó la decisión de no dar soporte a Self Setup. Como consecuencia se detuvo el desarrollo del mismo en aquellos casos puntuales para resolver incidencias de las otras soluciones.

Por esta razón el desarrollo ha tenido contratiempos, pero como no tenía fecha de finalización no ha causado ningún inconveniente.

La parte de la planificación se podría resumir en cuatro apartados.

3.2.1. Primer contacto y creación de estructura de datos

Antes de empezar se hace una reunión con el personal encargado del desarrollo principal y también los jefes de departamento encargados de monitorizar y gestionar el proyecto. Con ésta reunión se espera que se tome una decisión acerca de que es lo que se quiere conseguir al final.

Una primera fase sería la detección de un problema o en este caso la detección de una carencia de módulo. Luego hacer un análisis para ver lo que se necesita, lo que no se

necesita, lo que se tiene y lo que no, los factores que influyen dado que a veces son difíciles de definir sin tener retroalimentación a priori. Se establecen los objetivos principales y los requisitos que tiene que cumplir el proyecto.

Cuando se empieza a crear la solución hay que tener claro la estructura de datos que se va a implementar para gestionar los datos de una manera eficiente. Dependiendo de qué tipo de uso se le va a dar al código

3.2.2. Diseño de las páginas

La segunda parte del proyecto está formada por la creación/diseño de las páginas que representan la información para el usuario de manera entendible y accesible. Con este paso se llega a tener una comprensión gráfica más entendible de lo que se crea.

Cabe destacar que es la única parte de interacción con el usuario final, por lo tanto, es la puerta que une el producto con el mundo exterior.

3.2.3. La funcionalidad de los elementos

La funcionalidad sería el paso en el cual se crean todas las interacciones que se pueden hacer como por ejemplo cargar los datos de un cliente mediante un identificador único en el asistente de Self Setup para que así se reconozca el perfil del mismo. A la hora de crear toda la configuración para su empresa los datos ya quedarían guardados.

Otra parte que forma el punto de la funcionalidad son los *CodeUnits*, las páginas y tablas.

Los *CodeUnits* son unos contenedores de funciones que se encargan de proporcionar funcionalidad a nivel transparente para las páginas que muestra la solución.

Las páginas acceden a las tablas para recoger los datos y mostrar la información de forma gráfica, coherente y correcta.

3.2.4. La forma de entrega de la solución informática

La última fase consta en tomar la decisión para ver de qué forma se entregará la solución.

- Paquete de datos que se instala directamente en el *ERP*.
- Instalación directa en la nube como una extensión.

La más válida es la opción de instalación en la nube por flexibilidad, accesibilidad, etc.

3.3. Gestión del Alcance

3.3.1. Alcance Funcional

El sistema sirve para disminuir la acumulación de tiempo y ofrece simplicidad a la hora de realizar una configuración para cualquier empresa nueva.

1. El sistema permitirá la carga de los datos mediante un identificador único de cliente.
2. El sistema permitirá la creación de una configuración desde cero para una nueva empresa.
3. El sistema permitirá la configuración por separado de aquellas configuraciones que se necesiten.
4. El sistema permitirá el cambio de los datos hasta el momento de finalizar el proceso.

3.3.2. Alcance Organizativo

Self Setup es una solución que depende de *Business Central* y por lo tanto los usuarios que accedan a ella también necesitarán este producto.

Las áreas externas de trabajo y uso de la solución están formadas en una grande proporción por clientes. Otra parte que también usa el producto es la formada por los *partners* pero en una proporción menor comparada con la de los clientes.

Self Setup se puede utilizar independientemente del cliente y existe la posibilidad de hacer un uso por *tenants*[13]. Esto quiere decir que se da acceso a la solución a más personas a la vez.

Los sistemas externos con los cuales se relaciona no están fuera del entorno de trabajo, sino que pertenecen al mismo conjunto de aplicaciones. Debido a esta ventaja lo que implicaría la seguridad no sería ningún problema ya que Business Central cuenta con un amplio conocimiento en cuanto al mantenimiento de conexiones, seguridad de datos privados.

3.3.3. Alcance Informático

En cuanto al alcance informático los sistemas que interactúan con Self Setup son:

- **Business Central** siendo la plataforma base de trabajo.
- **Microsoft Azure**: es la plataforma de Microsoft basada en la nube creado para almacenamiento, testeo, lanzamiento, y creación de aplicaciones o servicios. Self Setup tiene funcionalidad almacenada en este sistema y cada vez que se ejecuta por ejemplo una comprobación de campos de usuario tienen que interactuar.
- **Portal Innova**[14]: es la página oficial de la empresa que también está almacenada en la nube de Microsoft Azure. Interactúan cuando se cargan datos de la base de datos de clientes y Self Setup.

3.4. Estructura de Descomposición del Trabajo (EDT)

En este apartado se realiza la creación de la estructura de todo el proyecto mediante un EDT. Las actividades se van a mostrar en un tipo de lista jerarquizada para poder entenderlas mejor.

1. Proyecto Self Setup

1.1. Desarrollo Planificación

1.1.1. Inicio

- 1.1.1.1. Revisar la definición del proyecto

- 1.1.1.1. Revisar requisitos del proyecto

- 1.1.1.1. Definir metodología de trabajo

1.1.2. Planificación

- 1.1.2.1. Gestión del alcance

- 1.1.2.1. Gestión de riesgos

- 1.1.2.1. Gestión de tiempo

- 1.1.2.1. Gestión de recursos humanos

- 1.1.2.1. Gestión de costes

- 1.1.2.1. Gestión de seguimiento

- 1.1.2.1. Gestión de cierre

1.2. Desarrollo técnico

1.2.1. Análisis

- 1.2.1.1. Diagrama de clases

- 1.2.1.1. Validación y verificación por parte de Project Manager

1.2.2. Diseño

- 1.2.2.1. Diseño de interfaces

- 1.2.2.1. Diseño relación ERP y módulo Solución

1.2.3. Implementación

- 1.2.3.1. Creación relación entre ERP y módulo Solución

- 1.2.3.1. Interfaces de los módulos

- 1.2.3.1. Crear paquete de la solución

1.2.4. Pruebas

- 1.2.4.1. Diseño de pruebas

- 1.2.4.1. Ejecución de pruebas

- 1.2.4.1. Comprobación y solución de errores

1.2.5. Entrega Proyecto

- 1.2.5.1. Auto formación usuarios

- 1.2.5.1. Mantenimiento solución

- 1.2.5.1. Publicación manual oficial

3.5. Gestión del tiempo

Saber cómo gestionar el tiempo es una de las actividades mas primordiales ya que proporciona una visión general de lo que cuesta completar cada tarea. Así se tiene una organización mucho más simple.

Existen herramientas para crear y gestionar el tiempo de cada actividad en particular para aquellos proyectos que lo necesiten. En este caso se ha usado el Diagrama de Gantt[15] para mostrar cada tarea con su correspondiente medida temporal. En el **Anexo A** se presenta el diagrama completo.

3.6. Gestión de recursos humanos

A pesar de los conflictos que puedan generarse entre los integrantes que forman los recursos humanos en cualquier proyecto hay que realizar una distribución, si es posible por roles, del personal.

En este apartado se muestran los roles y responsabilidades de los desarrolladores que trabajan actualmente en la solución. La siguiente sección hace un resumen de asignaciones.

Roles	
Rol	Responsabilidad
Gerente Proyecto	La persona que se encarga de dirigir el Proyecto y dictar las pautas a largo plazo.
Supervisor/Programador	La persona encargada del desarrollo y delegar las tareas a los otros programadores.
Programador	La persona que implementa las tareas dictadas por el Supervisor.

Tabla 3.1: Resumen roles del proyecto

Los roles anteriores se aplican a las siguientes personas:

- Mihai Manea: Programador, Diseñador gráfico, Tester.
- Carlos Rodriguez: Supervisor, Programador, Analista.
- Ignacio Sainz: Gerente Proyecto.

3.7. Estimación de recursos y costes del proyecto

La parte de gestión de costes mantiene información acerca de los costes económicos del hardware y software necesarios para la realización de las actividades del proyecto.

El coste total del proyecto ha estado influenciado por el tiempo, teniendo en cuenta el coste del desarrollo.

El siguiente apartado contiene la estructura de los puntos que definen la gestión de costes en forma de tablas. Se definirán los costes por separado agrupándolos en diferentes estructuras detallando los recursos utilizados en el proyecto para su desarrollo.

La cantidad de tiempo total de la estancia y por lo tanto que relaciona el proyecto con el desarrollo de la solución son de 300h laborales. Eso se transformaría en aproximadamente 2 meses de tiempo de un trabajador a tiempo completo.

A continuación, se van a presentar los costes reales mediante las siguientes tres tablas: recursos humanos, hardware y software para hacer una diferencia visible entre los factores que condicionan las tareas.

Como se puede observar en la Tabla 3.2, el personal disponible para el proyecto esta formado por solamente dos programadores, uno Junior y otro Senior. Esto no implica unos costes muy elevados en cuanto a personal, pero tampoco se espera una producción muy elevada ya que es un proyecto bastante ambicioso y necesita bastante tiempo de desarrollo.

Recursos Humanos				
Puesto	Precio/h	Horas/Semana	Horas/Mes	Total/Mes
Programador Junior	10€	6*5(días) = 30h	30*4 = 120	120*10 = 1200€
Supervisor Programador	13€	8*5(días) = 40h	40*4 = 160	160*13 = 2080€
Total				3280€

Tabla 3.2: Resumen costes de Recursos Humanos

En lo que se refiere a la Tabla 3.3, algunas de las licencias son de tipo gratuito debido a que la empresa es un *Independent software vendor (ISV)* y pertenece a *Silver Data Platform* de Microsoft.

Silver Data Platform es una plataforma encargada de abarcar aquellas empresas que son *partners* de *Microsoft* y que son confirmados por como que cumplen unos requisitos específicos como, por ejemplo, realizan trabajo efectivo, aportan ayuda a los clientes relacionados con productos y servicios específicos.

Al pertenecer a un grupo con privilegios se les proporciona licencias de uso gratuitas o con descuento para sus productos.

Software					
Producto	Licencia	Suscripción /Mes	Usuarios	Total/Mes	Total/Año
Business Central	ISV Partner	-	Hasta 20	Sin coste	Sin coste
Azure	ISV Partner	130€	-	130	130*12 = 1560€
Suscripción				500€	6000€
Visual Studio	ISV Partner	-	Hasta 15		
Windows Server	ISV Partner	-	Hasta 6		
Windows 10	ISV Partner	-	Hasta 25		
Office 365	ISV Partner	-	Hasta 50		
Total				630€	7560€

Tabla 3.3: Resumen costes de Software

En el apartado de hardware y tal como se muestra en la Tabla 3.4 existe un contenido bastante resumido ya que con lo que se tiene se puede completar el proyecto más que satisfactoriamente.

Hardware			
Producto	Cantidad	Precio	Total
Portátil Lenovo	1	850€	850€
Portátil Acer	1	450€	450€
Pantalla Acer (adicional)	2	2*75€	150€

Tabla 3.4: Resumen costes de Hardware

Después de calcular el presupuesto para los apartados anteriores se calcula el presupuesto total del proyecto que se puede encontrar en la Tabla 3.5.

Presupuesto Coste Total Proyecto		
Recursos	Total	Total/Meses del Proyecto
Total Recursos Humanos	-	3280€(mes)*2 = 6560€
Total Software	-	630€(mes)*2 = 1260€
Total Hardware	1450€	-
Total Proyecto	9270€	

Tabla 3.5: Resumen costes de Hardware

3.8. Gestión del Seguimiento

El siguiente apartado se van a indicar dos problemas a lo largo de la vida del proyecto. Estos escenarios se detectarán cuando el proyecto este al 45 % del desarrollo y al 90 % respectivamente. Para cada problema se estudiarán sus posibles soluciones, así como la descripción de cada escenario.

3.8.1. Escenario 1: 45 % del proyecto completado

El desarrollo del proyecto seguía una fluidez favorable hasta que el día 10/03/2019 uno de nuestros desarrolladores, Carlos Rodriguez, sufrió un accidente. Tras una visita al médico el diagnostico fue desfavorable recomendándole una baja laboral de 7 días. Tal indisponibilidad retrasó bastante el desarrollo del proyecto.

Las soluciones que se plantean son las siguientes:

1. trasladar un desarrollador de otro departamento al de Soluciones para sustituir la posición de Carlos Rodriguez.
2. Hacer que Carlos Rodriguez trabaje desde casa a un ritmo de eficiencia mucho menos ya que necesitaría reposo.
3. Retrasar la parte del proyecto en 7 días.
4. Asignar las tareas al otro programador, en este caso, al autor de esta memoria.

La primera opción no resultaría viable ya que se tendría el mismo problema, pero en otro departamento y no es lo que se busca. Se cubriría parcialmente el puesto que falta, pero eso llevaría a un retraso de la magnitud de unos días ya que se necesita entrar en concepto hasta poder seguir con el desarrollo.

La segunda opción es la viable y la elegida. Carlos Rodriguez puede seguir con el proyecto desde casa, aunque con una baja eficiencia. Es lo más conveniente ya que las otras opciones retrasan el proyecto mientras que ésta tiene un avance más lento pero seguro.

La tercera opción es muy poco probable para elegir ya que el proyecto se quedaría absolutamente sin ningún avance. Es cierto que no se tiene una fecha final de entrega, pero no se puede dejar de lado si hay otras posibilidades de seguir con el desarrollo por menores que sean.

Por ultimo la cuarta opción tampoco es viable porque generaría una carga de trabajo excesiva para Mihai Manea. Además, las tareas sobre las cuales trabaja Carlos tienen un

nivel de conocimiento superior al que Mihai Manea actualmente tiene, por lo tanto, no producirían ningún beneficio.

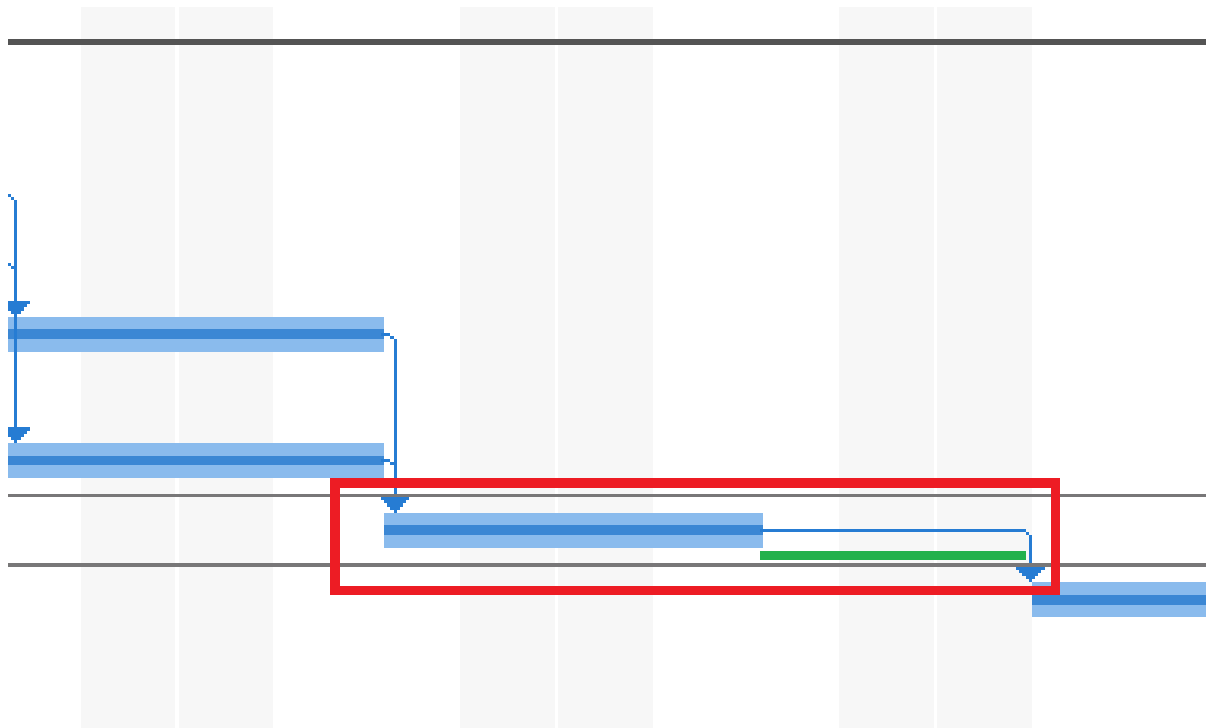


Figura 3.1: Cambios en las tareas retrasadas

3.8.2. Escenario 2: 90 % del proyecto completado

En el segundo escenario y pasado el problema inesperado de Carlos Rodriguez, el proyecto seguía con un buen desarrollo hasta llegar al porcentaje de 90%. Fue el día 01/05/2019 cuando se detecto un cambio en los requisitos del proyecto generando una pausa para determinar si realmente se consideraba como tal.

Efectivamente, el gerente confirmó que era necesario el cambio y ,por lo tanto, se necesitarían al menos 5 días para implementarlo en el proyecto y en el producto.

Ante este inconveniente se presentan las siguientes soluciones:

1. Contratar un programador adicional para ayudar con la tarea.
2. Retrasar el proyecto en 5 días.

La opción 1 no es viable porque por una parte se van a aumentar los gastos del proyecto más que si se elige la opción 2 y no es una buena elección.

Por otra parte, no sería conveniente ya que habría que familiarizar a los nuevos programadores con el entorno y el sistema.

La opción 2 es más viable, aunque retrase el proyecto. Es la única de las dos opciones que trae menores consecuencias en cuanto al ámbito económico y recursos humanos.

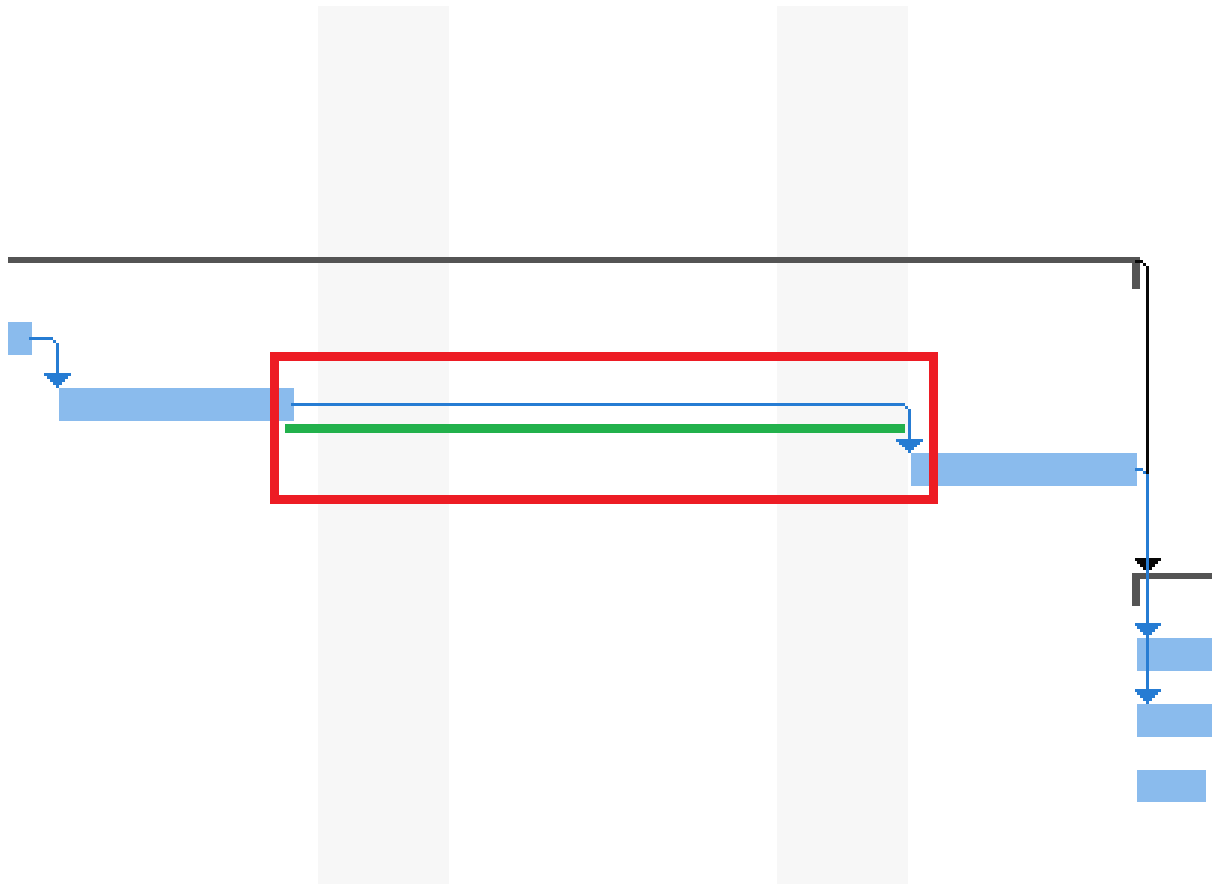


Figura 3.2: Cambios en las tareas retrasadas

Capítulo 4

Análisis y diseño del sistema

4.1. Análisis del sistema

La fase de análisis es una de las más importantes de todo el proyecto ya que convierte los requisitos establecidos en especificaciones que permiten diseñar el código a programar para que el sistema los implemente. Se va a representar mediante diseños visuales tales como diagramas, tablas, gráficos, etc.

Dentro de una empresa el análisis se refiere al proceso de mejorar la situación de la empresa y el funcionamiento mediante métodos y procesos.

4.1.1. Investigación y extracción de requisitos

En este apartado se van a usar diferentes técnicas para investigar y buscar información para poder determinar los requisitos.

No es una tarea difícil ya que principalmente consta de preguntar a los clientes qué desean que el producto haga, pero en nuestro caso hay un pequeño cambio. Hace referencia a que el contacto con los supuestos clientes no se ha llevado a cabo debido a qué: la solución ha surgido desde dentro de la empresa y no ha habido ninguna necesidad de toma de contacto externo.

Eso es un problema porque es posible que no tenga éxito a la hora de pasar la solución al mercado. Sin embargo, la definición de una buena parte de los requisitos se ha tomado a partir de las incidencias de otras soluciones.

A la investigación de requisitos se le suma la revisión del sistema de funcionamiento sobre el cual se debe tener conocimiento y control con el beneficio de determinar diferentes puntos débiles o fuertes.

De los documentos necesarios útiles para la identificación de datos de entrada y salida del sistema se han identificado los siguientes: documentos técnicos como por ejemplo el flujo de datos, el modelo de datos, informes, etc.

Sirve básicamente para tener constancia de qué es lo que tiene que hacer el sistema y compararlo con lo que se ha conseguido al final del proyecto.

También muestran la relación entre el comportamiento externo del sistema y la visión que los usuarios tienen del sistema que se va a desarrollar.

Las técnicas que pueden hacer una representación más entendible de los requisitos son las siguientes:

- La especificación de **casos de uso** que se complementa con el diagramas de casos de uso Unified Modeling Language (UML)[16].
- Los **requisitos de datos**.
- Las **historias de usuario**.

A continuación se van a definir los requisitos del proyecto:

1. El sistema debe detectar si un cliente está registrado, mediante un ID.
2. El sistema debe permitir añadir/modificar/borrar datos del sistema.
3. El sistema debe permitir guardar los datos hasta que finalice el proceso.
4. El sistema debe crear la configuración de ficheros para la nueva empresa.
5. El sistema debe crear el perfil de entorno de trabajo al finalizar el proceso.

A continuación se van a establecer los requisitos de datos para completar la especificación de los requisitos descritos anteriormente.

Requisitos de datos	
Código	RD01
Nombre	Cliente
Versión	1.0
Autores	Mihai Manea
Fuentes	Product owner
Datos específicos	Datos personales del cliente
Ocurrencia	Media de acceso media-alta Sin límite máximo
Importancia	Alta

Tabla 4.1: Plantilla requisitos de datos RD01

Requisitos de datos	
Código	RD02
Nombre	Business Central
Versión	1.0
Autores	Mihai Manea
Fuentes	Product owner
Datos específicos	Datos relacionados con el ERP base del cliente
Ocurrencia	Acceso por instalación de la solución
Importancia	Media-Alta

Tabla 4.2: Plantilla requisitos de datos RD02

Requisitos de datos	
Código	RD03
Nombre	Portal Innova
Versión	1.0
Autores	Mihai Manea
Fuentes	Product owner
Datos específicos	Datos que se le proporcionan al cliente
Ocurrencia	Se registra o se proporciona información al cliente
Importancia	Media-Alta

Tabla 4.3: Plantilla requisitos de datos RD03

Requisitos de datos	
Código	RD04
Nombre	Self Setup
Versión	1.0
Autores	Mihai Manea
Fuentes	Product owner
Datos específicos	Datos relacionados con la configuración de la nueva empresa
Ocurrencia	Configuración final en al solución
Importancia	Alta

Tabla 4.4: Plantilla requisitos de datos RD04

4.1.2. Diagrama de casos de uso

El diagrama de casos de uso es una representación visual en formato de diagramas más representativos de *UML* para la modelación de los contenidos referentes al comportamiento del sistema. En la Figura 4.1 podemos ver el diseño.

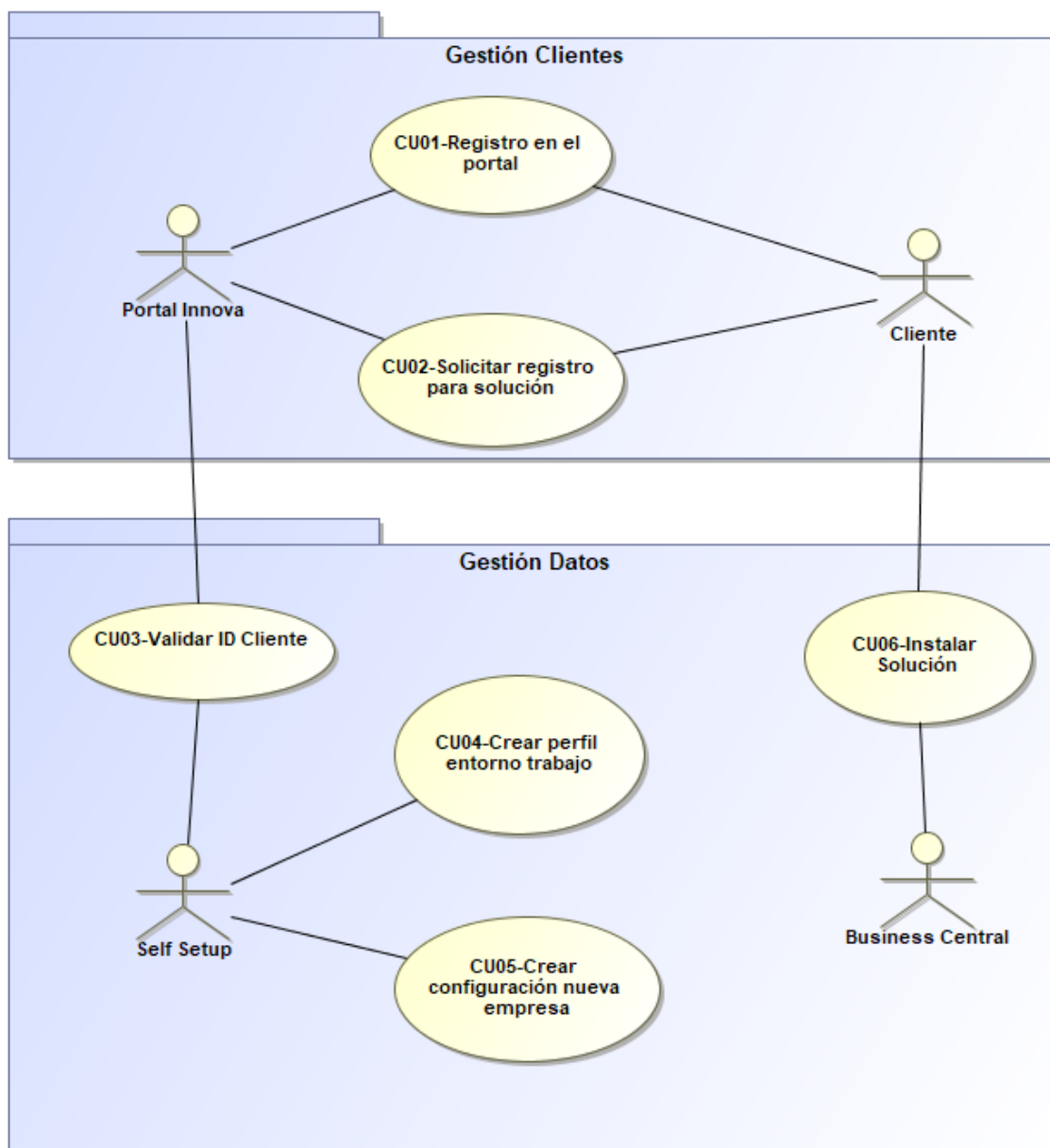


Figura 4.1: Diagrama Casos Uso

La especificación de casos de uso relacionados con el diagrama UML serán resumidos en tablas enumeradas desde el 4.5 hasta el 4.10.

Especificación de caso de uso	
Identificador	CU01
Nombre	Registro en el portal
Versión	1.0
Autores	Mihai Manea
Fuentes	Desarrollador
Descripción	El sistema tiene que permitir a los clientes registrarse
Alcance	Poder hacer uso de la solución
Nivel	Tarea
Actor principal	Cliente
Actores secundario	N/A
Relaciones	CU03-Validar ID cliente
Precondición	N/A
Condición fin con éxito	Registro satisfactorio en el portal
Condición fin con fracaso	Registro con datos erróneos en el portal
Trigger	Actualización del estado de los clientes
Secuencia normal	Acción
Excepciones	Excepción
1	El registro falla ya que se ha registrado un cliente con los mismos datos anteriormente.
Frecuencia esperada	Diaria
Importancia	Alta
Prioridad	A corto plazo
Comentarios	N/A

Tabla 4.5: Plantilla requisitos de datos para CU01

Especificación de caso de uso	
Identificador	CU02
Nombre	Solicitar registro para solución
Versión	1.0
Autores	Carlos Rodriguez
Fuentes	Desarrollador
Descripción	El sistema tiene que permitir a los clientes enviar una petición de registro al portal para poder recibir un enlace y tener acceso a la solución
Alcance	Poder hacer uso de la solución
Nivel	Tarea
Actor principal	Cliente
Actores secundario	N/A
Relaciones	CU03-Validar ID cliente
Precondición	N/A
Condición fin con éxito	Solicitud aceptada
Condición fin con fracaso	Solicitud denegada
Trigger	Actualización del estado de la petición
Secuencia normal	Acción
1	El cliente solicita registro a la solución
Excepciones	Excepción
1	La solicitud se rechaza ya que se ha solicitado 1 vez
Frecuencia esperada	Diaria
Importancia	Alta
Prioridad	A corto plazo
Comentarios	N/A

Tabla 4.6: Plantilla requisitos de datos para CU02

Especificación de caso de uso	
Identificador	CU03
Nombre	Validar ID Cliente
Versión	1.0
Autores	Self Setup
Fuentes	N/A
Descripción	El sistema tiene que comprobar que existe un cliente con el id proporcionado para poder continuar
Alcance	Terminar el proceso de configuración
Nivel	Subtarea
Actor principal	Cliente
Actores secundario	N/A
Relaciones	CU01-Registro en el portal CU02-Solicitar registro para solución
Precondición	Que exista un cliente registrado
Condición fin con éxito	Validación con éxito
Condición fin con fracaso	Validación con mensaje de error acerca del id del cliente
Trigger	Comprobar que existe un id del cliente
Secuencia normal	Acción
1	El sistema comprueba transparentemente el id del cliente
Excepciones	Excepción
1	id del cliente erróneo
Frecuencia esperada	Semanal
Importancia	Alta
Prioridad	A corto plazo
Comentarios	N/A

Tabla 4.7: Plantilla requisitos de datos para CU03

Especificación de caso de uso	
Identificador	CU04
Nombre	Crear perfil entorno trabajo
Versión	1.0
Autores	Self Setup
Fuentes	N/A
Descripción	El sistema tiene que crear un perfil específico a la configuración realizada
Alcance	Terminar el proceso de creación del perfil
Nivel	Tarea
Actor principal	Cliente
Actores secundario	N/A
Relaciones	CU03-Validar ID cliente
Precondición	Que exista un cliente registrado
Condición fin con éxito	Configurar correctamente el perfil del entorno de trabajo
Condición fin con fracaso	Configurar el perfil del entorno de trabajo por defecto
Trigger	Comprobar que existe un id del cliente y que los datos están guardados
Secuencia normal	Acción
1	El sistema comprueba transparentemente el id del cliente
Excepciones	Excepción
1	id del cliente erróneo
Frecuencia esperada	Semanal
Importancia	Alta
Prioridad	A corto plazo
Comentarios	N/A

Tabla 4.8: Plantilla requisitos de datos para CU04

Especificación de caso de uso	
Identificador	CU05
Nombre	Crear configuración nueva empresa
Versión	1.0
Autores	Self Setup
Fuentes	N/A
Descripción	El sistema tiene que crear la configuración definida por el cliente
Alcance	Crea la configuración correctamente
Nivel	Tarea
Actor principal	Cliente
Actores secundario	N/A
Relaciones	CU03-Validar ID cliente
Precondición	Que exista un cliente registrado
Condición fin con éxito	Configurar correctamente la estructura de la nueva empresa
Condición fin con fracaso	Configurar parcialmente la estructura de la nueva empresa
Trigger	Comprobar que existe un id del cliente y que los datos están guardados
Secuencia normal	Acción
1	El sistema comprueba transparentemente el id del cliente
Excepciones	Excepción
1	Que sean necesarios datos para la configuración y que estos no estén rellenos por el cliente
Frecuencia esperada	Semanal
Importancia	Alta
Prioridad	A corto plazo
Comentarios	N/A

Tabla 4.9: Plantilla requisitos de datos para CU05

Especificación de caso de uso	
Identificador	CU06
Nombre	Instalas solución
Versión	1.0
Autores	Cliente
Fuentes	N/A
Descripción	La solución se tiene que instalar en el ERP
Alcance	Instalar la solución correctamente
Nivel	Tarea
Actor principal	Cliente
Actores secundario	N/A
Relaciones	CU01-Registro en el portal CU02-Solicitar registro para solución
Precondición	Que exista un cliente con permisos para poder realizar la instalación
Condición fin con éxito	Instalación correcta en el ERP
Condición fin con fracaso	Instalación fallida en el ERP
Trigger	Realizar la petición a la tienda de soluciones
Secuencia normal	Acción
1	El cliente busca la solución y la instala
Excepciones	Excepción
1	Que el cliente no siga los pasos de la instalación
Frecuencia esperada	Semanal
Importancia	Media
Prioridad	A largo plazo
Comentarios	N/A

Tabla 4.10: Plantilla requisitos de datos para CU06

4.1.3. Historias de usuario

Las historias de usuario serán resumidas en tablas enumeradas desde el 4.11 hasta el 4.16.

HU01 - Instalar solución	
Descripción - Gestión de la solución	
HU01	Como cliente necesito poder instalar la solución en mi ERP para poder usarla
Feature	Instalar solución
Escenarios	
Escenario 1	Given que soy un cliente When intento instalar la solución en mi ERP Then la solución se instala correctamente
Escenario 2	Given que soy un cliente When intento instalar la solución en mi ERP Then se produce un error ya que no estoy registrado
Escenario 3	Given que soy un cliente When intento instalar la solución en mi ERP Then se produce un error ya que no he pagado la suscripción

Tabla 4.11: Plantilla historia de usuario HU01

HU02 - Registrarme en el portal	
Descripción - Gestión de clientes	
HU02	Como cliente necesito poder registrarme en el portal para adquirir la solución
Feature	Registro como cliente
Escenarios	
Escenario 1	Given que soy un cliente When intento registrarme en el portal Then el registro se realiza correctamente
Escenario 2	Given que soy un cliente When intento registrarme en el portal Then se produce un error ya que me he registrado previamente
Escenario 3	Given que soy un cliente When intento registrarme en el portal Then se produce un error ya que he introducido los datos requeridos mal

Tabla 4.12: Plantilla historia de usuario HU02

HU03 - Solicitar registro en el portal	
Descripción - Gestión de clientes	
HU01	Como cliente necesito poder solicitar el registro en el portal para poder adquirir la solución
Feature	Solicitar registro en el portal
Escenarios	
Escenario 1	Given que soy un cliente When intento solicitar registrarme en el portal Then la solicitud se realiza correctamente
Escenario 2	Given que soy un cliente When intento solicitar registrarme en el portal Then la solicitud no se realiza correctamente ya que no estoy registrado previamente
Escenario 3	Given que soy un cliente When intento solicitar registrarme en el portal Then la solicitud se realiza correctamente

Tabla 4.13: Plantilla historia de usuario HU03

HU04 - Configurar empresa	
Descripción - Gestión de datos	
HU01	Como cliente necesito poder modificar/añadir/borrar datos para configurar la nueva empresa
Feature	Configurar nueva empresa
Escenarios	
Escenario 1	Given que soy un cliente When intento añadir datos para configurar la empresa Then los datos se guardan satisfactoriamente
Escenario 2	Given que soy un cliente When intento borrar datos para configurar la empresa Then los datos se guardan satisfactoriamente
Escenario 3	Given que soy un cliente When intento modificar datos para configurar la empresa Then los datos se guardan satisfactoriamente
Escenario 4	Given que soy un cliente When intento añadir datos para configurar la empresa Then los datos no se guardan satisfactoriamente
Escenario 5	Given que soy un cliente When intento borrar datos para configurar la empresa Then los datos no se guardan satisfactoriamente
Escenario 6	Given que soy un cliente When intento modificar datos para configurar la empresa Then los datos no se guardan satisfactoriamente

Tabla 4.14: Plantilla historia de usuario HU04

HU05 - Crear perfil del entorno de trabajo	
Descripción - Gestión de datos	
HU01	Como cliente que realiza una configuración necesito que se cree un perfil específico para el entorno de trabajo
Feature	Crear perfil entorno
Escenarios	
Escenario 1	Given que soy un cliente When intento configurar la nueva empresa Then al finalizar se crea un perfil del entorno de trabajo relacionado con la nueva empresa
Escenario 2	Given que soy un cliente When intento configurar la nueva empresa Then al finalizar se crea un perfil del entorno de trabajo por defecto

Tabla 4.15: Plantilla historia de usuario HU05

HU06 - Validar Id	
Descripción - Gestión de datos	
HU01	Como cliente que realiza una configuración necesito que se me valide el id
Feature	Validar id cliente
Escenarios	
Escenario 1	Given que soy un cliente When intento configurar la nueva empresa Then la validación del id se realiza correctamente
Escenario 2	Given que soy un cliente When intento configurar la nueva empresa Then la validación del id no se realiza correctamente
Escenario 3	Given que soy un cliente When intento configurar la nueva empresa Then necesito proporcionar el id para que se valide correctamente

Tabla 4.16: Plantilla historia de usuario HU06

4.2. Diseño de la arquitectura del sistema

La arquitectura del sistema presenta una base sólida ya que el ERP de Microsoft es muy robusto. Los componentes que entran en juego a la hora del diseño son varios.

En un servidor de Azure están las funciones que consumen los datos de Self Setup para poder hacer todo el proceso de configuración.

En el portal de la empresa están todos los datos relacionados con los clientes que se van a necesitar en Self Setup para poder completar el proceso de configuración. Estos datos también serán tratados por funciones de Azure.

El ERP BC es la base sobre la cual va a estar presente la solución convirtiéndose así en el pilar principal de la arquitectura. Otro pilar secundario, que está desacoplado de BC, forma parte de la estructura y se va a fusionar como si fueran solo un sistema, es la solución en si.

Estos son los elementos que pertenecen a la arquitectura a nivel global. A continuación existen los *CodeUnits* que reflejan el diseño a nivel más interno.

Como se sabe de la definición anterior del concepto de *CodeUnit*, éstos se encargan de hacer el transporte de información para las páginas. La más importante, en el caso de Self Setup es la *CodeUnit SaveData* que se va a detallar en el capítulo siguiente.

En la Figura 4.2 se presenta el esquema de la arquitectura de sistema.

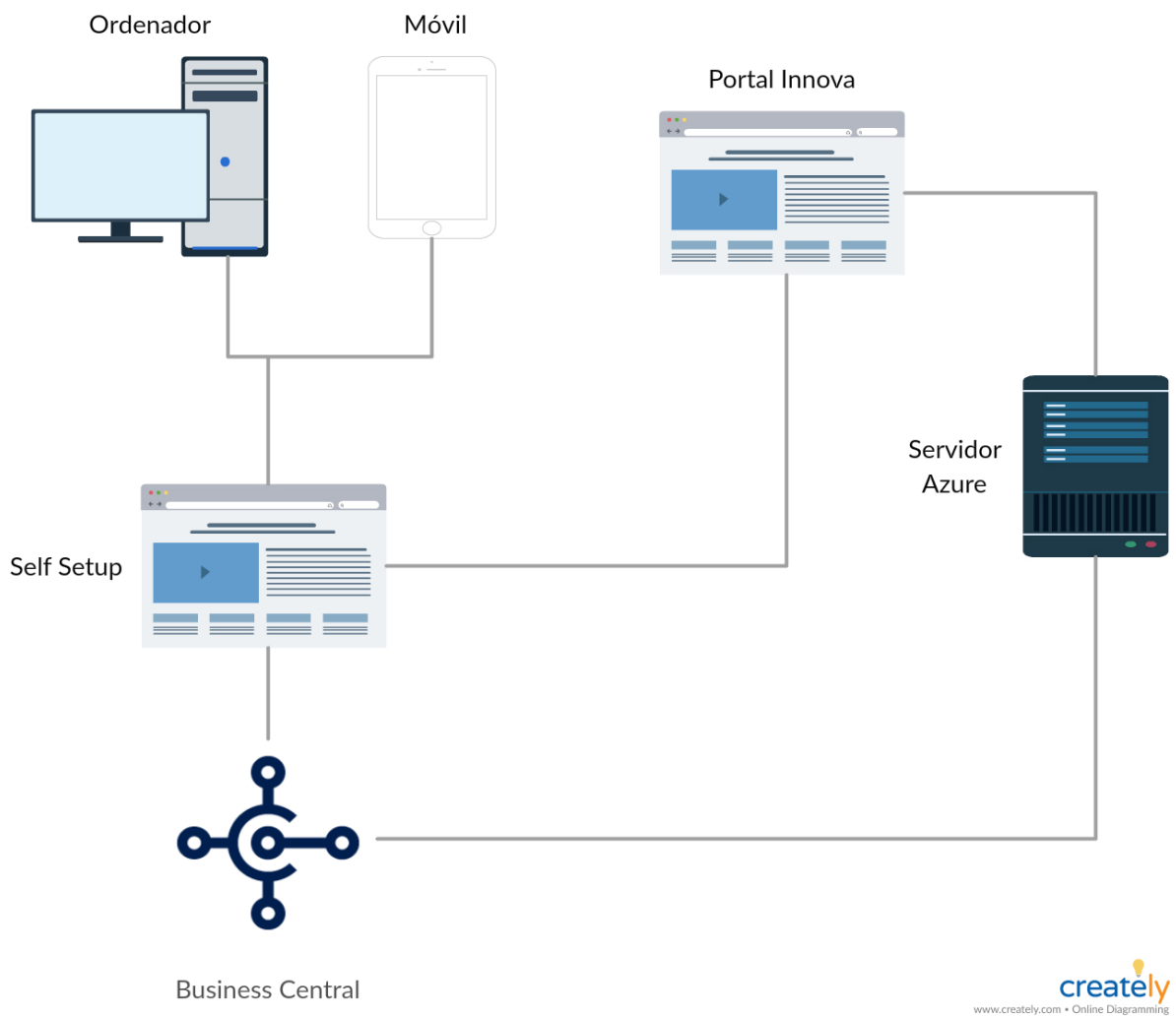


Figura 4.2: Esquema arquitectura Self Setup

4.3. Diseño de la interfaz

Un buen diseño de interfaces es lo que hace la diferencia entre un producto software de buena calidad y uno que no lo es. En este apartado se presentan las pantallas de configuración de la solución más importantes basadas en un diseño web.

Como se puede observar, en la Figura 4.3 hay un índice con todas las pantallas de configuración de Self Setup. En cada pantalla se pueden configurar diferentes parámetros pero debido a que hay bastantes solo se tratarán aquellas más importantes.

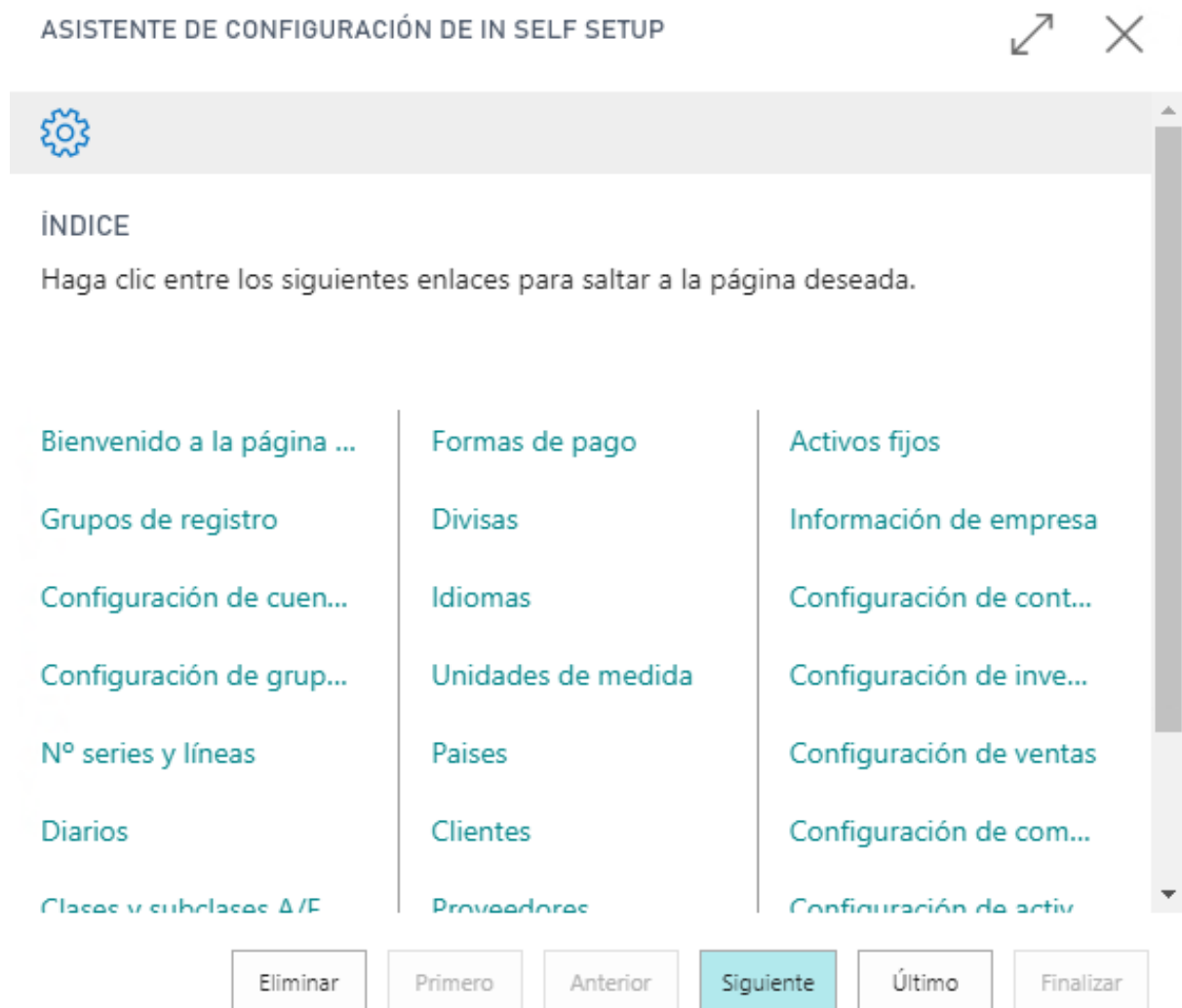


Figura 4.3: Página del índice

El trabajo del índice es poder proporcionar un acceso a cada pantalla mediante su *link* correspondiente. También existen botones en la parte inferior-derecha del asistente por la misma razón.

En la Figura 4.4 se muestra la primera página donde se tiene que añadir el **id** del cliente ya que es la pieza más importante del proceso de configuración. Sin él no se puede conseguir nada de la configuración para la nueva empresa y el proceso fallaría.

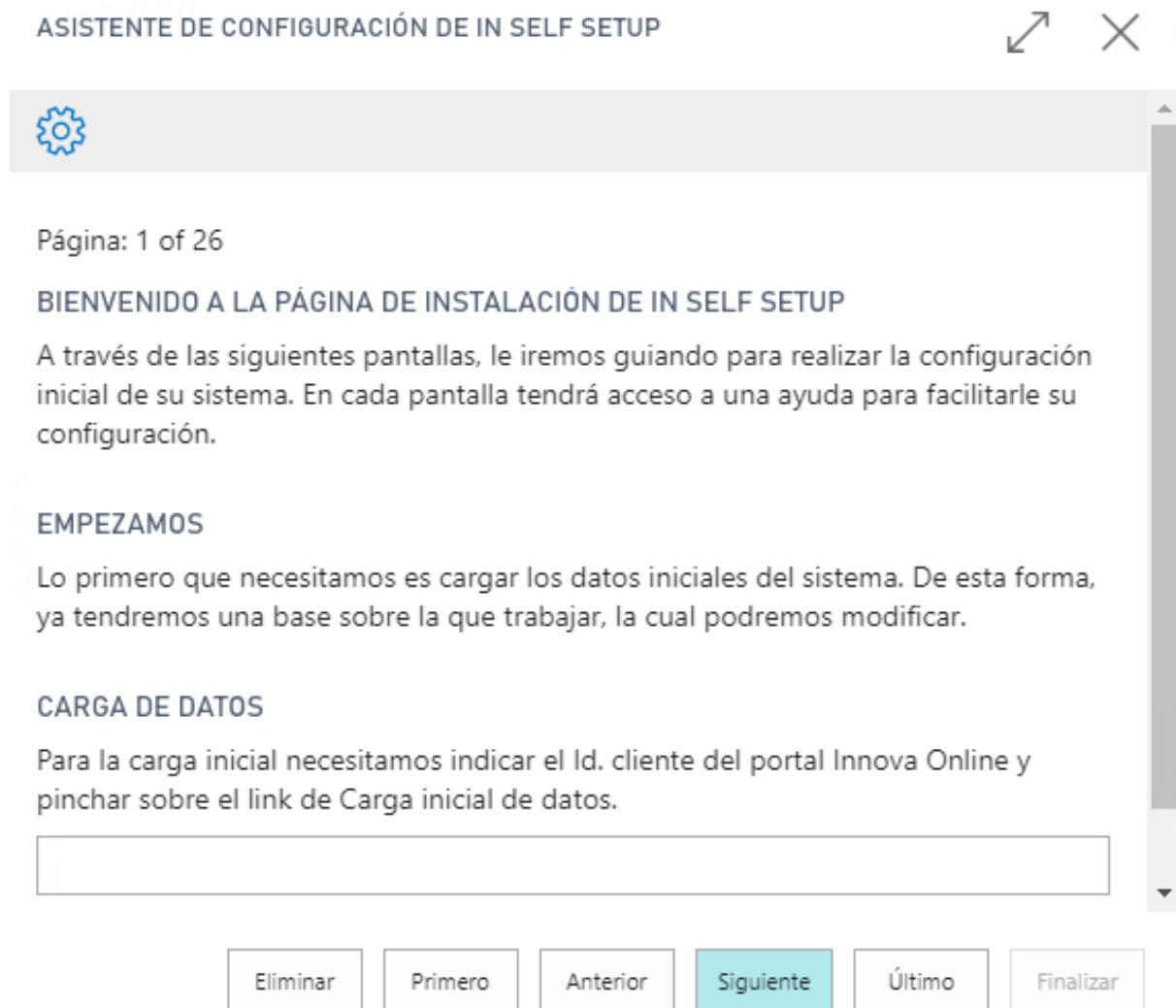


Figura 4.4: Pantalla de bienvenida

La información de la empresa se puede configurar rellenando los campos que se muestran en la Figura 4.5. El proceso es igual que las otras pantallas la única diferencia son los datos y la coherencia entre ellos. Dado que hay veces donde los datos a rellenar no dependen de ningún otro dato previo supone un menor riesgo de tener erratas.

ASISTENTE DE CONFIGURACIÓN DE IN SELF SETUP



Página: 20 of 26

INFORMACIÓN EMPRESA

Desde esta pantalla puede configurar los datos más importantes de la información empresa. Puede ver la ayuda en el siguiente link.

[Ayuda información empresa](#)

GENERAL

Nombre	<input type="text"/>
Dirección	<input type="text"/>
Población	<input type="text"/>
Código postal	<input type="text"/>
Cód. país/región	<input type="text"/>
CIF/NIF	<input type="text"/>
Código SIC de actividad	<input type="text"/>
E-mail	<input type="text"/>
Imagen	<input data-bbox="1182 1267 1217 1301" type="button" value="+"/>

Figura 4.5: Pantalla configuración de la empresa

La Figura 4.6 muestra la página final donde se detallan las acciones que se van a ejecutar así como los consejos antes de proceder a la finalización.

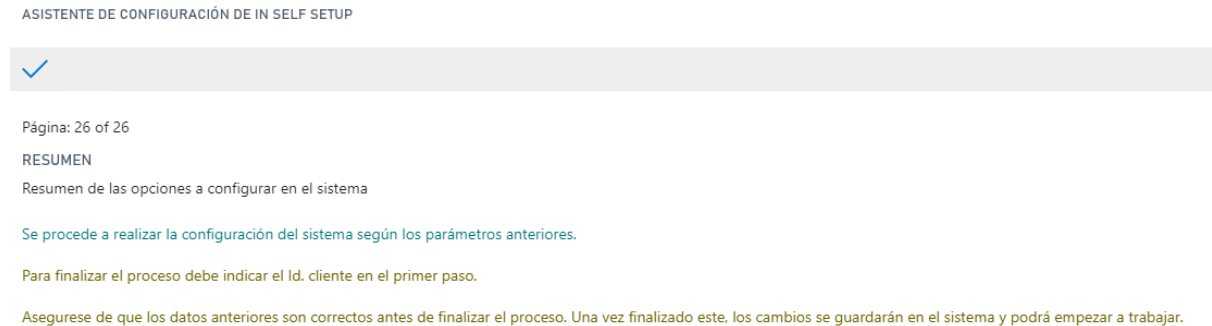


Figura 4.6: Pantalla configuración final

La Figura 4.7 contiene la imagen con todas las extensiones/soluciones del ERP Business Central entre las cuáles está instalada Self Setup. Desde la misma pantalla se puede gestionar cada extensión/solución por separado.

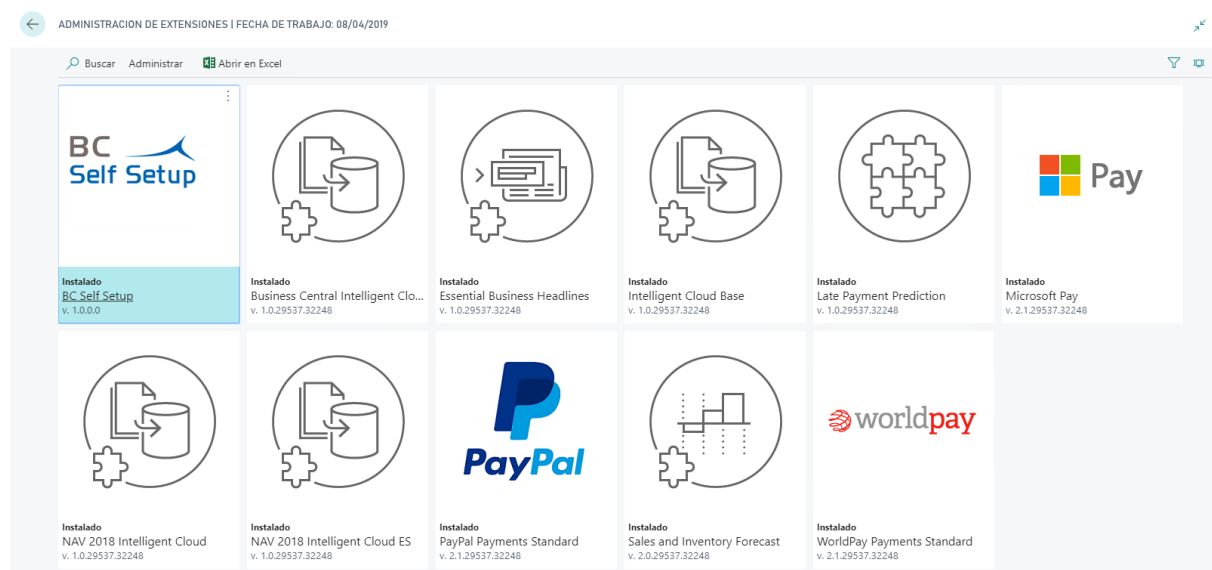


Figura 4.7: Pantalla de las extensiones

4.4. Diagrama de clases

Los diagramas de clases son los componentes que sirven para hacer una representación visual tanto de la información de la estructura del sistema como también del comportamiento mediante las operaciones de cada clase.

Es uno de los diagramas más importantes debido a que permite mostrar detalladamente las propiedades de sus elementos, las operaciones todas unidas en un objeto, las clases, las relaciones, paquetes, etc.

El modelado está formado por las clases del Cliente, Administrador, Business Central y Self Setup que se pueden ver en la Figura 4.8.

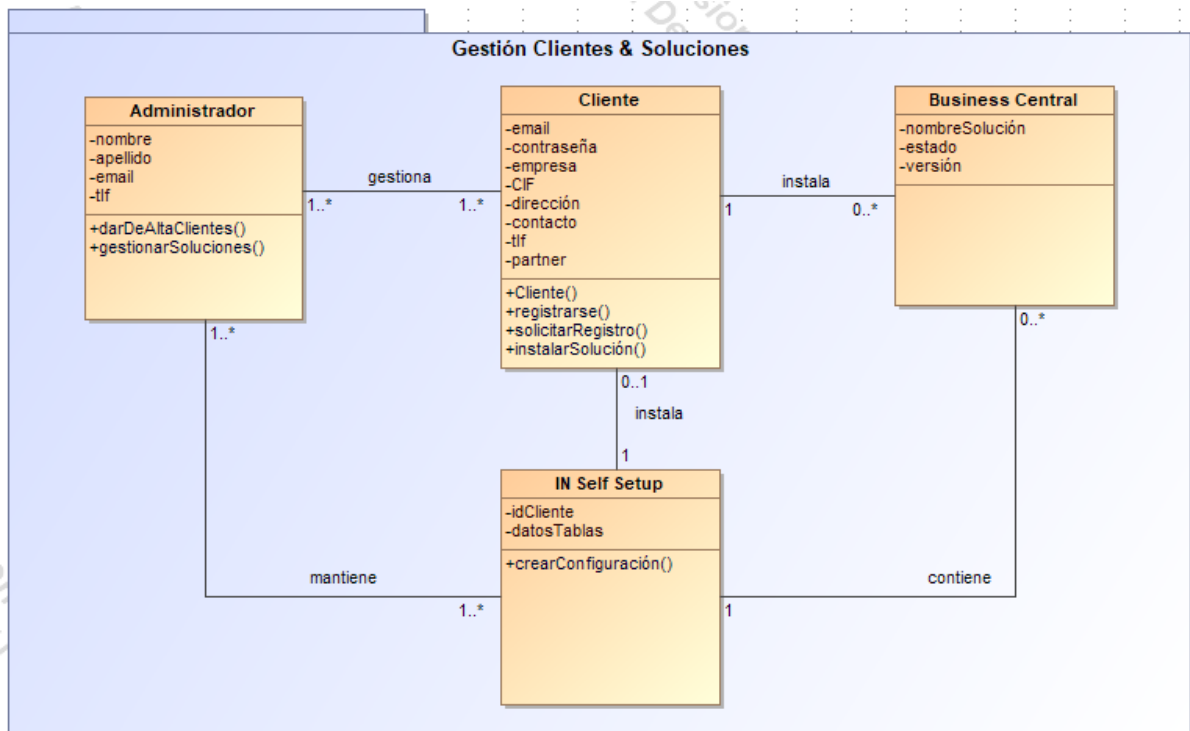


Figura 4.8: Diagrama de clases

Capítulo 5

Implementación y pruebas

En el capítulo de implementación y pruebas se presenta, por una parte, el código que hacer posible el funcionamiento de la solución, cómo están estructurados a nivel más bajo los componentes que intervienen en el proceso, aquellas *CodeUnits* que se encargan de los procesos, etc.

Por otra parte están las pruebas que no han tenido una especial dedicación. Obviamente se han hecho pruebas pero a nivel de desarrollador y éstas no han sido documentadas porque en el proceso que las pruebas tenía que realizar no existía un margen de error grande.

Las tarea principal de las *CodeUnits* es de guardar datos y por eso, en una gran parte, los datos vienen correctos porque los proporciona la solución al usuario final y no es el usuario quien tiene que introducirlos.

5.1. Detalles de implementación

Para entender mejor cómo está formado Business Central y saber qué componentes se utilizan por debajo cuando los datos se transmiten, se presenta un diseño[17] en la Figura 5.1.

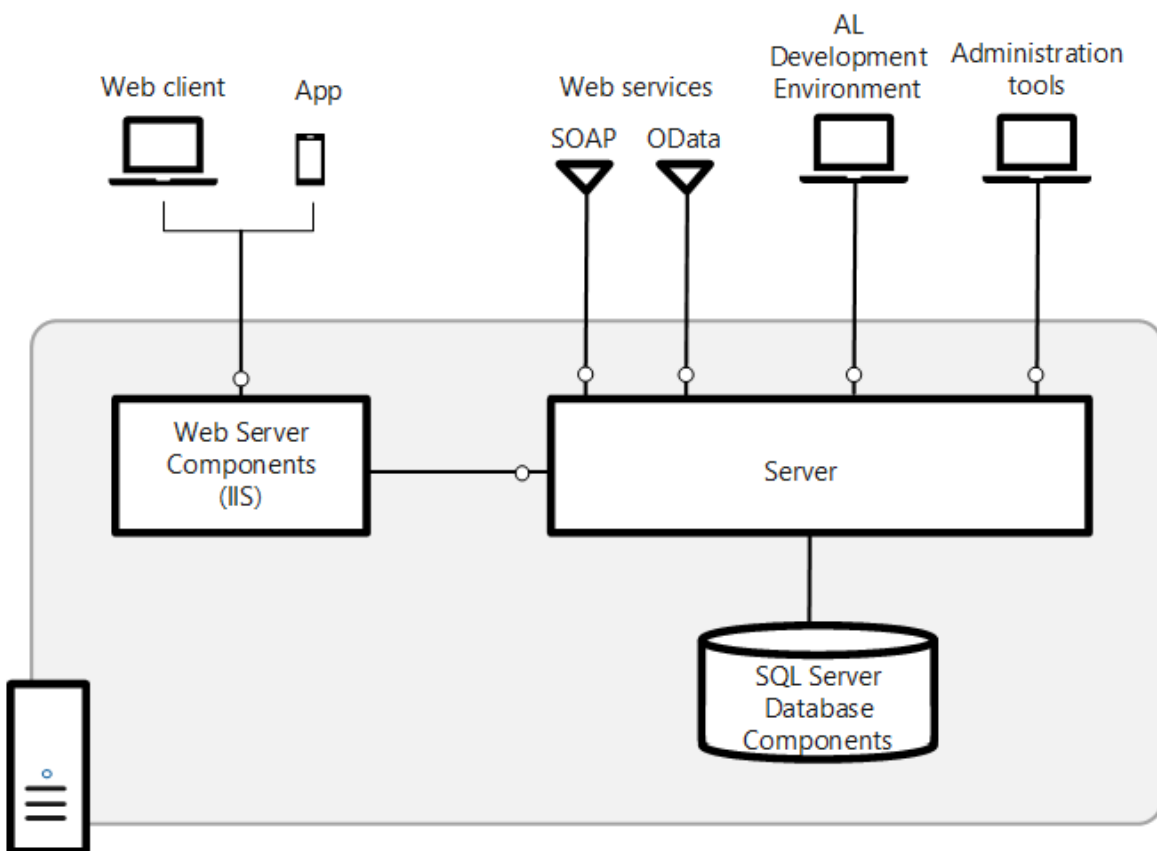


Figura 5.1: Componentes Business Central

A continuación se van a definir los componentes de la Figura 5.1 para que se entienda mejor el contexto.

1. **SQL Server:** puede estar separada en la parte de datos y la parte de la aplicación. Varios *tenants* pueden utilizar la aplicación. Un *tenant* es una instancia de uso.
2. **Server:** es quien gestiona la comunicación entre el cliente y la base de datos para controlar temas de autenticación, registro de eventos, tareas programadas, etc. Se basa en el entorno de trabajo .NET.
3. **Web Server:** es un Internet Information Service (IIS) que contiene elementos de la parte del servidor de Business Central para que den acceso desde la parte web del mismo. Un IIS es un servidor web creado para Microsoft Windows. Permite la publicación de páginas web desde cualquier ordenador que tenga instalado este servicio.
4. **Business Central App:** la aplicación en sí que puede ser usada desde varios dispositivos.
5. **Web Services:**
 - **SOAP**[18]: es un protocolo de intercambio de datos en formato Extensible Markup Language (XML).
 - **OData Web Service**[19]: habilita el consumo de Application Programming Interfaces (APIs)[20] que permiten a los clientes publicar y editar recursos web.

La *CodeUnit* llamada *SaveData* que forma parte de la solución se encarga básicamente de guardar los datos que el cliente edita en la web. La trayectoria que sigue pasa por diferentes puntos de control desde su inicio hasta su finalización.

La Figura 5.2 muestra todos los métodos que intervienen a la hora del guardado de datos.

```

trigger OnRun()
begin
    CheckSaveDataInPortal();

    if GuiAllowed() then
        SetAccountingPeriods();

    if GuiAllowed() then
        Window.Open(Text008Lbl + '\' + Text005Lbl + '\' + Text006Lbl + '\' + Text007Lbl);

    if CanExecute() then begin
        SaveVATBusPostingGroups();
        SaveVATProdPostingGroups();
        SaveGenBusPostingGroups();
        SaveGenProdPostingGroups();
        SaveAccounts();
        SaveVATPostingSetup();
        SaveGeneralPostingSetup();
        SaveCustomerPostingGroups();
        SaveVendorPostingGroups();
        SaveInventoryPostingGroups();
        SaveInventoryPostingSetup();
        SaveFAPostingGroups();
        SaveBankAccountPostingGroups();
        SaveNoSeriesAndLines();
        SaveGenJournalTemplates();
        SaveGenJournalBatches();
        SaveFAJournalTemplates();
        SaveFAJournalBatches();
        SavePaymentTerms();
        SavePaymentMethods();
        SaveCurrencies();
        SaveLanguages();
        SaveUnitsOfMeasure();
        SaveCountryRegions();
        SavePostCodes();
        SaveFAClass();
        SaveFASubclass();
        SaveDepreciationBooks();

        SaveSetup();
    end;
end;

```

Figura 5.2: CodeUnit SaveData

En la Figura 5.3 se presenta un ejemplo detallado de un método que toma los diferentes elementos que forman las tablas *VATBusPostingGroup* y *VATBusPostingGroupBase* para luego guardarlos en la base de datos.

Todo el proceso de guardado lo hace el propio Business Central.

```
1 reference
local procedure SaveVATBusPostingGroups()
var
    VATBusPostingGroup: Record "VAT Business Posting Group";
    VATBusPostingGroupBase: Record INNSSEVATBusPostingGroup;
begin
    if GuiAllowed() then
        Window.Update(1, VATBusPostingGroupBase.TableCaption());
    Index := 0;

    VATBusPostingGroupBase.SetRange(INNSSESavedInDB, false);
    Cont := VATBusPostingGroupBase.CountApprox();
    if VATBusPostingGroupBase.FindSet(true, false) then
        repeat
            VATBusPostingGroup.Init();
            VATBusPostingGroup.Code := VATBusPostingGroupBase.INNSSECode;
            VATBusPostingGroup.Description := VATBusPostingGroupBase.INNSSEDescription;
            VATBusPostingGroup.Id := CreateGuid();
            if VATBusPostingGroup.Insert(true) then begin
                VATBusPostingGroupBase.INNSSESavedInDB := true;
                VATBusPostingGroupBase.Modify();
            end;

            Index += 1;
            if GuiAllowed() then begin
                Window.Update(2, Index);
                Window.Update(3, Cont);
            end;
        until VATBusPostingGroupBase.Next() = 0;
end;
```

Figura 5.3: Método local para guardar los grupos contables

5.2. Verificación y validación

Para verificar si los objetivos generales están dentro del alcance del proyecto cada semana se ha establecido una reunión con los responsables del desarrollo de la solución. En la reunión se ha comprobado si los requisitos definidos al principio concuerdan con los que se definieron antes de crear la solución.

A medida que se ejecutaban este tipo de comprobaciones se han encontrado errores en el código relacionados con desviaciones del alcance y por lo tanto se han tenido que subsanar. Debido a que se estaban realizando verificaciones periódicas no se han generado errores graves.

En este caso, al determinar que la documentación acerca de proyecto está correcta solo hace falta hacer una validación con los clientes, pero no se llega a ese punto hasta que la solución se publica.

Hay que mencionar que antes de su publicación en la tienda de Microsoft (*AppSource*) se hacen unas pruebas unitarias para determinar si la solución es correcta y si acepta los requisitos de la tienda. Si las pruebas son favorables, el técnico que las realiza las válida para poder continuar.

En el caso de la validación acerca de los casos de uso, se cumple que hay al menos uno que representa la funcionalidad del sistema.

Capítulo 6

Conclusiones

Las conclusiones que se van a presentar en este capítulo van a ser de tres tipos: formativos, profesionales y personales.

6.1. Ámbito formativo

A nivel formativo el punto más a favor durante la estancia en prácticas ha sido el aprendizaje constante de todas las herramientas utilizadas para crear la solución como por ejemplo Visual Studio Code, Visual Studio, Object Designer, etc.

Al tener un supervisor, Carlos Rodriguez, con una experiencia abundante ha hecho que me sienta más seguro con la formación recibida al largo de la estancia.

En momentos puntuales he tenido la posibilidad de tratar con desarrolladores de los otros departamentos mediante reuniones que tenían como objetivo el estudio general acerca de Self Setup. Además se han tratado otras posibles implementaciones de la solución en países como, por ejemplo, México.

6.2. Ámbito profesional

En cuanto al nivel profesional hay que destacar que la empresa tiene un equipo muy bien dotado, con mucha experiencia en el sector de los sistemas de información y el punto más predominante es el trabajo en equipo.

La empresa ofrece *webinarios* que son formaciones a nivel *online* impartidos por el personal de cada departamento para aquellas personas interesadas en aprender y para las

que están usando la solución actualmente.

Las formaciones por parte de la empresa son totalmente sin cargo alguno ya que se intenta mantener una buena conducta entre el cliente y la empresa.

6.3. **Ámbito personal**

Personalmente creo que he adquirido bastantes conocimientos tanto teóricos como prácticos. El proceso de aprendizaje en el ámbito en el cual trabaja la empresa es bastante largo ya que se necesitan conocimientos de programación a un nivel que se enfoca a las herramientas de Microsoft.

Por otra parte, para poder desarrollar ciertas soluciones es necesario tener conocimientos de finanzas para desarrollar correctamente el código y tener claros los objetivos de la solución.

Finalmente tengo que añadir que ha sido un una experiencia agradable por el compañerismo, por el ambiente general de mi departamento y también por las nuevas instalaciones en la cuales he emprendido mi trabajo. No he tenido ninguna falta que tenga que ver con el material de *hardware* o de *software*.

Bibliografía

- [1] “Página innova consulting.” <https://www.innovaconsulting.es/>. [Consulta: 21 de Julio de 2019].
- [2] “Enterprise resource planning.” <https://www.oracle.com/applications/erp/what-is-erp.html>. [Consulta: 21 de Julio de 2019].
- [3] “Business central.” <https://dynamics.microsoft.com/en-us/business-central/overview/>. [Consulta: 21 de Julio de 2019].
- [4] “Visual studio ide.” <https://visualstudio.microsoft.com/vs/>. [Consulta: 21 de Julio de 2019].
- [5] “Microsoft azure.” <https://azure.microsoft.com/en-us/>. [Consulta: 21 de Julio de 2019].
- [6] “App source.” <https://appsource.microsoft.com/en-us/>. [Consulta: 21 de Julio de 2019].
- [7] “Visual studio code.” <https://code.visualstudio.com/>. [Consulta: 21 de Julio de 2019].
- [8] “Git.” <https://git-scm.com/>. [Consulta: 21 de Julio de 2019].
- [9] “Timetracker.” https://appsource.microsoft.com/en-us/product/dynamics-365-business-central/PUBID.pbsi_software%7CAID.247timetracker%7CPAPPID.9a12247e-8564-4b90-b80b-cd5f4b64217e?tab=Overview. [Consulta: 21 de Julio de 2019].
- [10] “Visual studio .net.” <https://visualstudio.microsoft.com/vs/features/net-development/>. [Consulta: 21 de Julio de 2019].
- [11] “Software development kit.” <https://techterms.com/definition/sdk>. [Consulta: 21 de Julio de 2019].
- [12] “Magic draw uml editor.” <https://www.magicdraw.com/main.php>. [Consulta: 21 de Julio de 2019].
- [13] “Tenant.” <https://docs.microsoft.com/en-us/azure/active-directory/develop/quickstart-create-new-tenant>. [Consulta: 21 de Julio de 2019].

- [14] “Página innova.” <http://www.innovaonline.es>. [Consulta: 21 de Julio de 2019].
- [15] “Diagrama de gantt.” https://es.wikipedia.org/wiki/Diagrama_de_Gantt. [Consulta: 21 de Julio de 2019].
- [16] “Uml.” https://en.wikipedia.org/wiki/Unified_Modeling_Language. [Consulta: 21 de Julio de 2019].
- [17] “Componentes business central.” <https://docs.microsoft.com/en-us/dynamics365/business-central/dev-itpro/deployment/product-and-architecture-overview>. [Consulta: 21 de Julio de 2019].
- [18] “Simple object access protocol.” https://es.wikipedia.org/wiki/Simple_Object_Access_Protocol. [Consulta: 21 de Julio de 2019].
- [19] “Open data protocol.” https://en.wikipedia.org/wiki/Open_Data_Protocol. [Consulta: 21 de Julio de 2019].
- [20] “Application programming interface.” https://en.wikipedia.org/wiki/Application_programming_interface. [Consulta: 21 de Julio de 2019].

Anexo A

Diagrama de Gantt

El siguiente apartado presenta, en la Figura A.1, el diagrama de Gantt con todas sus tareas y en la Figura A.2 la relación que existe entre las tareas y el tiempo en forma gráfica.

	i	Task Mode	Nombre de tarea	Duration	Predecessors	Start	Finish
1			BC Self Setup	121,5 days		Tue 19/02/19	Wed 06/11/19
2			Desarrollo Planificación	121 days		Tue 19/02/19	Tue 05/11/19
3			Inicio	6 days		Tue 19/02/19	Mon 04/03/19
4			Revisar definición proyecto	2 days		Tue 19/02/19	Fri 22/02/19
5			Revisar requisitos del proyecto	2 days	4	Fri 22/02/19	Tue 26/02/19
6			Definir metodología de trabajo	2 days	5	Wed 27/02/19	Mon 04/03/19
7			Planificación	36 days	3	Mon 04/03/19	Mon 20/05/19
8			Gestión del alcance	4 days	6	Mon 04/03/19	Tue 12/03/19
9			Gestión de riesgos	4 days	8	Tue 12/03/19	Wed 20/03/19
10			Gestión de tiempo	4 days	8	Tue 12/03/19	Wed 20/03/19
11			Gestión de recursos humanos	3 days	9;10	Wed 20/03/19	Tue 26/03/19
12			Gestión de costes	5 days	9;10	Wed 20/03/19	Mon 01/04/19
13			Gestión de calidad	5 days	11;12	Mon 01/04/19	Wed 10/04/19
14			Gestión de seguimiento	15 days	13	Thu 11/04/19	Mon 13/05/19
15			Gestión de cierre	3 days	14	Tue 14/05/19	Mon 20/05/19
16			Análisis	16 days	7	Mon 20/05/19	Mon 24/06/19
17			Diagrama de clases	6 days	7	Mon 20/05/19	Fri 31/05/19
18			Diagramas de actividades	5 days	17	Mon 03/06/19	Wed 12/06/19
19			Validación y verificación del análisis	5 days	18	Wed 12/06/19	Mon 24/06/19
20			Diseño	8 days		Mon 24/06/19	Wed 10/07/19
21			Diseño de interfaces	5 days	19	Mon 24/06/19	Wed 03/07/19
22			Diseño relación EPR y módulo Solución	3 days	21	Thu 04/07/19	Wed 10/07/19
23			Implementación	20 days	16	Wed 10/07/19	Thu 22/08/19
24			Creación relación entre ERP y módulo Solución	4 days	22	Wed 10/07/19	Thu 18/07/19
25			Interfaces de los módulos	14 days	24;21	Fri 19/07/19	Mon 19/08/19
26			Crear paquete de la solución	2 days	25;22	Mon 19/08/19	Thu 22/08/19
27			Pruebas	15 days	23	Thu 22/08/19	Tue 24/09/19

Figura A.1: Tareas del diagrama

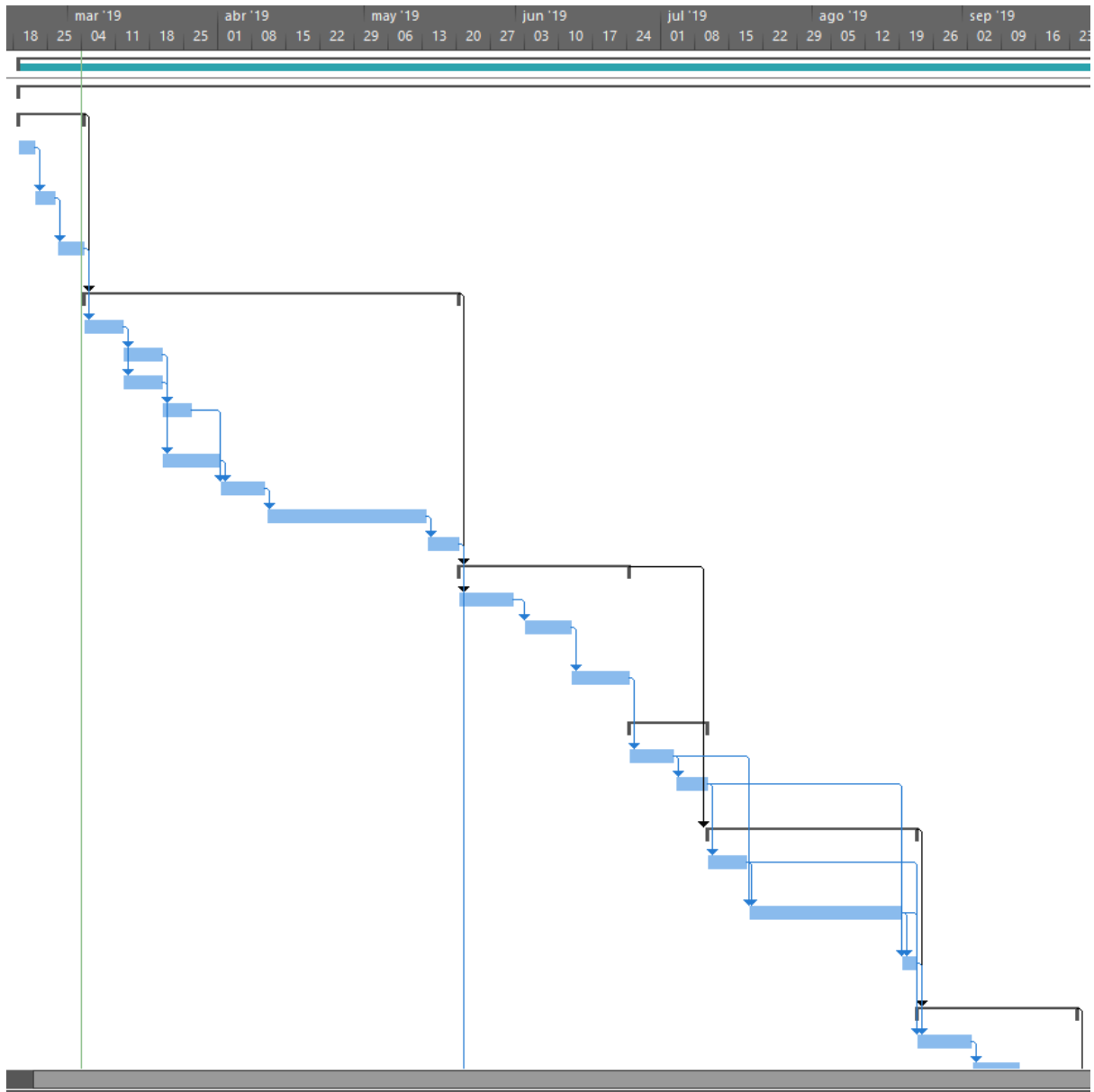


Figura A.2: Tareas del diagrama en formato gráfico