



GRADO EN INGENIERÍA INFORMÁTICA

TRABAJO DE FINAL DE GRADO

---

**Gestión de ficheros localizados en  
SharePoint para Microsoft Dynamics 365  
Business Central**

---

*Autor:*  
Óscar MONLLEÓ RECATALÁ

*Supervisor:*  
Sergi VILAR DOMENECH  
*Tutor académico:*  
Angel Pascual DEL POBIL FERRÉ

Fecha de lectura: 27 de Junio de 2024  
Curso académico 2023/2024

## **Resumen**

Este documento ofrece un resumen del proyecto desarrollado para integrar Microsoft Business Central con SharePoint Online, facilitando la gestión documental de la empresa. El proyecto ha sido diseñado para permitir el almacenamiento y acceso eficiente a documentos digitales a través de la interfaz de Business Central, utilizando SharePoint Online como plataforma principal de almacenamiento.

La funcionalidad principal del sistema incluye la carga de documentos desde diferentes entidades de Business Central, con opciones de selección directa o arrastre y soltado de archivos, que después se almacenarán en SharePoint. Cada documento cargado requiere la especificación de un tipo, que facilita su gestión y recuperación. Además, se ha implementado un factbox en las páginas de entidades seleccionadas, para mostrar y acceder a los documentos vinculados directamente desde SharePoint.

La configuración del sistema permite adaptar la estructura de carpetas en SharePoint de acuerdo con las necesidades específicas de cada entidad, lo cual, se puede ajustar mediante una tabla de mapeo. Este diseño también soporta la carga masiva de documentos, optimizando la gestión de grandes volúmenes de archivos y su asociación con los registros correspondientes en Business Central.

Este proyecto ha sido implementado inicialmente para las entidades de clientes, proveedores, contactos y facturas, con la posibilidad de extenderse a otras áreas del ERP, asegurando así una integración flexible y adaptativa a las diversas necesidades operativas y administrativas de la empresa.

## **Palabras clave**

Microsoft Business Central, ERP, SharePoint, Gestión de ficheros.

## **Keywords**

Microsoft Business Central, ERP, SharePoint, File Management.

# Índice general

<b>1. Introducción</b>	<b>11</b>
1.1. Contexto y motivación del proyecto . . . . .	11
1.2. Objetivo y alcance del producto . . . . .	11
1.2.1. Alcance funcional . . . . .	12
1.2.2. Alcance organizacional . . . . .	12
1.2.3. Alcance informático . . . . .	12
1.3. Objetivo y alcance del proyecto . . . . .	12
1.4. Descripción detallada del proyecto . . . . .	13
1.5. Estructura de la memoria . . . . .	14
<b>2. Planificación del proyecto</b>	<b>17</b>
2.1. Metodología . . . . .	17
2.2. Planificación temporal del proyecto . . . . .	18
2.3. Seguimiento del proyecto . . . . .	18
2.4. Costes . . . . .	21
2.4.1. Recursos humanos . . . . .	21
2.4.2. Software . . . . .	22
2.4.3. Hardware . . . . .	23
2.5. Riesgos . . . . .	23

2.5.1. Identificación de riesgos . . . . .	23
2.5.2. Análisis de riesgos . . . . .	24
2.5.3. Mitigación de riesgos . . . . .	24
2.5.4. Riesgos experimentados . . . . .	25
<b>3. Análisis del sistema/producto</b>	<b>27</b>
3.1. Definición de requisitos . . . . .	27
3.2. Análisis de requisitos . . . . .	28
<b>4. Diseño del sistema/producto</b>	<b>31</b>
4.1. Diseño de la base de datos . . . . .	31
4.2. Diseño de la arquitectura del sistema/producto . . . . .	32
4.3. Diseño de las interfaces . . . . .	33
<b>5. Implementación y pruebas</b>	<b>39</b>
5.1. Estructura del código . . . . .	39
5.2. Descripción técnica de la implementación . . . . .	41
5.3. Verificación y validación . . . . .	67
<b>6. Conclusiones</b>	<b>71</b>

# Índice de figuras

2.1. Diagrama de Gantt. . . . .	18
3.1. Diagrama de casos de uso. . . . .	28
4.1. Arquitectura de la base de datos. . . . .	31
4.2. Diseño base de datos relacional. . . . .	32
4.3. Arquitectura de tres capas. . . . .	33
4.4. Menú de navegación de Business Central. . . . .	34
4.5. SharePoint Setup. . . . .	34
4.6. Tabla de mapeo. . . . .	35
4.7. Tipos de documentos. . . . .	35
4.8. Funcionalidad de drag & drop. . . . .	36
4.9. Funcionalidad de clic. . . . .	36
4.10. Listado de ficheros en SharePoint. . . . .	36
4.11. Listado de funciones. . . . .	37
4.12. Estructura de carpetas en Sharepoint. . . . .	37
5.1. Estructura del proyecto. . . . .	40
5.2. Estructura directorio tests. . . . .	41
5.3. Tablas. . . . .	41
5.4. Tabla SharePoint Setup. . . . .	42

5.5. Tabla SharePoint Setup 2. . . . .	42
5.6. Tabla Mapping 1. . . . .	43
5.7. Tabla Mapping 2. . . . .	43
5.8. Tabla Mapping 3. . . . .	44
5.9. Tabla Mapping 4. . . . .	44
5.10. Tabla Mapping 5. . . . .	45
5.11. Tabla Mapping 6. . . . .	45
5.12. Tabla Mapping 7. . . . .	45
5.13. Tabla Links 1. . . . .	46
5.14. Tabla Links 2. . . . .	46
5.15. Tabla Links 3. . . . .	47
5.16. Tabla Links 4. . . . .	47
5.17. Tabla Links 5. . . . .	48
5.18. Tabla Links 6. . . . .	48
5.19. Tabla Document Type. . . . .	49
5.20. Páginas . . . . .	49
5.21. Página Document Type List. . . . .	50
5.22. Página Drag & Drop 1. . . . .	51
5.23. Página Drag & Drop 2. . . . .	51
5.24. Página Drag & Drop 3. . . . .	52
5.25. Página Links FactBox 1. . . . .	52
5.26. Página Links FactBox 2. . . . .	53
5.27. Página Links FactBox 3. . . . .	53
5.28. Página Links List 1. . . . .	54
5.29. Página Links List 2. . . . .	54

5.30. Página Links List 3. . . . .	55
5.31. Página Mapping 1. . . . .	56
5.32. Página Mapping 2. . . . .	56
5.33. Página Mapping 3. . . . .	57
5.34. Página SharePoint 1. . . . .	57
5.35. Página SharePoint 2. . . . .	58
5.36. Extensión de página. . . . .	58
5.37. Codeunit 1. . . . .	59
5.38. Codeunit 2. . . . .	60
5.39. Codeunit 3. . . . .	60
5.40. Codeunit 4. . . . .	61
5.41. Codeunit 5. . . . .	61
5.42. Codeunit 6. . . . .	62
5.43. Codeunit 7. . . . .	62
5.44. Codeunit 8. . . . .	63
5.45. Script.js 1. . . . .	64
5.46. Script.js 2. . . . .	64
5.47. Script.js 3. . . . .	65
5.48. Startup.js. . . . .	65
5.49. StyleSheet.css. . . . .	65
5.50. ControlAddin. . . . .	66
5.51. PermissionSet. . . . .	66
5.52. Event Create Link. . . . .	67
5.53. Event Delete Link. . . . .	67
5.54. Event Download Link. . . . .	68

5.55. Event Open Link. . . . .	68
5.56. Codeunit Test 1. . . . .	68
5.57. Codeunit Test 2. . . . .	69
5.58. Codeunit Test 3. . . . .	69
5.59. GitHub Actions CI/CD 1. . . . .	70
5.60. GitHub Actions CI/CD 2. . . . .	70

# Índice de cuadros

2.1. Costes de recursos humanos. . . . .	21
2.2. Costes de software. . . . .	22
2.3. Costes de hardware. . . . .	23
3.1. Definición caso de uso CU01 . . . . .	29
3.2. Definición caso de uso CU02 . . . . .	29
3.3. Definición caso de uso CU03 . . . . .	30
3.4. Definición caso de uso CU04 . . . . .	30



# Capítulo 1

## Introducción

### 1.1. Contexto y motivación del proyecto

Este documento presenta la memoria del Trabajo de Fin de Grado realizado durante el período de prácticas en Innova Advanced Consulting [2], empresa situada en Castellón de la Plana. Innova es una empresa especializada en consultoría y soluciones tecnológicas para sistemas ERP (Enterprise Resource Planning), en particular para Microsoft Dynamics 365 Business Central [6]. Innova cuenta con una amplia trayectoria y ha recibido diversos reconocimientos, como el Microsoft Partner Gold, por sus soluciones innovadoras y la alta competencia de su equipo técnico.

La motivación para el desarrollo de este proyecto surge de la necesidad de Innova Advanced Consulting de optimizar la gestión documental entre Microsoft Business Central y SharePoint Online [7].

El proyecto se centra en la creación de una interfaz integrada que facilite el manejo y acceso a documentos, con el objetivo de agilizar los procesos, reducir errores y mejorar la colaboración entre los departamentos y los socios comerciales. La solución permitirá a Innova ofrecer un valor añadido significativo a sus clientes mediante la mejora de la eficiencia operativa.

Los beneficios de este proyecto son claros: agilizar el proceso de gestión documental, reducir los tiempos de operación y proporcionar un sistema robusto para el almacenamiento seguro y accesible de documentos.

### 1.2. Objetivo y alcance del producto

El propósito de esta solución es mejorar el acceso y la organización de los documentos, automatizando el proceso de almacenamiento, recuperación y gestión de documentos en SharePoint directamente desde Business Central. Además, busca proporcionar una interfaz clara y eficiente que facilite a los usuarios la gestión de documentos y la colaboración entre departamentos.

### 1.2.1. Alcance funcional

- **Automatizar la carga de documentos:** Permitir a los usuarios cargar documentos desde Business Central a SharePoint Online de manera automática.
- **Gestionar el almacenamiento de documentos:** Configurar y gestionar la estructura de almacenamiento en SharePoint según las necesidades de cada departamento.
- **Facilitar el acceso y la recuperación de documentos:** Proporcionar interfaces en Business Central para visualizar y acceder a los documentos almacenados en SharePoint.
- **Integridad, seguridad y control de acceso:** Garantizar que los documentos solo sean accesibles por usuarios autorizados, manteniendo la conformidad con las políticas de seguridad de la empresa.

### 1.2.2. Alcance organizacional

- **Usuarios:** Empleados de Innova Advanced Consulting que gestionan y dependen de la documentación almacenada en SharePoint.

### 1.2.3. Alcance informático

- **Microsoft Business Central:** Actúa como el sistema ERP [9] principal donde se gestionan las operaciones empresariales.
- **SharePoint Online:** Sirve como plataforma de gestión documental donde se almacenan y gestionan los documentos.
- **Sistemas de autenticación:** Se integran con los sistemas de autenticación existentes para asegurar el acceso y la gestión segura de documentos.

## 1.3. Objetivo y alcance del proyecto

El objetivo principal de este proyecto es implementar una solución integrada que conecte Microsoft Business Central con SharePoint Online para automatizar y optimizar la gestión documental en Innova Advanced Consulting. Se busca diseñar un sistema que mejore la eficiencia, seguridad y accesibilidad de la documentación corporativa, facilitando la colaboración y productividad entre los distintos departamentos de la empresa.

El proyecto comienza con una fase de análisis y diseño, donde se estudian las necesidades actuales y los procesos de gestión documental de Innova Advanced Consulting.

Y concluye con la revisión final y comprobación del sistema garantizando que todas las funcionalidades requeridas están operativas y cumplen con las expectativas del cliente.

Este alcance asegura que el proyecto se mantenga enfocado en su objetivo principal, gestionando eficientemente los recursos y el tiempo para cumplir con las expectativas de Innova Advanced Consulting.

## 1.4. Descripción detallada del proyecto

Innova Advanced Consulting utiliza Microsoft Business Central como su ERP principal para gestionar procesos empresariales como finanzas, ventas y compras. Aunque BC es una herramienta poderosa y versátil, no ofrece capacidades nativas avanzadas para la gestión de documentos, que se integren de manera fluida con plataformas externas como SharePoint Online. Esto limita la capacidad de la empresa para gestionar eficientemente su creciente volumen de documentos digitales, afectando la accesibilidad y la seguridad de la información importante.

El proyecto tiene como objetivo mejorar la integración de Business Central con SharePoint Online. Para lograrlo, se llevan a cabo los siguientes pasos:

- **Análisis preliminar:** Estudio de las capacidades actuales de Business Central y las necesidades específicas de Innova Advanced Consulting en términos de gestión documental.
- **Diseño de la solución:** Propuesta de una arquitectura integrada que permita una gestión documental eficiente utilizando Business Central y SharePoint Online.
- **Implementación:** Desarrollo de la solución propuesta, asegurando la compatibilidad y la integración con las funcionalidades existentes de Business Central.
- **Pruebas y validación:** Realización de simulación de pruebas para verificar la funcionalidad, la seguridad y la integración del nuevo sistema.

Tecnologías y herramientas:

- **Microsoft Dynamics 365 Business Central:** Se utiliza como el ERP base. Es esencial para la gestión de los procesos de negocio y se integra con SharePoint para la gestión documental.
- **SharePoint Online:** Para el almacenamiento y gestión de documentos.
- **Visual Studio Code [11]:** Para el desarrollo de extensiones personalizadas en BC utilizando el lenguaje AL, facilitando la integración con SharePoint.
- **GitHub [1]:** Para el control de versiones y colaboración en el desarrollo de software. Se usa junto con GitHub Actions para automatizar pruebas y despliegues.

En la integración de Microsoft Business Central con SharePoint Online para la gestión documental, se emplean principalmente dos lenguajes de programación: AL y JavaScript, además de CSS para la estilización de la interfaz.

- **AL (Application Language):** AL es el lenguaje principal utilizado para desarrollar extensiones en Microsoft Dynamics 365 Business Central. Es un lenguaje de programación de alto nivel, diseñado específicamente para la personalización y extensión del ERP. En este proyecto, AL se utiliza para:
  - Crear objetos, páginas, y manejar la lógica de negocio en Business Central.
  - Integrar funcionalidades específicas que interactúan con SharePoint Online.
  - Desarrollar la funcionalidad back-end que facilita la carga y gestión de documentos entre Business Central y SharePoint.
- **JS (JavaScript):** JavaScript se utiliza para mejorar la interactividad de la interfaz de usuario en Business Central, específicamente para manejar eventos relacionados con la funcionalidad de arrastrar y soltar documentos. Este lenguaje permite:
  - Controlar eventos de usuario como arrastrar y soltar archivos dentro de una área específica en la interfaz de usuario (UI).
  - Manipular el DOM para modificar estilos y responder a interacciones del usuario, mejorando la experiencia del usuario final.
  - Codificar y decodificar información de archivos para su transferencia segura a través de la interfaz.
- **CSS (Cascading Style Sheets):** CSS se utiliza para definir el estilo y la presentación de la interfaz de usuario que soporta la funcionalidad de arrastrar y soltar, incluyendo:
  - Ajustes de diseño como el tamaño, color de fondo, opacidad, y la fuente.
  - Mejorar la visualización del área de arrastre y soltar con imágenes y estilos adecuados para hacerla más intuitiva y agradable al usuario.

## 1.5. Estructura de la memoria

- **Capítulo 1:** Introducción  
Este primer capítulo establece el marco fundamental del proyecto. Explica el contexto y la motivación detrás de la necesidad de integrar Microsoft Business Central con SharePoint Online para optimizar la gestión documental en Innova Advanced Consulting. Detalla los objetivos del producto y del proyecto, proporcionando una visión clara de lo que se espera lograr. Además, ofrece una descripción del desarrollo del proyecto, marcando el inicio de la memoria con una guía de su estructura organizativa.
- **Capítulo 2:** Planificación del proyecto  
Este capítulo detalla la planificación del proyecto, describe la metodología de gestión adoptada y el cronograma de actividades. Describe como se lleva a cabo el seguimiento del proyecto, presenta un desglose detallado de los costes asociados e identifica los riesgos potenciales con las estrategias de mitigación, estableciendo una base sólida para la gestión efectiva del proyecto.
- **Capítulo 3:** Análisis  
El tercer capítulo se centra en el análisis de requisitos, se definen y analizan tanto los requisitos funcionales como los no funcionales. Este análisis es esencial para asegurar que

el sistema desarrollado cumpla con las expectativas de los usuarios y las necesidades del negocio, facilitando el diseño adecuado del sistema.

- **Capítulo 4:** Diseño

Este capítulo muestra el diseño completo del sistema, incluyendo la arquitectura del sistema, el diseño de la base de datos y del software, y la interfaz de usuario. Explica las decisiones de diseño y cómo contribuyen a cumplir con los requisitos establecidos, proporcionando un enfoque coherente para la implementación efectiva del sistema.

- **Capítulo 5:** Implementación y pruebas

Este capítulo describe el proceso de implementación y las pruebas realizadas para verificar la funcionalidad y rendimiento del sistema. Cubre la estructura del código y las técnicas de implementación, así como las pruebas de verificación y validación que garantizan que el sistema cumple con los criterios de calidad y funcionamiento esperados.

- **Capítulo 6:** Conclusiones

El capítulo final reflexiona sobre el proyecto, evaluando su éxito en relación con los objetivos planteados inicialmente y el impacto en la organización. Se discuten las lecciones aprendidas y ofrece recomendaciones para futuros proyectos o mejoras del sistema actual.



## Capítulo 2

# Planificación del proyecto

### 2.1. Metodología

La metodología utilizada en el proyecto es la predictiva o en cascada, también conocida como secuencial o de ciclo de vida. Este método organiza las tareas de forma que el inicio de una depende de la finalización de su predecesora. Se elige esta metodología porque cada tarea del proyecto requiere de conocimientos y resultados específicos de las etapas anteriores para su ejecución.

La metodología predictiva es ideal para este proyecto dada la necesidad de una estructura clara y una secuencia definida, que son esenciales para la integración y la gestión de los archivos en la nube con Microsoft Dynamics 365 Business Central. Se utiliza el diagrama de Gantt para planificar y visualizar todas las fases del proyecto de forma secuencial, facilitando el seguimiento del progreso y la gestión de tiempos y recursos.

Fases del proyecto según la metodología en cascada:

- **Formación:** Al principio del proyecto, se realiza una formación sobre Dynamics 365 Business Central que proporciona los conocimientos necesarios para desarrollar el proyecto.
- **Planificación:** Una vez completada la formación, se elabora un plan detallado del proyecto, que incluye la definición de objetivos y la descomposición de tareas.
- **Análisis:** Esta fase se centra en comprender a fondo los requisitos del proyecto y preparar el terreno para las soluciones a diseñar.
- **Diseño y desarrollo:** Tras el análisis, se inicia el diseño y desarrollo del sistema, utilizando herramientas como Visual Studio y AL para construir el módulo requerido.
- **Pruebas:** Para finalizar el proyecto, se realizan una serie de pruebas que deben ser superadas para asegurar su funcionalidad y rendimiento.

Cada fase requiere la aprobación del supervisor del proyecto antes de avanzar a la siguiente, garantizando que cada entrega cumple con los estándares de calidad y funcionalidad requeridos.

## 2.2. Planificación temporal del proyecto

En esta parte del proyecto, se aborda la planificación temporal, la cual podemos ver ilustrada de forma gráfica en la Figura 2.1. Se opta por un diagrama de Gantt, ya que ofrece una visión intuitiva tanto de los costes temporales como de las fechas de ejecución de cada tarea.

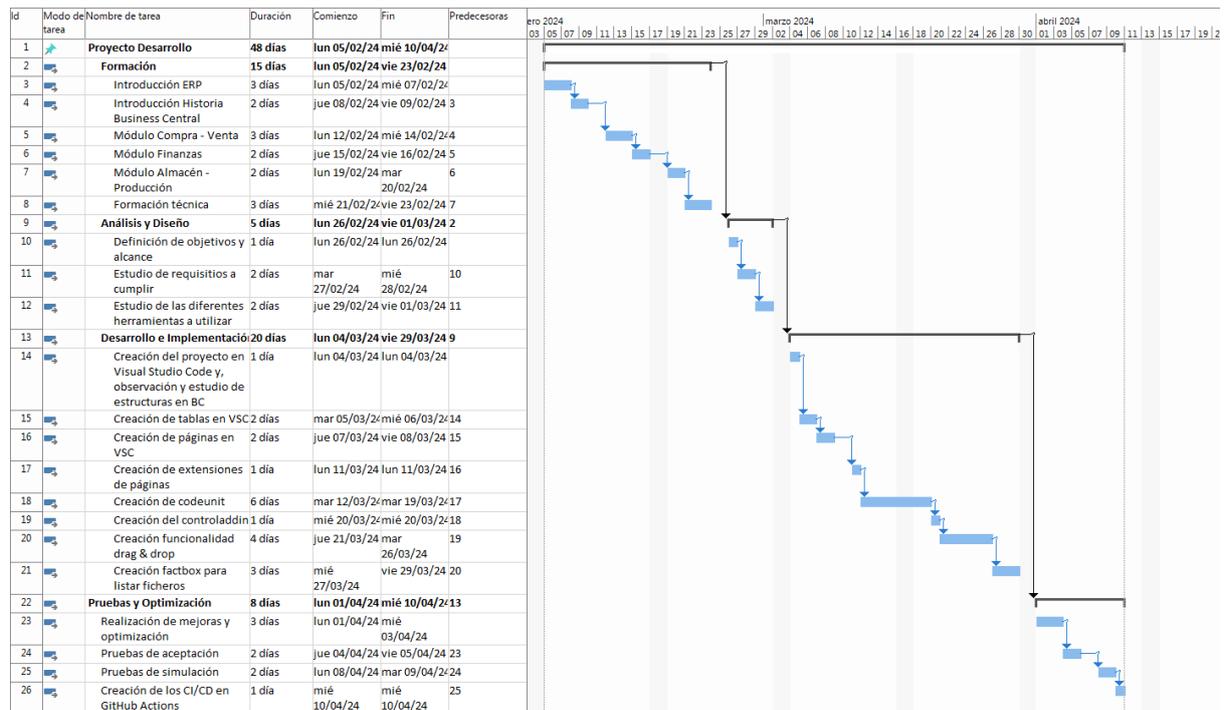


Figura 2.1: Diagrama de Gantt.

## 2.3. Seguimiento del proyecto

El seguimiento del proyecto es una componente esencial para garantizar la correcta ejecución y cumplimiento de los objetivos y plazos establecidos. Para ello, se establece un proceso de seguimiento consistente en la elaboración de informes detallados y la supervisión directa por parte del supervisor del proyecto.

Este proceso no solo asegura que todas las tareas se completen correctamente, sino que también permite identificar y abordar proactivamente cualquier desafío o desviación de los objetivos iniciales.

### **Primera quincena**

La primera quincena marca el inicio del proyecto con un período intensivo de formación en Microsoft Dynamics 365 Business Central. Principalmente, se centra en la comprensión de los conceptos fundamentales de un "Enterprise Resource Planning" (ERP) y en documentarse sobre las capacidades y funcionalidades de Business Central.

Durante las sesiones formativas, se explora en detalle varios módulos de Business Central. Uno de los módulos destacados es el módulo de 'Almacén', que permite al equipo entender y manipular el inventario, realizar movimientos de reclasificación, y configurar diferentes almacenes con sus características específicas. También se revisan los módulos de 'Compras' y de 'Ventas', examinando las diferencias entre albaranes, facturas y pedidos que gestiona Business Central, así como los distintos métodos de facturación. Para terminar con la formación, se estudia el módulo de 'Contabilidad'.

Al final de la primera quincena, se pasa a la parte práctica que empieza con la instalación de Docker para desplegar Business Central en el equipo.

### **Segunda quincena**

La segunda quincena del proyecto se centra en profundizar en la aplicación práctica de la formación recibida, trabajando directamente con la versión más reciente de Microsoft Dynamics 365 Business Central. Durante este período, el equipo lleva a cabo ejercicios prácticos que incluyen la creación de páginas y codeunits, consolidando así su comprensión del lenguaje de programación AL, utilizado ampliamente en Business Central.

Con la base práctica ya establecida, se empieza la fase de planificación y análisis del proyecto. Durante esta fase, se define con claridad los objetivos y se delimitan los requisitos específicos del proyecto según las necesidades del cliente. Este análisis permite la inicialización efectiva del proyecto dentro del entorno de desarrollo de Visual Studio.

Para comenzar con la implementación técnica, se realizan varios pasos en Visual Studio:

- Se descarga e instala Visual Studio, incorporando extensiones como AL CodeAction, AL Language Tools..., que son esenciales para el desarrollo de Business Central.
- Se inicia el proyecto utilizando "AL:GO", configurando el archivo 'launch.json' para sincronizar el entorno de desarrollo con el servidor de Business Central.
- Se utiliza "Download Symbols" para sincronizar y configurar el entorno de desarrollo con las bibliotecas estándar y extensiones instaladas en el servidor, asegurando una integración fluida y funcional.

### **Tercera quincena**

La tercera quincena del proyecto, se centra en el desarrollo estructural dentro de Microsoft Dynamics 365 Business Central, abordando específicamente la creación de las tablas y páginas necesarias para gestionar los archivos en la nube.

Comienza con la implementación de varias tablas cruciales que funcionan como la estructura principal para el almacenamiento de datos y la configuración dentro del ERP. Estas tablas están

diseñadas para almacenar información como configuraciones de conexión a servicios en la nube y mapeos de documentos, que facilitan la integración con SharePoint.

También se trabaja en el diseño y desarrollo de páginas que permiten a los usuarios interactuar con el sistema de forma intuitiva. Estas interfaces de usuario se desarrollan para proporcionar accesibilidad y facilidad de uso, permitiendo operaciones como la carga, visualización y gestión de documentos directamente desde la interfaz de Business Central. Las páginas están diseñadas para mejorar la experiencia del usuario, incluyendo funciones de arrastrar y soltar para la carga de archivos.

#### **Cuarta quincena**

La cuarta quincena del proyecto, se enfoca en desarrollar y afinar las funcionalidades principales del módulo de gestión de archivos en la nube para Microsoft Dynamics 365 Business Central. Esta etapa es crucial para garantizar que las capacidades del sistema se alineen completamente con las necesidades operativas y los objetivos estratégicos del proyecto.

Se inicia la quincena implementando la codeunit principal, que encapsula toda la lógica de negocio necesaria para la interacción con los servicios de almacenamiento en la nube. Esta codeunit incluye procedimientos para la creación de enlaces a documentos, su actualización, eliminación y recuperación, lo que permite la gestión eficiente de los archivos dentro de SharePoint y otros servicios integrados.

Además, se desarrolla un control add-in para mejorar la interactividad del usuario. Este componente permite integraciones como la capacidad de arrastrar y soltar archivos directamente en la interfaz de Business Central, proporcionando una experiencia de usuario fluida y coherente.

También se dedica tiempo a realizar mejoras generales en el proyecto, como refinar la interfaz de usuario para hacerla más intuitiva y accesible, así como mejorar la eficiencia de las operaciones de procesamiento de archivos.

El finalizar estas tareas marca un punto significativo en el desarrollo del proyecto, preparándolo para entrar en la fase final de pruebas.

#### **Quinta quincena**

Durante la quinta quincena del proyecto, el enfoque principal está en realizar pruebas exhaustivas utilizando la codeunit creada. Esta fase es esencial para asegurar que todas las funcionalidades del módulo desarrollado funcionan correctamente antes de su lanzamiento.

En la codeunit se configura un entorno de pruebas, incluyendo la inicialización de los registros necesarios con datos específicos esenciales para probar las funcionalidades del sistema.

Las pruebas están diseñadas para ser repetibles y automatizadas, permitiendo múltiples ejecuciones con diferentes configuraciones de datos para validar la robustez del sistema.

## 2.4. Costes

La estimación de los costes en cualquier proyecto es fundamental para garantizar su viabilidad financiera y la adecuada asignación de recursos.

A continuación, se presenta un desglose detallado de los costes, proporcionando una visión clara de la inversión en recursos humanos y tecnológicos, y otros gastos necesarios para la realización del proyecto.

### 2.4.1. Recursos humanos

Para la categoría de recursos humanos, se estructura la información de manera detallada, asignando roles específicos.

En este proyecto se han involucrado varios roles clave, cada uno con responsabilidades y horas de trabajo claramente definidas, reflejando el compromiso y las habilidades necesarias para llevar a cabo el desarrollo.

- **Programador junior:** Este rol es lo ejerce el estudiante en prácticas, quien trabaja en el desarrollo y configuración del proyecto, implementando soluciones y ajustándose a los requisitos técnicos específicos.
- **Analista de proyecto:** Supervisor dentro de la empresa, encargado de definir los requisitos, supervisar el progreso y asegurar la alineación del proyecto con los objetivos empresariales. Además, proporciona asistencia técnica y supervisión regular.
- **Formador:** Personal especializado de la empresa que proporciona formación sobre el uso y capacidades de las herramientas y tecnologías empleadas, incluyendo Dynamics 365 Business Central y Visual Studio Code.

A continuación, en el cuadro 2.1 se presentan los costes asociados a estos roles y las horas invertidas:

Rol	Sueldo/hora (€)	Horas	Coste total (€)
Programador junior	7	300	2.100
Analista	20	100	2.000
Empleados	15	33	495
<b>Total</b>			<b>4.595</b>

Cuadro 2.1: Costes de recursos humanos.

Adicionalmente, se consideran los costes de contratación y costes indirectos:

- **Costes de contratación:** Incluyen gastos como impuestos de la seguridad social y formación inicial, que pueden representar entre el 20 - 30 % sobre los costes directos de recursos humanos.

- **Costes indirectos:** Comprenden gastos no directamente atribuibles a tareas específicas, tales como el mantenimiento de equipos y seguros. Estos costes se estiman en un 20 % adicional sobre los costes de recursos humanos.

#### 2.4.2. Software

Para la categoría de software, se estructura la información detallando los costes asociados a cada herramienta utilizada en el desarrollo.

En este proyecto, se considera únicamente el coste del desarrollo, por lo que solo se incluyen los costes de los programas utilizados para este fin. A continuación, se detalla cada uno:

- **Dynamics 365 Business Central:** Esencial para el desarrollo, utilizando todas las funcionalidades de este ERP. Es necesario contratar una licencia activa con un coste de 93,60€ al mes por usuario. [5]
- **Visual Studio Code:** Como software de código abierto, se utiliza su versión gratuita para el desarrollo del código en lenguaje AL.
- **GitHub:** Empleado para el control de versiones y la gestión del código fuente, la versión utilizada en este proyecto es gratuita, adaptándose a las necesidades del proyecto.
- **SharePoint Online:** Necesario para la integración y manejo de documentos y colaboración en el proyecto, con un coste de 11,70€ por usuario al mes. [8]
- **Outlook y Microsoft Teams:** Ambas herramientas de Microsoft se utilizan para la comunicación y gestión del proyecto sin costo adicional, ya que forman parte de las licencias corporativas existentes.
- **Extensiones Visual Studio Code:** Herramientas adicionales para mejorar la funcionalidad de Visual Studio Code, utilizadas sin coste alguno.

Dado que el proyecto tiene una duración estimada de 3 meses, los costes se han prorrateado para este período. A continuación, en el cuadro 2.2 se muestra un resumen de los costes de software:

Producto	Precio/mes (€)	Coste total (€)
Business Central 365	93,60	280,80
Visual Studio Code	0	0
GitHub	0	0
Outlook	0	0
Microsoft Teams	0	0
SharePoint Online	11,70	35,10
Extensiones Visual Studio Code	0	0
<b>Total</b>		<b>315,90</b>

Cuadro 2.2: Costes de software.

### 2.4.3. Hardware

Para la categoría de hardware, se consideran los equipos físicos utilizados durante el desarrollo. Este costo se prorrateará según la duración del proyecto, aunque los equipos seguirán siendo útiles después de su conclusión.

Recursos hardware:

- **Medion Akoya E15423 MD62562:** Este ordenador portátil se utiliza para el desarrollo y prueba del software. Equipado con un procesador Intel Core i5-1155G7, 16GB de RAM y 512GB SSD, es adecuado para manejar las demandas del desarrollo en Business Central. Su costo de 359€, se considera una inversión en infraestructura de TI que se amortiza a lo largo de la vida útil del dispositivo, estimada en varios años.
- **Ratón Tovarín:** Un ratón ergonómico utilizado por el equipo para mejorar la eficiencia durante largas horas de programación y diseño, cuyo coste es de 8,59€.

A continuación, en el cuadro 2.3, se detallan los costes de estos recursos para proporcionar una visión clara del desembolso inicial en hardware necesario para el proyecto.

Producto	Precio (€)	Vida útil (años)	Total 3 meses (€)
Ordenador portátil	359	4	22,44
Ratón Tovarín	8,59	3,5	0,61
<b>Total</b>			<b>23,05</b>

Cuadro 2.3: Costes de hardware.

## 2.5. Riesgos

En esta sección se identifican y analizan los riesgos asociados al proyecto, proporcionando tanto medidas de prevención como de contingencia para minimizar su impacto.

### 2.5.1. Identificación de riesgos

Los riesgos identificados son los siguientes:

- **R01:** Breve duración del proyecto.
- **R02:** Falta de entendimiento de los requisitos del proyecto.
- **R03:** Falta de permisos para realizar ciertas acciones.
- **R04:** Falta de experiencia en el lenguaje de Business Central (AL).

### 2.5.2. Análisis de riesgos

A continuación, se observa el análisis detallado de cada riesgo, incluyendo su magnitud, descripción, impacto, indicadores, y tipo.

- **R01:** Breve duración del proyecto.
  - Magnitud: Media.
  - Descripción: La duración limitada del proyecto puede resultar en falta de tiempo.
  - Impacto: Puede causar recortes en funcionalidad o calidad.
  - Indicadores: Retrasos en el cronograma.
  - Tipo: Riesgo de proyecto.
- **R02:** Falta de entendimiento de los requisitos del proyecto.
  - Magnitud: Media.
  - Descripción: La falta de entendimiento de los requisitos puede resultar en una solución incorrecta.
  - Impacto: Se requieren cambios significativos.
  - Indicadores: Discrepancias con los requisitos originales.
  - Tipo: Riesgo de producto.
- **R03:** Falta de permisos para realizar ciertas acciones.
  - Magnitud: Alta.
  - Descripción: La falta de permisos puede limitar la ejecución de funciones específicas.
  - Impacto: Puede causar retrasos o limitaciones en la funcionalidad del producto.
  - Indicadores: Errores de permisos en la ejecución de ciertas tareas.
  - Tipo: Riesgo de producto.
- **R04:** Falta de experiencia en el lenguaje de Business Central (AL).
  - Magnitud: Media.
  - Descripción: La falta de experiencia en el lenguaje AL puede ralentizar el desarrollo.
  - Impacto: Aumenta el tiempo de finalización del proyecto.
  - Indicadores: Retrasos en el desarrollo de funcionalidades.
  - Tipo: Riesgo de producto/proyecto.

### 2.5.3. Mitigación de riesgos

Para cada riesgo identificado, se proponen planes de prevención y contingencia:

- **R01:** Breve duración del proyecto.

- Prevención: Desarrollar un plan de gestión de tiempos realista.
- Contingencia: Priorizar la funcionalidad básica para cumplir con los plazos.
- **R02:** Falta de entendimiento de los requisitos del proyecto.
  - Prevención: Clarificar los requisitos en reuniones con el analista.
  - Contingencia: Realizar ajustes tempranos o evaluaciones tardías para modificaciones críticas.
- **R03:** Falta de permisos para realizar ciertas acciones.
  - Prevención: Realizar un estudio detallado de las funciones y permisos necesarios.
  - Contingencia: Solicitar los permisos necesarios al administrador.
- **R04:** Falta de experiencia en el lenguaje de Business Central (AL).
  - Prevención: Proporcionar formación adicional en el lenguaje AL.
  - Contingencia: Pausar el desarrollo temporalmente para adquirir más conocimientos.

#### 2.5.4. Riesgos experimentados

Durante la implementación del proyecto surgen varios desafíos que requieren una respuesta proactiva. La falta de permisos suficientes, inicialmente dificulta la ejecución de algunas funciones esenciales, y limita el avance del desarrollo. Para solucionar este problema, se comunica al administrador del sistema, quien facilita los permisos necesarios para continuar el trabajo sin obstáculos.

La falta de experiencia en el lenguaje AL también se convierte en un desafío técnico importante, el cual se aborda a través de una formación intensiva en el lenguaje AL, logrando así mayor fluidez en el desarrollo. La falta de entendimiento de los requisitos iniciales resulta en una falta de alineación en las primeras etapas del desarrollo. Para solucionarlo, se llevan a cabo reuniones periódicas con el analista del proyecto, lo que garantiza que el desarrollo esté alineado con las expectativas y las necesidades.

A pesar de estos desafíos, el proyecto se lleva a cabo con éxito, gracias a medidas correctivas, y la flexibilidad y coordinación constante del equipo.



## Capítulo 3

# Análisis del sistema/producto

### 3.1. Definición de requisitos

Para abordar el proyecto de gestión de archivos en la nube mediante SharePoint en el entorno de Microsoft Dynamics 365 Business Central, se realiza un análisis de los requisitos necesarios. Estos se establecen mediante historias de usuario, un método adecuado para capturar las necesidades funcionales del proyecto. La mayor parte de las funcionalidades están orientadas al front-end, lo que facilita la interacción y la experiencia del usuario.

#### Funcionalidad de gestión de archivos

- **HU01 - Conexión con SharePoint**

Como usuario de Business Central, quiero poder establecer una conexión fluida y bidireccional con SharePoint, para facilitar la gestión de documentos en la nube.

- **HU02 - Subida de documentos**

Como usuario de Business Central, quiero poder subir documentos a la estructura de carpetas en SharePoint, para mantener una organización eficiente de los archivos.

- **HU03 - Organización por carpetas**

Como usuario de Business Central, quiero que los documentos subidos se almacenen en la carpeta correspondiente según su tipo y mapeo, para evitar desorganizaciones y facilitar su acceso.

#### Funcionalidad de interacción con archivos

- **HU04 - Visualización de documentos**

Como usuario de Business Central, quiero poder ver la lista de documentos asociados a una entidad de Business Central, para encontrar rápidamente la información relevante.

- **HU05 - Descarga de documentos**

Como usuario de Business Central, quiero poder descargar documentos específicos desde

SharePoint directamente desde la interfaz de Business Central, para trabajar con ellos de manera local.

- **HU06 - Eliminación de documentos**

Como usuario de Business Central, quiero poder eliminar documentos de SharePoint directamente desde la interfaz de Business Central, para mantener la base de datos actualizada.

### 3.2. Análisis de requisitos

Para este proyecto el análisis de requisitos implica comprender cómo el sistema alcanzará sus objetivos y cubrirá las necesidades planteadas.

Dado el enfoque del proyecto en la gestión de archivos en la nube y la integración con Microsoft Dynamics 365 Business Central, se realiza un diagrama básico de casos de uso para representar las funcionalidades principales de la aplicación, el cual se puede observar en la Figura 3.1 [10]. Esto incluye la carga y gestión de archivos, la estructura de carpetas en SharePoint y la integración con Business Central.

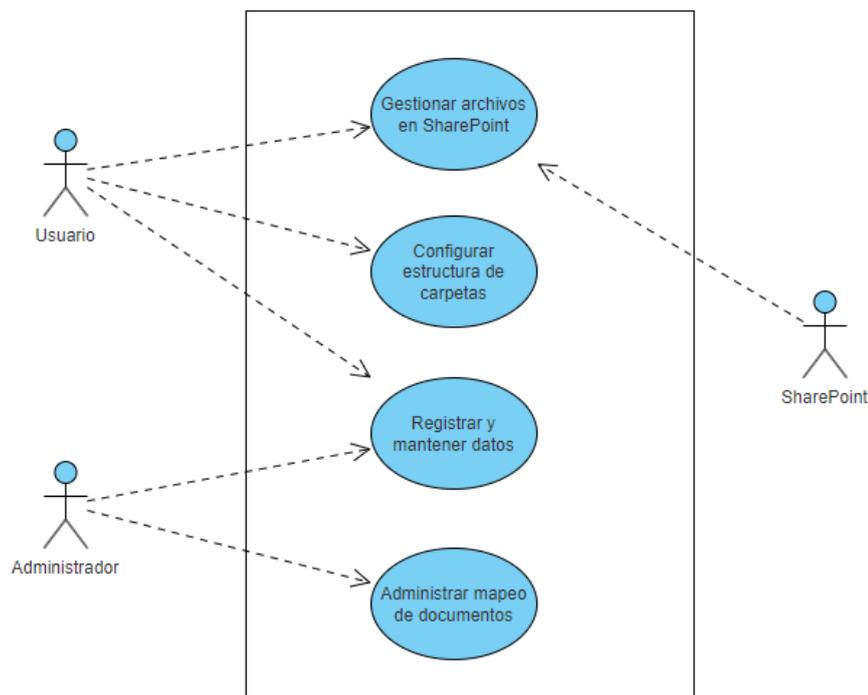


Figura 3.1: Diagrama de casos de uso.

En el diagrama, se identifican tres actores: el usuario, el administrador y SharePoint como un servicio externo. El usuario principal es el responsable de la aplicación, realizando tareas como cargar archivos, modificar estructuras de carpetas y visualizar documentos. El administrador, en cambio, gestiona la configuración y el mantenimiento del sistema.

A continuación, en base a la figura del diagrama de casos de uso, se presentan las tablas de la 3.1 hasta la 3.4 donde se describen las funcionalidades de cada uno de los casos de uso.

<b>Identificador</b>	<b>CU01</b>
Nombre:	Gestionar archivos en SharePoint
Versión:	1.0
Fuente:	Óscar Monlleó Recatalá
Actor principal:	Usuario
Descripción:	Permite al usuario cargar, organizar y gestionar archivos en SharePoint desde la interfaz de Business Central.
Secuencia normal:	<ol style="list-style-type: none"> <li>1. El usuario accede a la aplicación.</li> <li>2. Selecciona la opción de gestionar archivos.</li> <li>3. Carga o modifica archivos en SharePoint.</li> <li>4. Guarda y confirma los cambios.</li> </ol>

Cuadro 3.1: Definición caso de uso CU01

<b>Identificador</b>	<b>CU02</b>
Nombre:	Configurar estructura de carpetas
Versión:	1.0
Fuente:	Óscar Monlleó Recatalá
Actor principal:	Usuario
Descripción:	Permite al usuario definir la estructura de carpetas en SharePoint, asegurando una organización coherente de los archivos.
Secuencia normal:	<ol style="list-style-type: none"> <li>1. El usuario accede a la aplicación.</li> <li>2. Selecciona la opción de configuración de carpetas.</li> <li>3. Define o modifica la estructura de carpetas.</li> <li>4. Guarda y confirma los cambios.</li> </ol>

Cuadro 3.2: Definición caso de uso CU02

<b>Identificador</b>	<b>CU03</b>
Nombre:	Administrar mapeo de documentos
Versión:	1.0
Fuente:	Óscar Monlleó Recatalá
Actor principal:	Administrador
Descripción:	Permite al administrador definir cómo los documentos se mapean en SharePoint en relación con Business Central.
Secuencia normal:	<ol style="list-style-type: none"> <li>1. El administrador accede a la aplicación.</li> <li>2. Selecciona la opción de mapeo de documentos.</li> <li>3. Configura las reglas de mapeo.</li> <li>4. Guarda y confirma los cambios.</li> </ol>

Cuadro 3.3: Definición caso de uso CU03

<b>Identificador</b>	<b>CU04</b>
Nombre:	Registrar y mantener datos
Versión:	1.0
Fuente:	Óscar Monlleó Recatalá
Actor principal:	Usuario y Administrador
Descripción:	Permite registrar y mantener todos los cambios realizados en la aplicación.
Secuencia normal (Usuario):	<ol style="list-style-type: none"> <li>1. El usuario accede a la aplicación.</li> <li>2. Realiza cambios en archivos o carpetas.</li> <li>3. Confirma los cambios y los registra en SharePoint y Business Central.</li> </ol>
Secuencia normal (Administrador):	<ol style="list-style-type: none"> <li>1. El administrador accede a la aplicación</li> <li>2. Realiza modificaciones en la configuración.</li> <li>3. Guarda y confirma las actualizaciones.</li> </ol>

Cuadro 3.4: Definición caso de uso CU04

## Capítulo 4

# Diseño del sistema/producto

### 4.1. Diseño de la base de datos

En este apartado se aborda la estructura de la base de datos y su organización para satisfacer los requisitos del proyecto. Business Central, en su formato SaaS (software basado en la nube), utiliza la infraestructura de Azure SQL Database para almacenar los datos de la aplicación, la cual se puede observar en la Figura 4.1 [4]. Esta integración permite el flujo de datos eficiente entre las tablas de Business Central y los servicios en la nube, como en este caso SharePoint.

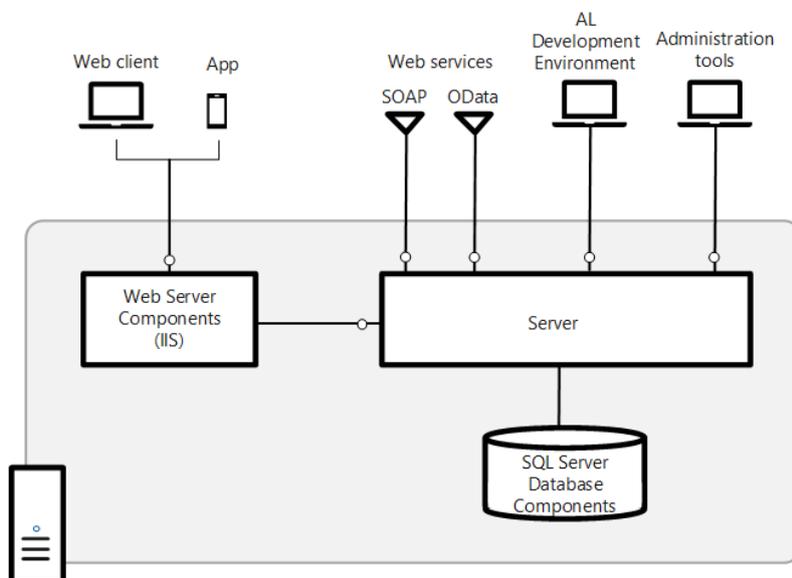


Figura 4.1: Arquitectura de la base de datos.

En este proyecto, se crean nuevas tablas y se extienden otras ya existentes para almacenar la información necesaria para la gestión de archivos en la nube. La estructura de la base de datos se diseñó de manera que facilite la conexión entre las tablas de Business Central y SharePoint, permitiendo una integración fluida y bidireccional.

Las nuevas tablas almacenan datos relacionados con la configuración de SharePoint, los enlaces de archivos y la estructura de mapeo para la organización de carpetas en la nube. También se añaden campos a tablas existentes para enriquecer la funcionalidad del sistema y mantener una relación coherente entre las diferentes entidades.

En la Figura 4.2 se muestra el diagrama de la base de datos desarrollada. Esta figura destaca la relación entre las nuevas tablas y las extensiones de tablas existentes, proporcionando una visión completa de la arquitectura de la base de datos y su flujo de datos. Además, se incluyen algunas tablas existentes que se utilizan, pero debido a la gran cantidad de atributos que poseen, solo se muestran aquellos que se emplean. El diagrama ha sido elaborado utilizando la plataforma Lucidchart [3].

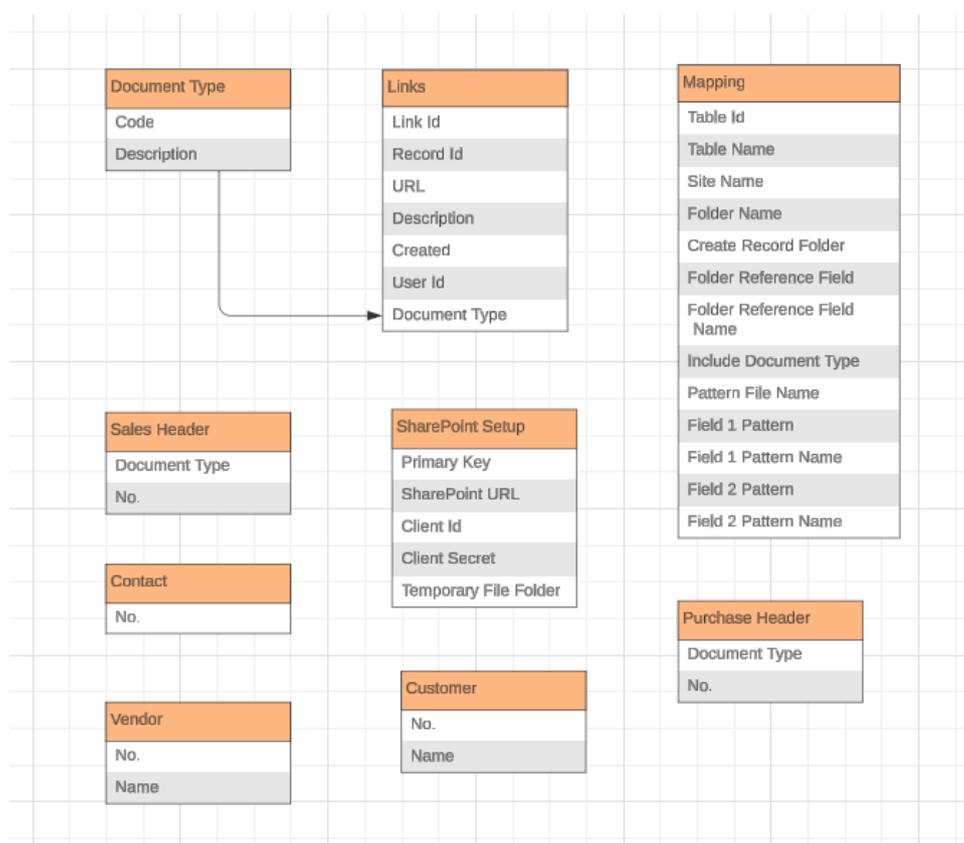


Figura 4.2: Diseño base de datos relacional.

## 4.2. Diseño de la arquitectura del sistema/producto

El diseño de la arquitectura del sistema para este proyecto se basa en una estructura de tres capas, como la que se puede observar en la Figura 4.3, que proporciona un flujo de datos fluido y una gestión eficiente de la funcionalidad del sistema. Esta arquitectura permite la interacción eficiente entre las capas de presentación, negocio y acceso a datos.

La capa de presentación incluye la interfaz de usuario en Business Central, que proporciona acceso directo a la funcionalidad del sistema y permite a los usuarios interactuar con los dife-

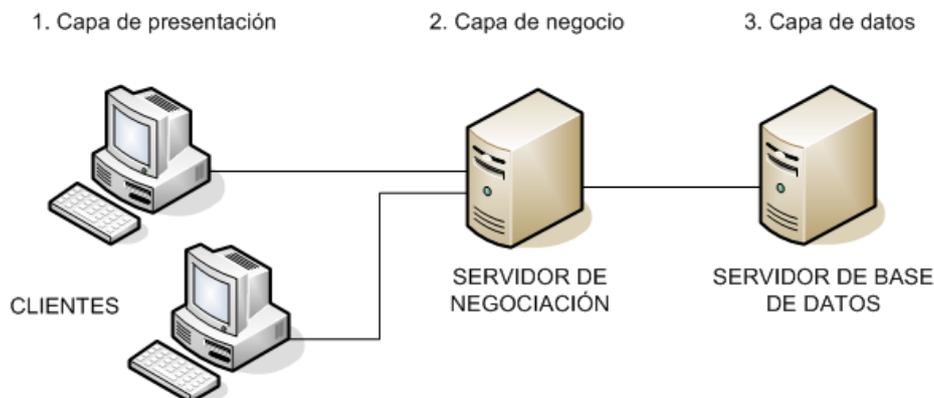


Figura 4.3: Arquitectura de tres capas.

rentes servicios de SharePoint. Aquí es donde los usuarios pueden cargar y gestionar archivos, así como acceder a enlaces y documentos relacionados.

La capa de negocio contiene la lógica de la aplicación, incluyendo las conexiones a SharePoint y la ejecución de funciones de negocio. Aquí se procesan las peticiones de los usuarios, y se implementan las funciones para la gestión de archivos y su integración con la estructura de carpetas en la nube.

La capa de acceso a datos permite la conexión entre las funciones de negocio y la base de datos de Business Central, así como con los servicios en la nube, como SharePoint. Esta capa facilita el almacenamiento y la recuperación de datos, permitiendo que la aplicación mantenga una relación coherente entre la estructura de la base de datos y la organización de archivos en la nube.

La arquitectura también integra una infraestructura de servidor gestionada por Microsoft, que aloja la instancia de Business Central y su base de datos en la nube. Los servicios web SOAP y OData permiten una comunicación bidireccional eficiente entre el servidor y los componentes del sistema, facilitando la gestión de archivos y datos.

### 4.3. Diseño de las interfaces

Dado que el proyecto se basa en la integración de SharePoint con un ERP ya existente, las interfaces principales vienen definidas por defecto. Esto garantiza una interfaz intuitiva y coherente para los usuarios finales, a pesar de limitar la personalización de los desarrolladores. La ventaja radica en que los usuarios familiarizados con el ERP encontrarán la interfaz intuitiva, lo que facilita la adopción del nuevo sistema. Sin embargo, la desventaja es la restricción en la personalización y el estilo, limitando la capacidad de los desarrolladores para modificar el diseño visual.

El menú de navegación del ERP se ha ampliado para incluir nuevas páginas desarrolladas específicamente para la integración de SharePoint. Estas páginas son SharePoint Setup y

Mapping, tal y como se puede observar en la Figura 4.4.



Figura 4.4: Menú de navegación de Business Central.

La página de SharePoint Setup, la cual podemos observar en la Figura 4.5 permite a los usuarios configurar la conexión con el servicio de SharePoint, lo que incluye varios elementos esenciales. El usuario puede ingresar la dirección web del sitio de SharePoint que se utilizará para la gestión documental, estableciendo así la conexión base entre el ERP y la plataforma en la nube. Además, se pueden ingresar el ID del cliente y el secreto del cliente (Client ID y Client Secret). Estas credenciales son esenciales para la autenticación, garantizando un acceso seguro y autorizado a la plataforma de SharePoint desde el ERP.

La página de Mapping, que se puede observar en la Figura 4.6 facilita la configuración de la relación entre las tablas del ERP y las carpetas de SharePoint. Esta función permite a los usuarios mapear registros específicos del ERP a carpetas específicas en SharePoint, por ejemplo, la tabla de clientes del ERP puede estar vinculada a una carpeta específica en SharePoint destinada a almacenar documentos relacionados con dichos clientes. La estructura de carpetas en SharePoint se va modificando de manera dinámica, siguiendo la configuración de la relación establecida en la página de Mapping. Esto garantiza que los documentos se almacenen en las carpetas correctas, según la estructura definida para la organización, facilitando la gestión y el acceso a los documentos. La página de Mapping también permite configurar detalles adicionales, como la inclusión del tipo de documento, tal y como se observa en la Figura 4.7, en la estructura de la carpeta o el uso de patrones de nombres para los archivos, lo que mejora aún más la organización documental.

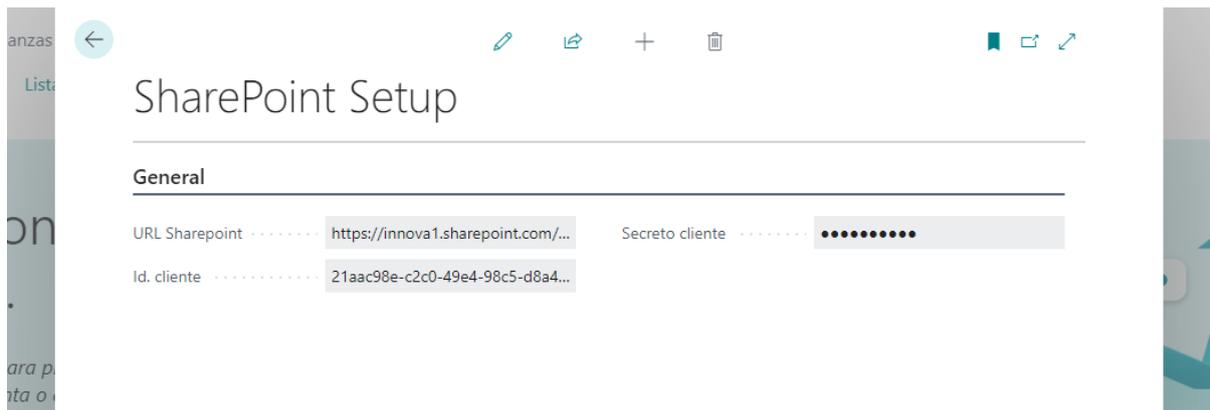


Figura 4.5: SharePoint Setup.

Se han desarrollado extensiones que mejoran la funcionalidad de la plataforma, permitiendo una integración más efectiva entre el ERP y SharePoint. Estas extensiones se centran principalmente en dos áreas: la carga de documentos y la gestión de los archivos asociados a cada registro.

En primer lugar, la funcionalidad de arrastrar y soltar (drag-and-drop), que podemos obser-

CRONUS ES | Finanzas | Tesorería | Ventas | Compras | Shopify | Todos los informes

Lista mapeos: Todo | Buscar | Analizar | + Nuevo | Eliminar | Editar lista

Id. tabla ↑	Nombre tabla	Nombre sitio	Nombre carpeta	Crear subcarpeta por registro	Campo referencia carpeta	Nombre campo referencia carpeta	Incluir tipo documento	Patrón nombre fichero	Campo 1 patrón	Nombre campo
18	Customer	https://innova1.sharepoi...	Customer	<input checked="" type="checkbox"/>	1	Nº	<input type="checkbox"/>	%3	2	Nombre
23	Vendor	https://innova1.sharepoi...	Vendor	<input checked="" type="checkbox"/>	1	Nº	<input type="checkbox"/>	%3	2	Nombre
36	Sales Header	https://innova1.sharepoi...	Sales	<input type="checkbox"/>	0		<input type="checkbox"/>	%1 %2 - %3	1	Tipo documen
38	Purchase Header	https://innova1.sharepoi...	Header	<input type="checkbox"/>	0		<input checked="" type="checkbox"/>	Vendedor %1 %3	1	Tipo documen

Figura 4.6: Tabla de mapeo.

Lista tipo documentos | Buscar | + Nuevo | Editar lista

Código ↑	Descripción
→ FACTURA	Factura
OFERTA	Oferta
VARIOS	Varios

Aceptar Cancelar

Figura 4.7: Tipos de documentos.

var en la Figura 4.8 y en la Figura 4.9, permite a los usuarios cargar archivos directamente en la interfaz del ERP de manera intuitiva y rápida. Esto simplifica considerablemente el proceso de carga de documentos en la plataforma, evitando la necesidad de navegar por menús o ventanas emergentes adicionales. Al arrastrar un archivo desde el sistema de archivos local y soltarlo en la sección correspondiente de la interfaz del ERP, el documento se carga automáticamente y se almacena en la ubicación definida en SharePoint.

Además, se ha implementado una lista de documentos asociada a cada registro específico, se puede visualizar en la Figura 4.10. Esta lista permite a los usuarios ver de forma rápida y clara todos los archivos vinculados a un registro particular, como un cliente, un proveedor o un pedido. Esto proporciona un acceso inmediato y completo a toda la información necesaria para gestionar la documentación asociada a cada registro.

La lista de documentos también incluye opciones para gestionar cada archivo de manera integral. Los usuarios pueden seleccionar un archivo específico y optar por visualizarlo o borrarlo directamente desde la interfaz del ERP, sin necesidad de salir del sistema. Esto permite una gestión documental más eficiente y coherente, integrando todas las operaciones en una única plataforma. En la Figura 4.11 podemos visualizar todas las funciones de las que disponemos en la factbox.

## SharePoint

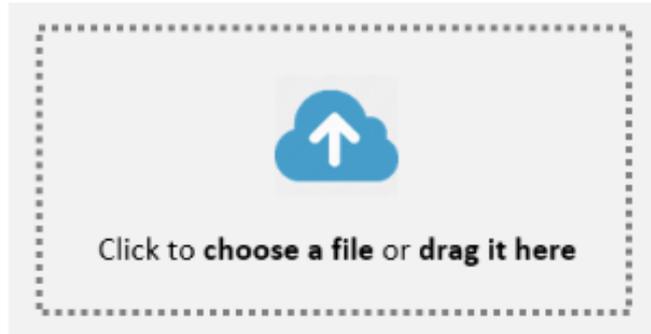


Figura 4.8: Funcionalidad de drag & drop.

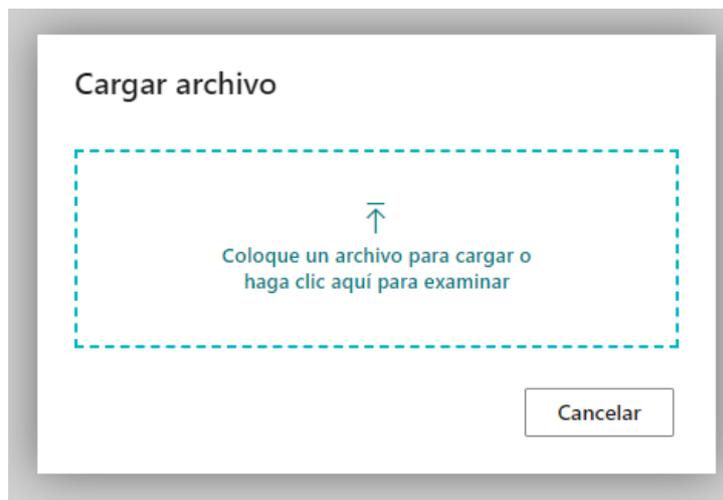


Figura 4.9: Funcionalidad de clic.

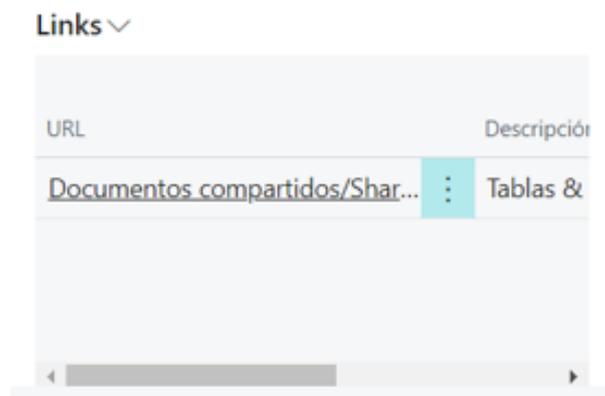


Figura 4.10: Listado de ficheros en SharePoint.

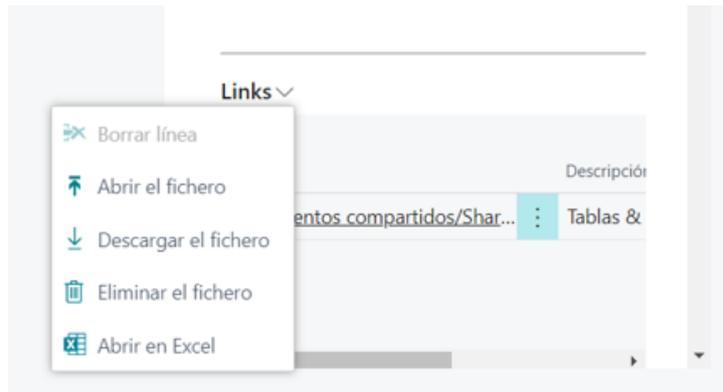


Figura 4.11: Listado de funciones.

En SharePoint no podemos modificar la interfaz, simplemente modificaremos la estructura de carpetas, esta estructura se visualiza como en la Figura 4.12. La estructura de carpetas en SharePoint se modifica dinámicamente según la configuración realizada en la página de Mapping. Esto permite mapear de forma efectiva directorios de SharePoint con registros del ERP, asegurando una estructura cohesiva para la gestión documental.

La página de Mapping permite establecer relaciones entre las tablas de datos del ERP y las carpetas en SharePoint, lo que facilita la organización documental. Esta configuración posibilita definir las estructuras de carpetas de manera lógica, de acuerdo con la naturaleza de los registros gestionados en el ERP.

Además, Mapping permite establecer reglas de nomenclatura para las carpetas y los archivos dentro de SharePoint. Esto garantiza la consistencia en la estructura documental, lo que resulta fundamental para una gestión eficiente de archivos en la nube. Las reglas de nomenclatura pueden incorporar diferentes elementos, como el tipo de documento, el nombre del cliente, o la fecha de creación, lo que contribuye a mantener una estructura clara y ordenada.

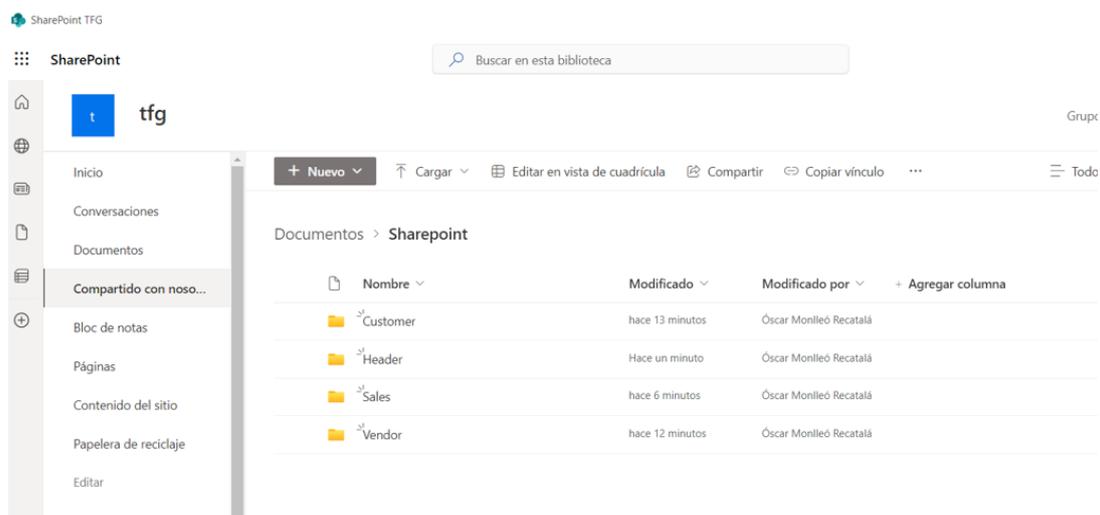


Figura 4.12: Estructura de carpetas en Sharepoint.



## Capítulo 5

# Implementación y pruebas

### 5.1. Estructura del código

En el desarrollo del código, se mantiene la estructura estándar que se utiliza en Business Central. Esta estructura se basa en la separación del código en diferentes objetos de AL (Application Language), cada uno con un propósito específico dentro del sistema ERP. Los objetos utilizados son:

- **Table:** Representan tablas de la base de datos. En este proyecto, las tablas como "Document Type", "Links", y "Mapping" son cruciales, ya que almacenan la configuración y los datos necesarios para la gestión de documentos y enlaces de SharePoint.
- **Page:** Representan páginas en la interfaz de usuario de Business Central. Estas páginas, como "Document Type List" y "Mapping List", permiten a los usuarios interactuar con los datos almacenados en las tablas desde la interfaz gráfica.
- **Page Extension:** Modifican las páginas existentes, añadiendo o eliminando elementos para personalizar la funcionalidad y la presentación de la información. Estas extensiones son vitales para incorporar los factboxes de arrastrar y soltar y para enlazar documentos directamente en páginas estándar como las de "Contactos" o "Facturas".
- **Codeunit:** Contienen lógica de negocio reutilizable y se pueden invocar desde cualquier parte del sistema. En este proyecto, el codeunit "SharePoint Management" encapsula toda la lógica necesaria para interactuar con SharePoint, facilitando operaciones como guardar, recuperar y eliminar documentos.
- **Control Add-in:** Manejan funcionalidades de interfaz de usuario no estándares, como el control para arrastrar y soltar archivos. Estos add-ins mejoran significativamente la interactividad de las páginas y proporcionan funcionalidades avanzadas directamente en la interfaz de usuario de Business Central.
- **Permission Set:** Definen los permisos de acceso para diferentes usuarios o grupos de usuarios dentro de Business Central, asegurando que solo los usuarios autorizados puedan realizar operaciones específicas.

El proyecto se divide en dos directorios principales: uno para los tests que se puede observar en la Figura 5.2 y otro para el código principal, que se muestra en la Figura 5.1, facilitando la separación clara entre el desarrollo y la validación del código. Dentro de cada carpeta, se incluyen elementos esenciales como:

- **.alpackages:** Contiene las definiciones de estructuras utilizadas en el ERP.
- **launch.json:** Configuración necesaria para ejecutar el código localmente durante el desarrollo.
- **Translations:** Carpeta con archivos .xlf para la localización y traducción del proyecto.
- **app.json:** Configuración general del proyecto.

Se crea una carpeta para cada tipo de objeto AL, organizando el código de manera que facilita la localización y modificación de cada componente.

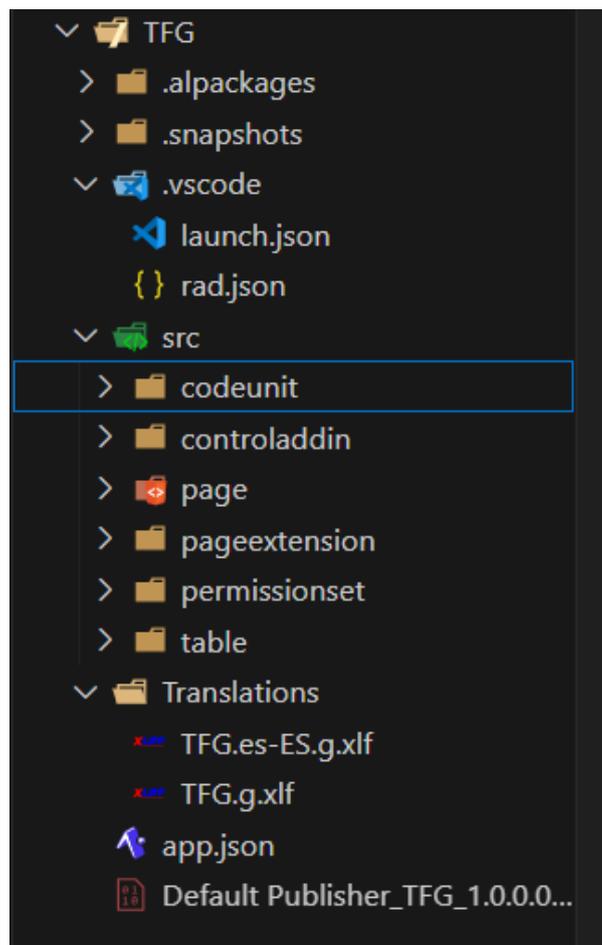


Figura 5.1: Estructura del proyecto.

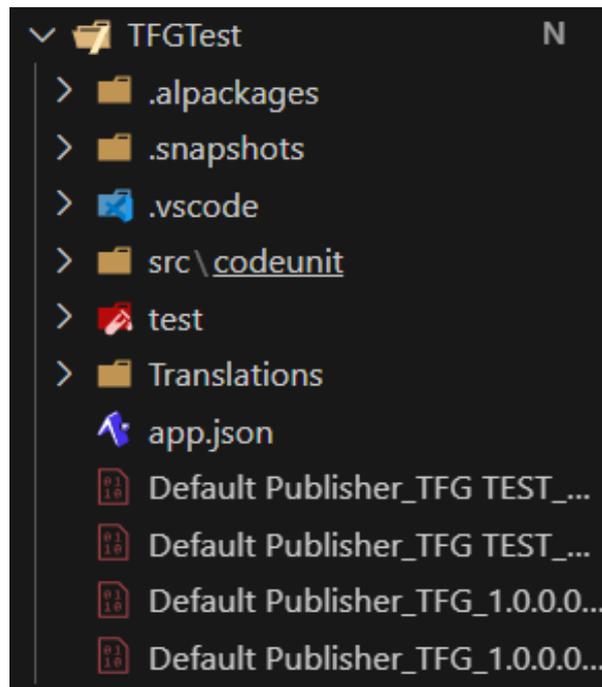


Figura 5.2: Estructura directorio tests.

## 5.2. Descripción técnica de la implementación

### Tablas

En la estructura del proyecto se crean varias tablas esenciales para gestionar la documentación y los enlaces entre Business Central y SharePoint, las cuales se pueden observar en la Figura 5.3. Las tablas implementadas incluyen 'Document Type', 'Mapping' y 'Links'. Cada tabla se define en archivos AL con la nomenclatura NombreObjeto.Table.al, facilitando su identificación y mantenimiento. La definición de las tablas se realiza mediante el lenguaje AL, sin necesidad de comandos SQL directos, lo que permite una integración más fluida dentro del ecosistema de Business Central. Cada campo de la tabla tiene una traducción al español especificada en los archivos .xlf generados automáticamente al compilar el proyecto.

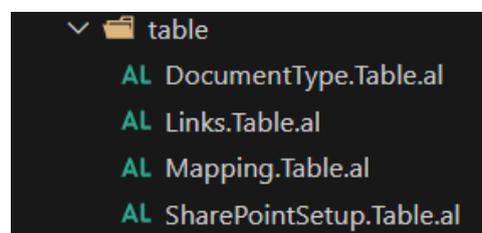


Figura 5.3: Tablas.

- **SharePoint Setup:** Esta tabla se utiliza para almacenar la configuración necesaria para conectar y operar con un sitio de SharePoint. Sirve como almacenamiento centralizado para los parámetros de configuración que se necesitan para acceder y gestionar recursos

en SharePoint. En las Figuras 5.4 y 5.5 se representa la tabla SharePoint Setup.

```
AL SharePointSetup.Table.al X
TFG > src > table > AL SharePointSetup.Table.al > Table 50100 "SharePoint Setup" > fields > "SharePoint URL": Text[250]
7 references
1 table 50100 "SharePoint Setup"
2 {
3     DataClassification = CustomerContent;
4     Caption = 'SharePoint Setup', Comment = 'ESP="Configuración SharePoint"';
5
6     fields
7     {
8         1 reference
9         field(1; "Primary Key"; Code[10])
10        {
11            Caption = 'Primary Key', Comment = 'ESP="Clave Primaria"';
12            DataClassification = CustomerContent;
13        }
14        4 references
15        field(2; "SharePoint URL"; Text[250])
16        {
17            Caption = 'SharePoint URL', Comment = 'ESP="URL Sharepoint"';
18            DataClassification = CustomerContent;
19        }
20        2 references
21        field(3; "Client Id"; Text[250])
22        {
23            Caption = 'Client Id', Comment = 'ESP="Id. cliente"';
24            DataClassification = EndUserIdentifiableInformation;
25        }
26        2 references
27        field(4; "Client Secret"; Text[250])
28        {
29            Caption = 'Client Secret', Comment = 'ESP="Secreto cliente"';
30            DataClassification = EndUserIdentifiableInformation;
31        }
32    }
33 }
34 }
```

Figura 5.4: Tabla SharePoint Setup.

```
0 references
28 field(5; "Temporary File Folder"; Text[250])
29 {
30     Caption = 'Temporary File Folder', Comment = 'ESP="Carpeta temporal archivos"';
31     DataClassification = CustomerContent;
32 }
33 }
34
35 keys
36 {
37     0 references
38     key(PK; "Primary Key")
39     {
40         Clustered = true;
41     }
42 }
```

Figura 5.5: Tabla SharePoint Setup 2.

- **Mapping:** Esta tabla se utiliza para gestionar la estructura y almacenamiento de documentos y datos dentro de SharePoint. Permite definir y personalizar la relación entre registros en Business Central y su representación en SharePoint. En las Figuras 5.6 hasta 5.12 se representa la estructura de la tabla Mapping.

```

AL Mapping.Table.al X
TFG > src > table > AL Mapping.Table.al > Table 50102 Mapping > fields > "Field 1 Pattern Name": Text[80]
4 references
1 table 50102 Mapping
2 {
3   DataClassification = CustomerContent;
4   Caption = 'Mapping', Comment = 'ESP="Mapeos"';
5   DrillDownPageId = "Mapping List";
6   LookupPageId = "Mapping List";
7
8   fields
9   {
10    7 references
11    field(1; "Table Id"; Integer)
12    {
13      Caption = 'Table Id', Comment = 'ESP="Id. tabla"';
14      DataClassification = CustomerContent;
15      TableRelation = AllObjWithCaption."Object ID" where("Object Type" = filter(Table));
16    }
17    1 reference
18    field(10; "Table Name"; Text[100])
19    {
20      Caption = 'Table Name', Comment = 'ESP="Nombre tabla"';
21      FieldClass = FlowField;
22      CalcFormula = lookup(AllObjWithCaption."Object Name" where("Object Type" = const(Table), "Object ID" = field("Table ID")));
23      Editable = false;
24    }
25    1 reference
26    field(2; "Site Name"; Text[250])
27    {
28      Caption = 'Site Name', Comment = 'ESP="Nombre sitio"';
29      DataClassification = CustomerContent;
30    }
31  }
32 }
2 references

```

Figura 5.6: Tabla Mapping 1.

```

28 2 references
29 field(3; "Folder Name"; Text[250])
30 {
31   Caption = 'Folder Name', Comment = 'ESP="Nombre carpeta"';
32   DataClassification = CustomerContent;
33 }
34 2 references
35 field(4; "Create Record Folder"; Boolean)
36 {
37   Caption = 'Create Record Folder', Comment = 'ESP="Crear subcarpeta por registro"';
38   DataClassification = CustomerContent;
39 }
40 4 references
41 field(5; "Folder Reference Field"; Integer)
42 {
43   Caption = 'Folder Reference Field', Comment = 'ESP="Campo referencia carpeta"';
44   DataClassification = CustomerContent;
45 }
46 0 references
47 trigger OnLookup();
48 var
49   NewFieldId: Integer;
50 begin
51   if LookupFieldID(NewFieldId) then
52     Validate("Folder Reference Field", NewFieldId);
53 end;
54 1 reference
55 field(11; "Folder Reference Field Name"; Text[80])
56 {
57   Caption = 'Folder Reference Field caption', Comment = 'ESP="Nombre campo referencia carpeta"';
58   FieldClass = FlowField;
59   CalcFormula = lookup(Field."Field Caption" where(TableNo = field("Table Id"), "No." = field("Folder Reference Field")));
60   Editable = false;
61 }

```

Figura 5.7: Tabla Mapping 2.

```

58      2 references
59      field(6; "Include Document Type"; Boolean)
60      {
61          Caption = 'Include Document Type', Comment = 'ESP="Incluir tipo documento"';
62          DataClassification = CustomerContent;
63      }
64      2 references
65      field(7; "Pattern File Name"; Text[250])
66      {
67          Caption = 'Pattern File Name', Comment = 'ESP="Patrón nombre fichero"';
68          DataClassification = CustomerContent;
69      }
70      4 references
71      field(8; "Field 1 Pattern"; Integer)
72      {
73          Caption = 'Field 1 Pattern', Comment = 'ESP="Campo 1 patrón"';
74          DataClassification = CustomerContent;
75      }
76      0 references
77      trigger OnLookup();
78      var
79          NewFieldId: Integer;
80      begin
81          if LookupFieldID(NewFieldId) then
82              Validate("Field 1 Pattern", NewFieldId);
83      end;
84      1 reference
85      field(12; "Field 1 Pattern Name"; Text[80])
86      {
87          Caption = 'Field 1 Pattern caption', Comment = 'ESP="Nombre campo 1 patrón"';
88          FieldClass = FlowField;
89          CalcFormula = lookup(Field."Field Caption" where(TableNo = field("Table Id"), "No." = field("Field 1 Pattern")));
90          Editable = false;
91      }

```

Figura 5.8: Tabla Mapping 3.

```

88      4 references
89      field(9; "Field 2 Pattern"; Integer)
90      {
91          Caption = 'Field 2 Pattern', Comment = 'ESP="Campo 2 patrón"';
92          DataClassification = CustomerContent;
93      }
94      0 references
95      trigger OnLookup();
96      var
97          NewFieldId: Integer;
98      begin
99          if LookupFieldID(NewFieldId) then
100              Validate("Field 2 Pattern", NewFieldId);
101      end;
102      1 reference
103      field(13; "Field 2 Pattern Name"; Text[80])
104      {
105          Caption = 'Field 2 Pattern caption', Comment = 'ESP="Nombre campo 2 patrón"';
106          FieldClass = FlowField;
107          CalcFormula = lookup(Field."Field Caption" where(TableNo = field("Table Id"), "No." = field("Field 2 Pattern")));
108          Editable = false;
109      }
110      keys
111      {
112          0 references
113          key(PK; "Table Id")
114          {
115              Clustered = true;
116          }
117      }

```

Figura 5.9: Tabla Mapping 4.

```

1 reference
118 procedure CalculatePath(RecordId: RecordId; DocumentTypeCode: Code[10]): Text
119 var
120     RecordRef: RecordRef;
121     FieldRef: FieldRef;
122     Path: Text;
123     FixedPathPart: Text;
124 begin
125     FixedPathPart := 'Documentos compartidos/Sharepoint/';
126     Path := FixedPathPart + "Folder Name" + '/';
127
128     RecordRef.Get(RecordId);
129     if "Create Record Folder" then begin
130         FieldRef := RecordRef.Field("Folder Reference Field");
131         Path += Format(FieldRef.Value) + '/';
132     end;
133
134     if "Include Document Type" then
135         Path += DocumentTypeCode + '/';
136
137     exit(Path);
138 end;
139

```

Figura 5.10: Tabla Mapping 5.

```

1 reference
140 procedure CalculateFileName(RecordId: RecordId; FileName: Text): Text
141 var
142     RecordRef: RecordRef;
143     FieldRef: FieldRef;
144     FieldRef2: FieldRef;
145     SharePointFileName: Text;
146 begin
147     RecordRef.Get(RecordId);
148
149     FieldRef := RecordRef.Field("Field 1 Pattern");
150     FieldRef2 := RecordRef.Field("Field 2 Pattern");
151     SharePointFileName := StrSubstNo("Pattern File Name", FieldRef.Value, FieldRef2.Value, FileName);
152
153     exit(SharePointFileName);
154 end;
155

```

Figura 5.11: Tabla Mapping 6.

```

3 references
156 local procedure LookupFieldID(var NewFieldId: Integer): Boolean;
157 var
158     FieldTable: Record Field;
159     FieldsLookup: Page "Fields Lookup";
160 begin
161     FieldTable.FilterGroup(2);
162
163     FieldTable.SetRange(TableNo, "Table Id");
164     FieldTable.SetRange(Class, FieldTable.Class::Normal);
165
166     FieldTable.SetFilter(Type, '%1|%2|%3|%4|%5|%6|%7|%8|%9', FieldTable.Type::Code, FieldTable.Type::Text, FieldTable.Type::Integer,
167     FieldTable.Type::Decimal, FieldTable.Type::Date, FieldTable.Type::Datetime, FieldTable.Type::Option, FieldTable.Type::Boolean,
168     FieldTable.Type::GUID);
169     FieldTable.FilterGroup(0);
170
171     FieldsLookup.SetRecord(FieldTable);
172     FieldsLookup.SetTableView(FieldTable);
173     FieldsLookup.LookupMode(true);
174
175     if FieldsLookup.RunModal() = Action::LookupOK then begin
176         FieldsLookup.GetRecord(FieldTable);
177         NewFieldId := FieldTable.No.;
178         exit(true);
179     end;
180
181     exit(false);
182 end;

```

Figura 5.12: Tabla Mapping 7.

- Links:** Gestiona los enlaces generados entre los documentos de Business Central y sus ubicaciones correspondientes en SharePoint, facilitando un seguimiento eficiente de dónde y cómo se almacenan los documentos externamente. Podemos ver representada la estructura de la tabla Links en las Figuras 5.13 hasta 5.18.

```

AL Links.Table.al X
TFG > src > table > AL Links.Table.al > Table 50101 Links
8 references
1 table 50101 Links
2 {
3   DataClassification = CustomerContent;
4   Caption = 'Links', Comment = 'ESP="Vinculos"';
5   LookupPageId = "Links List";
6   DrillDownPageId = "Links List";
7
8   fields
9   {
10      7 references
11      field(1; "Link Id"; Integer)
12      {
13        Caption = 'Link Id', Comment = 'ESP="Id. vínculo"';
14        DataClassification = CustomerContent;
15        AutoIncrement = true;
16      }
17      3 references
18      field(2; "Record Id"; RecordId)
19      {
20        Caption = 'Record Id', Comment = 'ESP="Id. registro"';
21        DataClassification = CustomerContent;
22      }
23      9 references
24      field(3; "URL"; Text[2048])
25      {
26        Caption = 'URL', Comment = 'ESP="URL"';
27        DataClassification = CustomerContent;
28      }
29      10 references
30      field(4; "Description"; Text[250])
31      {
32        Caption = 'Description', Comment = 'ESP="Descripción"';
33        DataClassification = CustomerContent;
34      }
35    }
36 }
  
```

Figura 5.13: Tabla Links 1.

```

2 references
31 field(5; "Created"; DateTime)
32 {
33   Caption = 'Created', Comment = 'ESP="Creado"';
34   DataClassification = CustomerContent;
35 }
36 3 references
37 field(6; "User Id"; Code[50])
38 {
39   Caption = 'User Id', Comment = 'ESP="Id. usuario"';
40   DataClassification = CustomerContent;
41   TableRelation = User;
42 }
43 3 references
44 field(7; "Document Type"; Code[10])
45 {
46   Caption = 'Document Type', Comment = 'ESP="tipo documento"';
47   DataClassification = CustomerContent;
48   TableRelation = "Document Type";
49 }
50 keys
51 {
52   0 references
53   key(PK; "Link Id")
54   {
55     Clustered = true;
56   }
57 }
  
```

Figura 5.14: Tabla Links 2.

```

57 // Definición de eventos de integración para simular las acciones (TEST)
58 [IntegrationEvent(false, false)]
59 1 reference
60 local procedure OnBeforeCreateLink(var SharePointPath: Text; var SharePointFileName: Text; var FileInStream: InStream; var Handled: Boolean)
61 begin
62 end;
63
64 [IntegrationEvent(false, false)]
65 1 reference
66 local procedure OnBeforeDeleteLink(var SharePointPathURL: Text; var LinkId: Integer; var Handled: Boolean)
67 begin
68 end;
69
70 [IntegrationEvent(false, false)]
71 1 reference
72 local procedure OnBeforeDownloadLink(var SharePointPathURL: Text; var Handled: Boolean)
73 begin
74 end;
75
76 [IntegrationEvent(false, false)]
77 1 reference
78 local procedure OnBeforeOpenLink(var SharePointPathURL: Text; var Handled: Boolean)
79 begin
80 end;

```

Figura 5.15: Tabla Links 3.

```

79 procedure CreateLink(FileInStream: InStream; GlobalTableId: Integer; GlobalRecordId: RecordId; DocumentTypeCode: Code[10]; FileName: Text)
80 var
81     Mapping: Record Mapping;
82     Links: Record Links;
83     SharePointManagement: Codeunit "Sharepoint Management";
84     SharePointPath: Text;
85     SharePointFileName: Text;
86     LinkId: Integer;
87     Handled: Boolean;
88     FileSaved: Boolean;
89 begin
90     Mapping.Get(GlobalTableId);
91     SharePointPath := Mapping.CalculatePath(GlobalRecordId, DocumentTypeCode);
92     SharePointFileName := Mapping.CalculateFileName(GlobalRecordId, FileName);
93
94     OnBeforeCreateLink(SharePointPath, SharePointFileName, FileInStream, Handled);
95     if not Handled then
96         if SharePointManagement.SaveFile(SharePointPath, SharePointFileName, FileInStream) then
97             FileSaved := true;
98
99     if Handled or FileSaved then begin
100         if Links.FindLast() then
101             LinkId := Links."Link Id" + 1
102         else
103             LinkId := 1;
104
105         Links.Init();
106         Links."Link Id" := LinkId;
107         Links.Description := CopyStr(FileName, 1, 250);
108         Links."Record Id" := GlobalRecordId;
109         Links.URL := CopyStr(SharePointPath, 1, MaxStrLen(Links.URL));
110         Links.Created := CurrentDateTime;
111         Links."User Id" := CopyStr(UserId, 1, MaxStrLen(Links."User Id"));
112         Links."Document Type" := DocumentTypeCode;
113         Links.Insert(true);
114     end;

```

Figura 5.16: Tabla Links 4.

```

117 procedure Deletelink()
118 var
119     Links: Record Links;
120     SharePointSetup: Record "SharePoint Setup";
121     SharePointManagement: Codeunit "Sharepoint Management";
122     SharePointPathURL: Text;
123     Handled: Boolean;
124 begin
125     SharePointSetup.Get();
126     SharePointPathURL := Rec.URL + Rec.Description;
127     OnBeforeDeleteLink(SharePointPathURL, Rec."Link Id", Handled);
128     if not Handled then
129         SharePointManagement.DeleteFile(SharePointPathURL, Rec."Link Id");
130
131     if Links.Get(Rec."Link Id") then
132         Links.Delete();
133 end;
134
135 1 reference
136 procedure DownloadLink()
137 var
138     SharePointSetup: Record "SharePoint Setup";
139     SharePointManagement: Codeunit "Sharepoint Management";
140     SharePointPathURL: Text;
141     Handled: Boolean;
142 begin
143     SharePointSetup.Get();
144     if Rec.Description <> '' then begin
145         SharePointPathURL := Rec.URL + Rec.Description;
146
147         OnBeforeDownloadLink(SharePointPathURL, Handled);
148         if not Handled then
149             SharePointManagement.DownloadFile(SharePointPathURL);
150     end;
151 end;

```

Figura 5.17: Tabla Links 5.

```

152 1 reference
153 procedure OpenLink()
154 var
155     SharePointSetup: Record "SharePoint Setup";
156     SharePointPathURL: Text;
157     Handled: Boolean;
158 begin
159     SharePointSetup.Get();
160     if Rec.Description <> '' then begin
161         SharePointPathURL := SharePointSetup."SharePoint URL" + Rec.URL + Rec.Description;
162
163         OnBeforeOpenLink(SharePointPathURL, Handled);
164         if not Handled then
165             Hyperlink(SharePointPathURL);
166     end;
167 end;

```

Figura 5.18: Tabla Links 6.

- Document Type:** Proporciona una clasificación para los tipos de documentos que se manejan, permitiendo a los usuarios definir y categorizar los documentos según las necesidades del negocio. En la Figura 5.19 aparece representada la estructura de la tabla 'Document Type'.

```

AL DocumentType.Table.al x
TFG > src > table > AL DocumentType.Table.al > Table 50103 "Document Type"
5 references
1 table 50103 "Document Type"
2 {
3   DataClassification = CustomerContent;
4   Caption = 'Document Type', Comment = 'ESP="Tipo documento"';
5   LookupPageId = "Document Type List";
6   DrillDownPageId = "Document Type List";
7
8   fields
9   {
10    4 references
11    field(1; "Code"; Code[10])
12    {
13      Caption = 'Code', Comment = 'ESP="Código"';
14      DataClassification = CustomerContent;
15      NotBlank = true;
16    }
17    1 reference
18    field(2; "Description"; Text[50])
19    {
20      Caption = 'Description', Comment = 'ESP="Descripción"';
21      DataClassification = CustomerContent;
22    }
23  }
24  keys
25  {
26    0 references
27    key(PK; "Code")
28    {
29      Clustered = true;
30    }
31  }
32 }

```

Figura 5.19: Tabla Document Type.

## Páginas

Definida la estructura de datos, se procede a desarrollar páginas que permiten a los usuarios interactuar con los datos. Las páginas se diseñan para introducir, mostrar o modificar los datos, incorporando grupos de campos, partes para detalles adicionales y acciones para la interacción del usuario. Cada página incluye tooltips en español y gatillos que se activan en acciones específicas, como seleccionar un cliente para facturación automática. La estructura de las páginas queda definida tal y como se puede observar en la Figura 5.20

```

page
AL DocumentTypeList.Page.al
AL DragAndDropFactbox.Pag...
AL LinksFactBox.Page.al
AL LinksList.Page.al
AL MappingList.Page.al
AL SharePointSetup.Page.al

```

Figura 5.20: Páginas

- **Document Type List:** Esta página de tipo List permite a los usuarios visualizar y gestionar los tipos de documentos configurados en el sistema. Su principal función es enumerar todos los tipos de documentos que se pueden asignar a diversos archivos en SharePoint. En la Figura 5.21 se muestra la estructura de la página.

```

1 page 50100 "Document Type List"
2 {
3   Caption = 'Document Type List', Comment = 'ESP="Lista tipo documentos"';
4   PageType = List;
5   ApplicationArea = All;
6   UsageCategory = Lists;
7   SourceTable = "Document Type";
8
9   layout
10  {
11    0 references
12    area(Content)
13    {
14      0 references
15      repeater("Document Type List")
16      {
17        Caption = 'Document Type List', Comment = 'ESP="Lista tipo documento"';
18        0 references
19        field(Code; Rec.Code)
20        {
21          Tooltip = 'Specifies the value of the Code field.', Comment = 'ESP="Especifica el valor del campo Código.';
22        }
23        0 references
24        field(Description; Rec.Description)
25        {
26          Tooltip = 'Specifies the value of the Description field.', Comment = 'ESP="Especifica el valor del campo Descripción.';
27        }
28      }
29    }
30  }
31 }

```

Figura 5.21: Página Document Type List.

- **DragAndDrop Factbox:** Esta CardPart permite la interacción con archivos mediante operaciones de arrastrar y soltar (drag and drop). Es una herramienta útil para subir documentos a SharePoint de forma intuitiva y directa desde la interfaz de usuario. En las Figuras 5.22 hasta 5.24 se observa la estructura de la página de Drag & Drop.

```

AL DragAndDropFactbox.Page.al X
TFG > src > page > AL DragAndDropFactbox.Page.al > Page 50104 "DragAndDrop Factbox"
11 references
1 page 50104 "DragAndDrop Factbox"
2 {
3     Caption = 'SharePoint', Comment = 'ESP="SharePoint"';
4     PageType = CardPart;
5
6     layout
7     {
8         0 references
9         area(content)
10        {
11            0 references
12            group(Control)
13            {
14                Caption = 'Drag & Drop', Comment = 'Drag & Drop';
15                ShowCaption = false;
16                0 references
17                usercontrol(FileDragAndDrop; "File Drag and Drop")
18                {
19                    ApplicationArea = All;
20
21                    1 reference
22                    trigger OnDrop(FileName: Text; FileData: Text)
23                    var
24                        TempBlob: Codeunit "Temp Blob";
25                        Base64Convert: Codeunit "Base64 Convert";
26                        DocumentTypeList: Page "Document Type List";
27                        FileOutStream: OutStream;
28                        FileInStream: InStream;
29
30                    begin
31                        DocumentTypeList.LookupMode(true);
32                        if DocumentTypeList.RunModal() = Action::LookupOK then begin
33                            DocumentTypeList.GetRecord(DocumentType);
34
35                            TempBlob.CreateOutStream(FileOutStream);
36                            Base64Convert.FromBase64(FileData, FileOutStream);
37                            TempBlob.CreateInStream(FileInStream);

```

Figura 5.22: Página Drag & Drop 1.

```

33
34
35     Links.CreateLink(FileInStream, GlobalTableId, GlobalRecordId, DocumentType.Code, FileName);
36 end;
37
38 1 reference
39 trigger OnClick()
40 var
41     DocumentTypeList: Page "Document Type List";
42     FileInStream: InStream;
43     FileName: Text;
44     UploadFileLbl: Label 'Upload file', Comment = 'ESP="Cargar archivo"';
45     AllFilesLbl: Label 'All files (*.*)|*.*', Locked = true;
46
47 begin
48     if UploadIntoStream(UploadFileLbl, '', AllFilesLbl, FileName, FileInStream) then
49         DocumentTypeList.LookupMode(true);
50     if DocumentTypeList.RunModal() = Action::LookupOK then begin
51         DocumentTypeList.GetRecord(DocumentType);
52
53         Links.CreateLink(FileInStream, GlobalTableId, GlobalRecordId, DocumentType.Code, FileName);
54     end;
55 end;
56 }
57 }
58

```

Figura 5.23: Página Drag & Drop 2.

```

10 references
59 procedure SetSource(TableId: Integer; RecId: RecordId)
60 begin
61     GlobalTableId := TableId;
62     GlobalRecordId := RecId;
63 end;
64
65 var
66     Links: Record Links;
67     DocumentType: Record "Document Type";
68     GlobalRecordId: RecordId;
69     GlobalTableId: Integer;
70 }

```

Figura 5.24: Página Drag & Drop 3.

- Links Factbox:** Esta ListPart muestra los enlaces a documentos almacenados en Share-Point asociados a registros específicos en Business Central. Permite a los usuarios acceder rápidamente a los documentos vinculados, ofreciendo funcionalidades de apertura, descarga y eliminación de los mismos. En las Figuras 5.25 hasta 5.27 se observa la estructura de la factbox de Links.

```

AL LinksFactBox.Page.al x
TFG > src > page > AL LinksFactBox.Page.al > Page 50105 "Links FactBox"
11 references
1 page 50105 "Links FactBox"
2 {
3     Caption = 'Links', Comment = 'ESP="Links"';
4     PageType = ListPart;
5     SourceTable = Links;
6     Editable = false;
7     ApplicationArea = All;
8
9     layout
10    {
11        0 references
12        area(content)
13        {
14            0 references
15            repeater(FactBoxList)
16            {
17                ShowCaption = false;
18                0 references
19                field(URL; Rec.URL)
20                {
21                    Tooltip = 'Specifies the value of the URL field.', Comment = 'ESP="Especifica el valor del campo URL."';
22                    Visible = true;
23                }
24                0 references
25                field(Description; Rec.Description)
26                {
27                    Tooltip = 'Specifies the value of the Description field.', Comment = 'ESP="Especifica el valor del campo Descripción."';
28                }
29                0 references
30                field("Document Type"; Rec."Document Type")
31                {
32                    Tooltip = 'Specifies the value of the Document Type field.', Comment = 'ESP="Especifica el valor del campo Tipo documento."';
33                }
34            }
35        }
36    }
37 }

```

Figura 5.25: Página Links FactBox 1.

```

32
33 actions
34 {
35     0 references
36     area(Processing)
37     {
38         0 references
39         action(OpenFile)
40         {
41             ApplicationArea = All;
42             Caption = 'Open the file', Comment = 'ESP="Abrir el fichero"';
43             Image = Open;
44             Tooltip = 'Open the file in SharePoint', Comment = 'ESP="Abrir el fichero en Sharepoint"';
45
46             0 references
47             trigger OnAction()
48             begin
49                 Rec.OpenLink();
50             end;
51
52             0 references
53             action(DownloadFile)
54             {
55                 ApplicationArea = All;
56                 Caption = 'Download the file', Comment = 'ESP="Descargar el fichero"';
57                 Image = Download;
58                 Tooltip = 'Download the SharePoint file to your computer', Comment = 'ESP="Descargar el fichero de SharePoint en nuestro equipo"';
59
60                 0 references
61                 trigger OnAction()
62                 begin
63                     Rec.DownloadLink();
64                 end;
65             }
66         }
67     }
68     0 references

```

Figura 5.26: Página Links FactBox 2.

```

62     0 references
63     action(Delete)
64     {
65         ApplicationArea = All;
66         Caption = 'Delete the file', Comment = 'ESP="Eliminar el fichero"';
67         Image = Delete;
68         Tooltip = 'Delete the file located in SharePoint', Comment = 'ESP="Eliminar el fichero ubicado en SharePoint"';
69
70         0 references
71         trigger OnAction()
72         begin
73             Rec.DeleteLink();
74         end;
75     }
76
77     10 references
78     procedure SetSource(RecId: RecordId)
79     begin
80         GlobalRecordId := RecId;
81         Rec.SetRange("Record Id", RecId);
82
83         CurrPage.Update();
84     end;
85
86     var
87         1 reference
88         GlobalRecordId: RecordId;
89     }

```

Figura 5.27: Página Links FactBox 3.

- Links List:** Esta página de tipo List es utilizada para administrar todos los enlaces creados entre los documentos de Business Central y SharePoint. Proporciona una vista completa de todos los enlaces almacenados, permitiendo acciones como la edición y eliminación. En las Figuras 5.28 hasta 5.30 se observa la estructura de la página Links List.

```

AL LinksList.Page.al X
TFG > src > page > AL LinksList.Page.al > Page 50102 "Links List"
3 references
1 page 50102 "Links List"
2 {
3   Caption = 'Links List', Comment = 'ESP="Lista links"';
4   PageType = List;
5   SourceTable = Links;
6   ApplicationArea = All;
7   UsageCategory = Lists;
8
9   layout
10  {
11    0 references
12    area(Content)
13    {
14      0 references
15      repeater(Links)
16      {
17        0 references
18        field("Link Id"; Rec."Link Id")
19        {
20          Tooltip = 'Specifies the value of the Link Id field.', Comment = 'ESP="Especifica el valor del campo Id. vínculo.';
21          ApplicationArea = All;
22        }
23        0 references
24        field("Record Id"; Rec."Record Id")
25        {
26          Tooltip = 'Specifies the value of the Record Id field.', Comment = 'ESP="Especifica el valor del campo Id. registro.';
27          ApplicationArea = All;
28        }
29        0 references
30        field(URL; Rec.URL)
31        {
32          Tooltip = 'Specifies the value of the URL field.', Comment = 'ESP="Especifica el valor del campo URL.';
33          ApplicationArea = All;
34        }
35      }
36    }
37  }
38 }

```

Figura 5.28: Página Links List 1.

```

30 field(Description; Rec.Description)
31 {
32   Tooltip = 'Specifies the value of the Description field.', Comment = 'ESP="Especifica el valor del campo Descripción.';
33   ApplicationArea = All;
34 }
35 0 references
36 field(Created; Rec.Created)
37 {
38   Tooltip = 'Specifies the value of the Created field.', Comment = 'ESP="Especifica el valor del campo Creado.';
39   ApplicationArea = All;
40 }
41 0 references
42 field("User Id"; Rec."User Id")
43 {
44   Tooltip = 'Specifies the value of the User Id field.', Comment = 'ESP="Especifica el valor del campo Id. usuario.';
45   ApplicationArea = All;
46 }
47 0 references
48 field("Document Type"; Rec."Document Type")
49 {
50   Tooltip = 'Specifies the value of the Document Type field.', Comment = 'ESP="Especifica el valor del campo Tipo documento.';
51   ApplicationArea = All;
52 }
53 }

```

Figura 5.29: Página Links List 2.

```

54 trigger OnOpenPage()
55     TempSharepointFile: Record "SharePoint File" temporary;
56     TempSharepointFolder: Record "SharePoint Folder" temporary;
57     SharepointManagement: Codeunit "Sharepoint Management";
58     begin
59         if ParentFolderURL <> '' then
60             SharepointManagement.GetFilesFromServerRelativeURL(ParentFolderURL, TempSharepointFolder, TempSharepointFile);
61             ParentFolderURL := SharepointManagement.GetDocumentsRootFiles(TempSharepointFolder, TempSharepointFile);
62
63         if TempSharepointFolder.FindSet() then
64             repeat
65                 Rec.Init();
66                 Rec.Description := TempSharepointFolder.Name;
67                 Rec.URL := TempSharepointFolder."Server Relative Url";
68                 Rec.Insert();
69             until TempSharepointFolder.Next() = 0;
70
71         if TempSharepointFile.FindSet() then
72             repeat
73                 Rec.Init();
74                 Rec.Description := TempSharepointFile.Name;
75                 Rec.URL := TempSharepointFile."Server Relative Url";
76                 Rec.Insert();
77             until TempSharepointFile.Next() = 0;
78         end;
79     end;
80
81     0 references
82     procedure SetParentFolderURL(NewURL: Text)
83     begin
84         ParentFolderURL := NewURL;
85     end;
86
87     var
88         4 references
89         ParentFolderURL: Text;

```

Figura 5.30: Página Links List 3.

- Mapping List:** Esta página de tipo List se utiliza para configurar y visualizar las reglas de mapeo entre los registros de Business Central y las estructuras de carpetas en SharePoint. Es crucial para la correcta organización y almacenamiento de documentos en entornos de SharePoint. En las Figuras 5.31 hasta 5.33 se observa la estructura de la página Mapping.

```

AL MappingList.Page.al X
TFG > src > page > AL MappingList.Page.al > Page 50101 "Mapping List" > layout > { } Area Content > [ ] Repeater Mapping > Field "Table Id": Integer
3 references
1 page 50101 "Mapping List"
2 {
3     Caption = 'Mapping List', Comment = 'ESP="Lista mapeos"';
4     PageType = List;
5     ApplicationArea = All;
6     UsageCategory = Lists;
7     SourceTable = Mapping;
8
9     layout
10    {
11        0 references
12        area(Content)
13        {
14            0 references
15            repeater(Mapping)
16            {
17                0 references
18                field("Table Id"; Rec."Table Id")
19                {
20                    Tooltip = 'Specifies the value of the Table Id field.', Comment = 'ESP="Especifica el valor del campo Id. tabla.';
21                }
22                0 references
23                field("Table Name"; Rec."Table Name")
24                {
25                    Tooltip = 'Specifies the value of the Table Name field.', Comment = 'ESP="Especifica el valor del campo Nombre tabla.';
26                }
27                0 references
28                field("Site Name"; Rec."Site Name")
29                {
30                    Tooltip = 'Specifies the value of the Site Name field.', Comment = 'ESP="Especifica el valor del campo Nombre sitio.';
31                }
32                0 references
33                field("Folder Name"; Rec."Folder Name")
34                {
35                    Tooltip = 'Specifies the value of the Folder Name field.', Comment = 'ESP="Especifica el valor del campo Nombre carpeta.';
36                }
37            }
38        }
39    }
40 }

```

Figura 5.31: Página Mapping 1.

```

31 0 references
32 field("Create Record Folder"; Rec."Create Record Folder")
33 {
34     Tooltip = 'Specifies the value of the Create Record Folder field.', Comment = 'ESP="Especifica el valor del campo Crear subcarpeta de';
35 }
36 0 references
37 field("Folder Reference Field"; Rec."Folder Reference Field")
38 {
39     Tooltip = 'Specifies the value of the Folder Reference Field field.', Comment = 'ESP="Especifica el valor del campo Campo referencia';
40 }
41 0 references
42 field("Folder Reference Field Name"; Rec."Folder Reference Field Name")
43 {
44     Tooltip = 'Specifies the value of the Folder Reference Field caption field.', Comment = 'Especifica el valor del campo Nombre campo r';
45 }
46 0 references
47 field("Include Document Type"; Rec."Include Document Type")
48 {
49     Tooltip = 'Specifies the value of the Include Document Type field.', Comment = 'ESP="Especifica el valor del campo Incluir tipo docum';
50 }
51 0 references
52 field("Pattern File Name"; Rec."Pattern File Name")
53 {
54     Tooltip = 'Specifies the value of the Pattern File Name field.\%1 - Pattern field 1\%2 - Pattern field 2\%3 - File name', Comment = '
55 }
56 0 references
57 field("Field 1 Pattern"; Rec."Field 1 Pattern")
58 {
59     Tooltip = 'Specifies the value of the Field 1 Pattern field.', Comment = 'ESP="Especifica el valor del campo Campo 1 patrón.';
60 }
61 0 references

```

Figura 5.32: Página Mapping 2.

```

55     field("Field 1 Pattern Name"; Rec."Field 1 Pattern Name")
56     {
57         Tooltip = 'Specifies the value of the Field 1 Pattern caption field.', Comment = 'Nombre campo 1 patrón';
58     }
59     field("Field 2 Pattern"; Rec."Field 2 Pattern")
60     {
61         Tooltip = 'Specifies the value of the Field 2 Pattern field.', Comment = 'ESP="Especifica el valor del campo Campo 2 patrón.';
62     }
63     field("Field 2 Pattern Name"; Rec."Field 2 Pattern Name")
64     {
65         Tooltip = 'Specifies the value of the Field 2 Pattern caption field.', Comment = 'Nombre campo 2 patrón';
66     }
67 }
68 }
69 }
70 }

```

Figura 5.33: Página Mapping 3.

- **SharePoint Setup:** Esta página de tipo Card es esencial para la configuración inicial y la gestión de la conexión con SharePoint. Permite a los administradores establecer y modificar los parámetros necesarios para la integración con SharePoint. En las Figuras 5.34 y 5.35 se representa la estructura de la página de SharePoint Setup.

```

AL SharePointSetup.Page.al
1  page 50103 "SharePoint Setup"
2  {
3      Caption = 'SharePoint Setup', Comment = 'ESP="Configuración Sharepoint"';
4      PageType = Card;
5      ApplicationArea = All;
6      UsageCategory = Administration;
7      SourceTable = "SharePoint Setup";
8      DeleteAllowed = false;
9      InsertAllowed = false;
10     ModifyAllowed = true;
11
12     layout
13     {
14         area(Content)
15         {
16             group(General)
17             {
18                 Caption = 'General', Comment = 'ESP="General"';
19                 field("SharePoint URL"; Rec."SharePoint URL")
20                 {
21                     Tooltip = 'Specifies the value of the SharePoint URL field.', Comment = 'ESP="Especifica el valor del campo URL Sharepoint.';
22                 }

```

Figura 5.34: Página SharePoint 1.

```

23 |         0 references
24 |         field("Client Id"; Rec."Client Id")
25 |         {
26 |             Tooltip = 'Specifies the value of the Client Id field.', Comment = 'ESP="Especifica el valor del campo Id. cliente."';
27 |         }
28 |         0 references
29 |         field("Client Secret"; Rec."Client Secret")
30 |         {
31 |             Tooltip = 'Specifies the value of the Client Secret field.', Comment = 'ESP="Especifica el valor del campo Secreto cliente."';
32 |             ExtendedDatatype = Masked;
33 |         }
34 |     }
35 | }
36 |
37 | 0 references
38 | trigger OnOpenPage()
39 | begin
40 |     if not Rec.Get() then
41 |         Rec.Insert();
42 | end;

```

Figura 5.35: Página SharePoint 2.

### Extensiones de páginas

Las extensiones de página se utilizan para modificar las páginas estándar de Business Central, añadiendo funcionalidades específicas sin alterar el código base original. Por ejemplo, las extensiones de página para 'Contact Card' y 'Customer Card' integran factboxes de arrastrar y soltar, y de enlaces, mejorando la interacción del usuario con documentos y datos externos. Estas extensiones se organizan en archivos con la nomenclatura NombreObjeto.PageExt.al, asegurando una estructura clara y coherente.

Todas las extensiones que hemos creado en nuestro proyecto van a tener la misma funcionalidad, permitir gestionar documentos y enlaces para múltiples clientes, facturas ..., y todas las extensiones tendrán también la misma estructura la cual podemos observar en la Figura 5.36.

```

AL CustomerCardExt.PageExt.al
TFG > src > pageextension > AL CustomerCardExt.PageExt.al > ...
1 | pageextension 50100 "Customer Card Ext." extends "Customer Card"
2 | {
3 |     layout
4 |     {
5 |         addfirst(factboxes)
6 |         {
7 |             1 reference
8 |             part(DragAndDropFactbox; "DragAndDrop Factbox")
9 |             {
10 |                 ApplicationArea = All;
11 |             }
12 |             1 reference
13 |             part(LinksFactBox; "Links FactBox")
14 |             {
15 |                 ApplicationArea = All;
16 |             }
17 |         }
18 |
19 | 0 references
20 | trigger OnAfterGetCurrRecord()
21 | begin
22 |     CurrPage.DragAndDropFactbox.Page.SetSource(Database::Customer, Rec.RecordId);
23 |     CurrPage.LinksFactBox.Page.SetSource(Rec.RecordId);
24 | end;

```

Figura 5.36: Extensión de página.

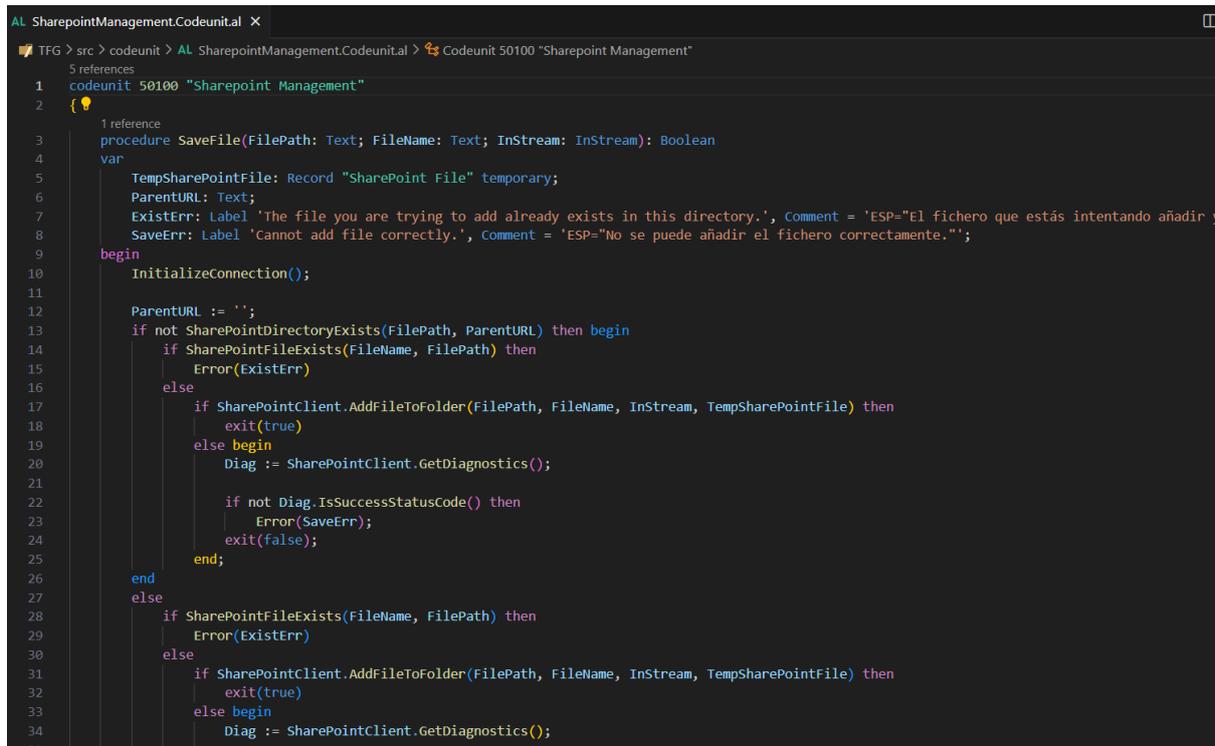
## Codeunits

Se desarrolla la codeunit "SharePoint Management", diseñada para facilitar la integración y gestión de archivos entre Business Central y SharePoint. Esta codeunit centraliza las funciones necesarias para interactuar con SharePoint, asegurando que los archivos y directorios se gestionen de manera eficiente y segura.

Esta codeunit posee unas funcionalidades principales:

- **Gestión de archivos y directorios:** Guardar el fichero y comprobar si este existe ya en SharePoint, o si el directorio donde la queremos guardar existe.
- **Operaciones de archivo:** Descargar y borrar el fichero.
- **Obtención de información de directorios:** Para recuperar archivos y subcarpetas dentro de un directorio en SharePoint.

La codeunit se encarga de inicializar la conexión con SharePoint y manejar la autenticación utilizando OAuth (realizar las operaciones bajo un marco seguro), y también implementa un manejo de errores para asegurar cualquier fallo en la comunicación. La estructura de la codeunit se encuentra representada en las Figuras 5.37 hasta 5.44.



```
AL SharepointManagement.Codeunit1al X
TFG > src > codeunit > AL SharepointManagement.Codeunit1al > Codeunit 50100 "Sharepoint Management"
5 references
1 codeunit 50100 "Sharepoint Management"
2 {
3     1 reference
4     procedure SaveFile(FilePath: Text; FileName: Text; InStream: InStream): Boolean
5     var
6         TempSharePointFile: Record "SharePoint File" temporary;
7         ParentURL: Text;
8         ExistErr: Label 'The file you are trying to add already exists in this directory.', Comment = 'ESP="El fichero que estás intentando añadir y
9         SaveErr: Label 'Cannot add file correctly.', Comment = 'ESP="No se puede añadir el fichero correctamente.';
10    begin
11        InitializeConnection();
12
13        ParentURL := '';
14        if not SharePointDirectoryExists(FilePath, ParentURL) then begin
15            if SharePointFileExists(FileName, FilePath) then
16                Error(ExistErr)
17            else
18                if SharePointClient.AddFileToFolder(FilePath, FileName, InStream, TempSharePointFile) then
19                    exit(true)
20                else begin
21                    Diag := SharePointClient.GetDiagnostics();
22
23                    if not Diag.IsSuccessStatusCode() then
24                        Error(SaveErr);
25                    exit(false);
26                end;
27            end
28        else
29            if SharePointFileExists(FileName, FilePath) then
30                Error(ExistErr)
31            else
32                if SharePointClient.AddFileToFolder(FilePath, FileName, InStream, TempSharePointFile) then
33                    exit(true)
34                else begin
35                    Diag := SharePointClient.GetDiagnostics();
```

Figura 5.37: Codeunit 1.

```

35
36         if not Diag.IsSuccessStatusCode() then
37             Error(SaveErr);
38             exit(false);
39         end;
40     end;
41
42     4 references
43     procedure SharePointDirectoryExists(DirectoryPath: Text; ParentURL: Text): Boolean
44     var
45         TempSharePointFolder: Record "SharePoint Folder" temporary;
46         FolderName: Text;
47         RelativeURL: Text;
48     begin
49         if ParentURL = '' then
50             DirectoryPath := DirectoryPath;
51
52         if DirectoryPath = '' then
53             exit(true);
54
55         if StrPos(DirectoryPath, '/') = 0 then begin
56             FolderName := DirectoryPath;
57             DirectoryPath := '';
58         end else begin
59             FolderName := CopyStr(DirectoryPath, 1, StrPos(DirectoryPath, '/') - 1);
60             DirectoryPath := CopyStr(DirectoryPath, StrPos(DirectoryPath, '/') + 1);
61         end;
62
63         RelativeURL := ParentURL + '/' + FolderName;
64

```

Figura 5.38: Codeunit 2.

```

65         if SharePointClient.GetSubFoldersByServerRelativeUrl(ParentURL, TempSharePointFolder) then begin
66             TempSharePointFolder.SetRange(TempSharePointFolder.Name, FolderName);
67             if TempSharePointFolder.FindFirst() then begin
68                 RelativeURL := TempSharePointFolder."Server Relative Url";
69                 exit(SharePointDirectoryExists(DirectoryPath, RelativeURL));
70             end else begin
71                 SharePointClient.CreateFolder(RelativeUrl, TempSharepointFolder);
72                 exit(SharePointDirectoryExists(DirectoryPath, RelativeURL));
73             end;
74         end else begin
75             SharePointClient.CreateFolder(RelativeUrl + '/' + FolderName, TempSharepointFolder);
76             exit(SharePointDirectoryExists(DirectoryPath, RelativeURL));
77         end;
78     end;
79
80     3 references
81     procedure SharePointFileExists(FileName: Text; FilePath: text): Boolean
82     var
83         SharepointFile: Record "SharePoint File";
84         TempSharePointList: Record "SharePoint List" temporary;
85         TempSharePointFolder: Record "SharePoint Folder" temporary;
86     begin
87         InitializeConnection();
88         if not FilePath.StartsWith('/sites') then
89             if SharePointClient.GetLists(TempSharePointList) then begin
90                 TempSharePointList.SetRange(Title, 'Documents');
91                 if TempSharePointList.FindFirst() then
92                     if SharePointClient.GetDocumentLibraryRootFolder(TempSharePointList.OdataId,
93                         TempSharepointFolder) then
94                         if StrPos(FilePath, '/') = 1 then
95                             FilePath := TempSharepointFolder."Server Relative Url" + FilePath
96                         else
97                             FilePath := TempSharepointFolder."Server Relative Url" + '/' + FilePath;

```

Figura 5.39: Codeunit 3.

```

97     end;
98     SharePointClient.GetFolderFilesByServerRelativeUrl(FilePath, SharePointFile);
99     SharepointFile.SetRange(Name, FileName);
100    if not SharepointFile.IsEmpty then
101        exit(true);
102    end;
103
104    1 reference
105    procedure DownloadFile(FileDirectory: Text)
106    var
107        TempSharePointFile: Record "SharePoint File" temporary;
108        FileName: Text;
109    begin
110        InitializeConnection();
111
112        FileName := FileManagement.GetFileName(FileDirectory);
113        if FileName = '' then
114            Error(FileNotFounderErr);
115
116        FileDirectory := FileManagement.GetDirectoryName(FileDirectory);
117
118        if not SharePointClient.GetFolderFilesByServerRelativeUrl(FileDirectory, TempSharePointFile) then
119            Error(FileNotFounderErr);
120
121        TempSharePointFile.SetRange(Name, FileName);
122        if TempSharePointFile.FindFirst() then
123            SharePointClient.DownloadFileContent(TempSharePointFile.OdataId, FileName)
124        else
125            Error(FileNotFounderErr);
126    end;
127

```

Figura 5.40: Codeunit 4.

```

128    1 reference
129    procedure DeleteFile(FileDirectory: Text; LinkId: Integer)
130    var
131        TempSharePointFile: Record "SharePoint File" temporary;
132        Links: Record Links;
133        FileName: Text;
134        FilePath: Text;
135    begin
136        InitializeConnection();
137
138        FileName := FileManagement.GetFileName(FileDirectory);
139        FilePath := FileManagement.GetDirectoryName(FileDirectory);
140
141        if not SharePointFileExists(FileName, FilePath) then begin
142            if Links.Get(LinkId) then
143                Links.Delete();
144        end else begin
145            if not SharePointClient.GetFolderFilesByServerRelativeUrl(FilePath, TempSharePointFile) then
146                Error(FileNotFounderErr);
147
148            TempSharePointFile.SetRange(Name, FileName);
149            if TempSharePointFile.FindFirst() then
150                SharePointClient.DeleteFileByServerRelativeUrl(TempSharePointFile."Server Relative Url")
151            else
152                Error(FileNotFounderErr);
153    end;
154

```

Figura 5.41: Codeunit 5.

```

155 procedure GetDocumentsRootFiles(var SharePointFolder: Record "SharePoint Folder" temporary; var SharepointFile: Record "SharePoint File"): Text
156 var
157     TempSharePointList: Record "SharePoint List" temporary;
158 begin
159     InitializeConnection();
160     if SharePointClient.GetLists(TempSharePointList) then begin
161         TempSharePointList.SetRange(Title, 'Documents');
162         if TempSharePointList.FindFirst() then begin
163             if SharePointClient.GetDocumentLibraryRootFolder(TempSharePointList.OdataId, SharePointFolder) then
164                 SharePointClient.GetFolderFilesByServerRelativeUrl(SharePointFolder."Server Relative Url", SharepointFile);
165             SharePointClient.GetSubFoldersByServerRelativeUrl(SharePointFolder."Server Relative Url", SharePointFolder);
166             exit(SharePointFolder."Server Relative Url");
167         end;
168     end;
169 end;
170
171 1 reference
172 procedure GetFilesFromServerRelativeURL(ServerRelativeURL: Text; var SharepointFolder: Record "SharePoint Folder" temporary; var SharePointFile:
173 begin
174     InitializeConnection();
175     SharePointClient.GetFolderFilesByServerRelativeUrl(ServerRelativeURL, SharePointFile);
176     SharePointClient.GetSubFoldersByServerRelativeUrl(ServerRelativeURL, SharepointFolder);
177 end;

```

Figura 5.42: Codeunit 6.

```

178 0 references
179 local procedure InitializeConnection()
180 var
181     TempSharePointList: Record "SharePoint List" temporary;
182     AadTenantId: Text;
183 begin
184     if Connected then
185         exit;
186     SharePointSetup.Get();
187
188     AadTenantId := GetAadTenantNameFromBaseUrl(SharePointSetup."SharePoint URL");
189     SharePointClient.Initialize(SharePointSetup."SharePoint URL", GetSharePointAuthorization(AadTenantId));
190
191     SharePointClient.GetLists(TempSharePointList);
192
193     Diag := SharePointClient.GetDiagnostics();
194
195     if (not Diag.IsSuccessStatusCode()) then
196         Error(DiagErr, Diag.GetErrorMessage())
197     else
198         Connected := true;
199 end;
200
201 1 reference
202 local procedure GetSharePointAuthorization(AadTenantId: Text): Interface "SharePoint Authorization"
203 var
204     SharePointAuth: Codeunit "SharePoint Auth.";
205     Scopes: List of [Text];
206 begin
207     SharePointSetup.Get();
208     Scopes.Add('https://innova1.sharepoint.com/.default');
209     exit(SharePointAuth.CreateAuthorizationCode(AadTenantId, SharePointSetup."Client Id", SharePointSetup."Client Secret", Scopes))
210 end;

```

Figura 5.43: Codeunit 7.

```

211 local procedure GetAadTenantNameFromBaseUrl(BaseUrl: Text): Text
212 var
213     Uri: Codeunit Uri;
214     MySiteHostSuffixTxt: Label '-my.sharepoint.com', Locked = true;
215     SharePointHostSuffixTxt: Label '.sharepoint.com', Locked = true;
216     OnMicrosoftTxt: Label '.onmicrosoft.com', Locked = true;
217     UriInvalidErr: Label 'The Base Url %1 does not seem to be a valid SharePoint Online Url.', Comment = 'ESP="La Url Base %1 no parece ser una
218     Host: Text;
219 begin
220     Uri.Init(BaseUrl);
221     Host := Uri.GetHost();
222     if not Host.EndsWith(SharePointHostSuffixTxt) then
223         Error(UriInvalidErr);
224     if Host.EndsWith(MySiteHostSuffixTxt) then
225         exit(CopyStr(Host, 1, StrPos(Host, MySiteHostSuffixTxt) - 1) + OnMicrosoftTxt);
226     exit(CopyStr(Host, 1, StrPos(Host, SharePointHostSuffixTxt) - 1) + OnMicrosoftTxt);
227 end;
228
229 var
230     6 references SharePointSetup: Record "SharePoint Setup";
231     23 references SharePointClient: Codeunit "SharePoint Client";
232     4 references FileManagement: Codeunit "File Management";
233
234     7 references Diag: Interface "HTTP Diagnostics";
235     2 references Connected: Boolean;
236     1 reference DiagErr: Label 'Sharepoint Management error: \\%1', Comment = 'ESP="Sharepoint Management error: \\%1";
237     5 references FileNotFounderErr: Label 'File not found.', Comment = 'ESP="Fichero no encontrado.';
238 }

```

Figura 5.44: Codeunit 8.

## ControlAddIns

Para la funcionalidad de arrastrar y soltar, se implementan ControlAddIns que permiten integrar componentes de UI personalizados en las páginas de Business Central. Estos ControlAddIns incluyen scripts de JavaScript y hojas de estilo CSS para manejar la interactividad y el estilo visual, respectivamente. Los scripts gestionan eventos como 'dragover' y 'drop', mientras que el CSS define aspectos como la imagen de fondo y la opacidad. Los ControlAddIns se organizan en carpetas específicas y se referencian en archivos AL con la nomenclatura NombreObjeto.ControlAddin.al, lo que facilita su gestión y actualización.

- **Script.js:** JavaScript encargado de manejar los eventos de interacción del usuario como 'drag', 'drop' y 'click'. El script se representa en las Figuras 5.45 hasta 5.47.

```
JS Script.js X
TFG > src > controladdin > FDDFileDragandDrop > JS Script.js > ...
1  var controlAddIn
2
3  function init() {
4      controlAddIn = document.getElementById('controlAddIn');
5
6      controlAddIn.addEventListener('dragover', handleDragOver, false)
7      controlAddIn.addEventListener('dragleave', handleDragLeave, false);
8      controlAddIn.addEventListener("drop", handleDrop);
9      controlAddIn.addEventListener("click", handleClick);
10 }
11
12 function handleDragOver(evt) {
13     evt.stopPropagation();
14     evt.preventDefault();
15     controlAddIn.style["opacity"] = "0.5";
16 }
17
18 function handleDragLeave(evt) {
19     evt.stopPropagation();
20     evt.preventDefault();
21     controlAddIn.style["opacity"] = "1";
22 }
23
24 function handleDrop(evt) {
25     evt.stopPropagation();
26     evt.preventDefault();
27
28     var files = evt.dataTransfer.files;
29
30     f = files[0];
31     var reader = new FileReader();
32     reader.onload = function(e) {
33         FileData = reader.result;
34         FileDataEncoded = btoa(FileData);
35     }
36 }
```

Figura 5.45: Script.js 1.

```
36     FileData = reader.result;
37     FileDataEncoded = btoa(FileData);
38     Microsoft.Dynamics.NAV.InvokeExtensibilityMethod('OnDrop', [FileName, FileDataEncoded]);
39 }
40 }
41 reader.readAsBinaryString(f);
42 }
43
44 function handleDropMulti(evt) {
45     evt.stopPropagation();
46     evt.preventDefault();
47
48     var files = evt.dataTransfer.files;
49
50     for (var i = 0; i < files.length; i++) {
51         var f = files[i];
52         var reader = new FileReader();
53         reader.onload = function(e) {
54             FileData = reader.result;
55             FileDataEncoded = btoa(FileData);
56
57             FileName = f.name;
58             controlAddIn.style["opacity"] = "1";
59
60             Microsoft.Dynamics.NAV.InvokeExtensibilityMethod('OnDrop', [FileName, FileDataEncoded]);
61         }
62     }
63     reader.readAsBinaryString(f);
64 }
```

Figura 5.46: Script.js 2.

```

66 function handleClick(evt) {
67     evt.stopPropagation();
68     evt.preventDefault();
69
70     Microsoft.Dynamics.NAV.InvokeExtensibilityMethod('onClick', null);
71 }

```

Figura 5.47: Script.js 3.

- **Startup.js:** Activa la función 'init()' definida en 'Script.js' cuando la página que contiene el control add-in se carga, asegurando que los eventos estén adecuadamente preparados y listos para responder a las interacciones del usuario desde el momento en el que este accede a la funcionalidad. Startup.js se encuentra representado en la Figura 5.48.



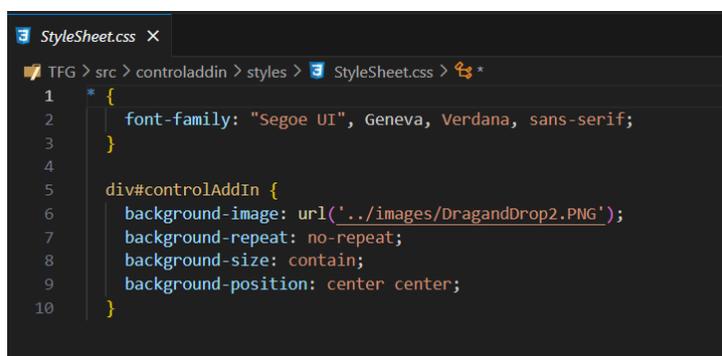
```

JS Startup.js x JS Script
TFG > src > controladdin
1 init();

```

Figura 5.48: Startup.js.

- **StyleSheet.css:** Define los estilos visuales del control Add-in, como la fuente y el fondo. El fondo del área de control se establece con una imagen específica, lo que mejora la interfaz de usuario visual para indicar que el área puede aceptar archivos arrastrados. Esta hoja de estilos está representada en la Figura 5.49.



```

StyleSheet.css x
TFG > src > controladdin > styles > StyleSheet.css > *
1 * {
2     font-family: "Segoe UI", Geneva, Verdana, sans-serif;
3 }
4
5 div#controlAddIn {
6     background-image: url('../images/DragandDrop2.PNG');
7     background-repeat: no-repeat;
8     background-size: contain;
9     background-position: center center;
10 }

```

Figura 5.49: StyleSheet.css.

- **FileDragAndDrop.ControlAddin:** Vincula los scripts y la hoja de estilos para su funcionamiento, así como la imagen de fondo utilizada para la representación visual del área de arraste. En la Figura 5.50 se representa el Controladdin creado.

```

AL FileDragandDrop.ControlAddin.al X
TFG > src > controladdin > FileDragandDrop > AL FileDragandDrop.ControlAddin.al > {} ControlAddin "File D
1 reference
1 controladdin "File Drag and Drop"
2 {
3     RequestedHeight = 200;
4     RequestedWidth = 400;
5
6     HorizontalStretch = true;
7     VerticalStretch = true;
8
9     HorizontalShrink = true;
10    VerticalShrink = true;
11
12    Scripts = 'src/controladdin/FDDFileDragandDrop/Script.js';
13    StyleSheets = 'src/controladdin/styles/StyleSheet.css';
14    StartupScript = 'src/controladdin/FDDFileDragandDrop/Startup.js';
15
16    Images = 'src/controladdin/images/DrageandDrop2.PNG';
17
18    1 reference
19    event OnDrop(FileName: Text; FileData: Text)
20    1 reference
    event OnClick();
}

```

Figura 5.50: ControlAddin.

### Permission Set

Los conjuntos de permisos definen los derechos de acceso y modificación que tienen los usuarios sobre las funcionalidades y datos del proyecto. Se establecen permisos detallados para operaciones sobre tablas, ejecución de codeunits y acceso a páginas, asegurando que solo los usuarios autorizados puedan realizar operaciones críticas. Los Permission Sets se definen en archivos con la nomenclatura NombreObjeto.PermissionSet.al, permitiendo una configuración clara y sistemática de los permisos en el entorno de Business Central. Se puede ver representado el fichero de los permisos en la Figura 5.51.

```

AL GeneratedPermission.PermissionSet.al X
TFG > src > permissionset > AL GeneratedPermission.PermissionSet.al > {}
0 references
1 permissionset 50100 GeneratedPermission
2 {
3     Assignable = true;
4     Permissions = tabledata "Document Type" = RIMD,
5         tabledata Links = RIMD,
6         tabledata Mapping = RIMD,
7         tabledata "SharePoint Setup" = RIMD,
8         table "Document Type" = X,
9         table Links = X,
10        table Mapping = X,
11        table "SharePoint Setup" = X,
12        codeunit "Sharepoint Management" = X,
13        page "Document Type List" = X,
14        page "DragAndDrop Factbox" = X,
15        page "Links FactBox" = X,
16        page "Links List" = X,
17        page "Mapping List" = X,
18        page "SharePoint Setup" = X;
19 }

```

Figura 5.51: PermissionSet.

### 5.3. Verificación y validación

En el desarrollo del sistema de integración entre Business Central y SharePoint, la verificación y validación de las funcionalidades implementadas han sido fundamentales para garantizar la calidad y el rendimiento del producto final. Las pruebas se organizan y realizan siguiendo una metodología estructurada para cubrir diferentes niveles y aspectos del sistema.

El proyecto llevado a cabo realiza varios tipos de pruebas, adecuadas a las necesidades y la estructura del sistema:

- **Pruebas de integración:** Se llevan a cabo pruebas de integración para verificar que diferentes partes del sistema trabajan correctamente en conjunto. Esto incluye la interacción entre tablas y codeunits, así como la correcta generación y manejo de eventos.
- **Pruebas de sistema:** Se centran en verificar que el sistema completo funciona según lo esperado. Estas pruebas evalúan el sistema en su conjunto, verificando que todas las funcionalidades implementadas operen correctamente bajo condiciones que simulan el uso real.

Para la ejecución eficiente de las pruebas, se emplean varias técnicas:

- **Uso de eventos en Codeunits:** Las codeunits de eventos como 'EventCreateLink', 'EventOpenLink', 'EventDownloadLink' y 'EventDeleteLink', son diseñadas para manipular la lógica de negocio sin alterar el código base, permitiendo una fácil extensibilidad y mantenimiento. Se testean para asegurar que los eventos se manejan correctamente. En las Figuras 5.52 hasta 5.55, se observan todos los eventos creados, y en las Figuras 5.56 hasta 5.58 se representa la codeunit para los tests.

```
1 codeunit 50202 EventCreateLink
2 {
3     [EventSubscriber(ObjectType::Table, Database::Links, 'OnBeforeCreateLink', '', false, false)]
4     local procedure MyBeforeCreateLinkSubscriber(SharePointPath: Text; SharePointFileName: Text; FileInStream: InStream; var Handled: Boolean)
5     begin
6         Handled := true;
7     end;
8 }
```

Figura 5.52: Event Create Link.

```
1 codeunit 50201 EventDeleteLink
2 {
3     [EventSubscriber(ObjectType::Table, Database::Links, 'OnBeforeDeleteLink', '', true, true)]
4     local procedure MyBeforeDeleteLinkSubscriber(var SharePointPathURL: Text; var LinkId: Integer; var Handled: Boolean)
5     begin
6         Handled := true;
7     end;
8 }
```

Figura 5.53: Event Delete Link.

```

0 references
1 codeunit 50203 EventDownloadLink
2 {
3     [EventSubscriber(ObjectType::Table, Database::Links, 'OnBeforeDownloadLink', '', false, false)]
4     local procedure MyBeforeCreateLinkSubscriber(SharePointPathURL: Text; var Handled: Boolean)
5     begin
6         Handled := true
7     end;
8 }

```

Figura 5.54: Event Download Link.

```

TFGTest > src > codeunit > AL EventOpenLink.Codeunit.al > Codeunit 50204 EventOpenLink
0 references
1 codeunit 50204 EventOpenLink
2 {
3     [EventSubscriber(ObjectType::Table, Database::Links, 'OnBeforeOpenLink', '', false, false)]
4     local procedure MyBeforeCreateLinkSubscriber(SharePointPathURL: Text; var Handled: Boolean)
5     begin
6         Handled := true;
7     end;
8 }

```

Figura 5.55: Event Open Link.

```

AL CodeUnitTest.Codeunit.al x
TFGTest > test > AL CodeUnitTest.Codeunit.al
0 references
1 codeunit 50200 CodeUnitTest
2 {
3     Subtype = Test;
4     TestPermissions = Disabled;
5
6     2 references
7     local procedure Initialize()
8     begin
9         Mapping.Init();
10        Mapping."Table Id" := Database::Customer;
11        Mapping."Site Name" := 'https://innova1.sharepoint.com/sites/tfg/';
12        Mapping."Folder Name" := 'Customer';
13        Mapping."Create Record Folder" := true;
14        Mapping."Folder Reference Field" := 1;
15        Mapping."Include Document Type" := true;
16        Mapping."Pattern File Name" := '%3';
17        Mapping."Field 1 Pattern" := 2;
18        Mapping."Field 2 Pattern" := 3;
19        if Mapping.Insert() then;
20
21        Customer.Init();
22        if Customer.Insert() then;
23            Customer.Name := 'LUIS';
24            Customer."Search Name" := 'LUIS';
25            Customer.Modify();

```

Figura 5.56: Codeunit Test 1.

```

28 [Test]
29 0 references
29 procedure TestCreateLink()
30 var
31   FileInStream: InStream;
32   GlobalTableId: Integer;
33   DocumentTypeCode: Code[10];
34   FileName: Text;
35 begin
36   Initialize();
37   // LibrarySales.CreateCustomer(customer);
38   Customer.FindFirst();
39
40   TempBlob.CreateInStream(FileInStream);
41
42   GlobalTableId := Database::Customer;
43
44   DocumentTypeCode := 'FACTURA';
45   FileName := 'Report.txt';
46   Links.CreateLink(FileInStream, GlobalTableId, Customer.RecordId, DocumentTypeCode, FileName);
47
48   Assert.IsTrue(Links.FindLast(), 'El enlace no se creó correctamente.');
```

Figura 5.57: Codeunit Test 2.

```

55 [Test]
56 0 references
56 procedure TestDeleteLink()
57 var
58   FileInStream: InStream;
59   GlobalTableId: Integer;
60   DocumentTypeCode: Code[10];
61   FileName: Text;
62   LinkId: Integer;
63 begin
64   Initialize();
65   // LibrarySales.CreateCustomer(customer);
66   Customer.FindFirst();
67   TempBlob.CreateInStream(FileInStream);
68   GlobalTableId := Database::Customer;
69   DocumentTypeCode := 'FACTURA';
70   FileName := 'Report.txt';
71
72   Links.CreateLink(FileInStream, GlobalTableId, Customer.RecordId, DocumentTypeCode, FileName);
73   if Links.FindLast() then
74     LinkId := Links."Link Id" + 1
75   else
76     LinkId := 1;
77   Assert.IsTrue(LinkId > 0, 'El enlace no se creó correctamente.');
```

Figura 5.58: Codeunit Test 3.

- Continuous Integration y Continuous Delivery (CI/CD):** Se integra el proyecto en un entorno CI/CD utilizando GitHub Actions, tal como se puede visualizar en las Figuras 5.59 y 5.60, lo que permite automatizar la compilación del código, la ejecución de pruebas y la implementación de cambios. Esto ayuda a mantener la calidad del código y a acelerar el ciclo de lanzamiento del software.

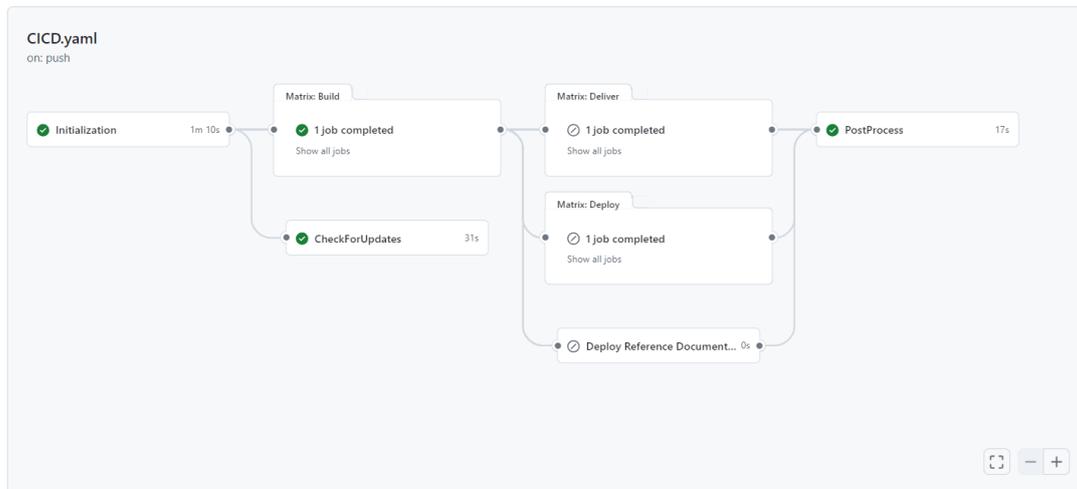


Figura 5.59: GitHub Actions CI/CD 1.

Build . (Default) / . (Default) summary

Test app	Tests	Passed	Failed	Skipped	Time
TFG TEST	2	2 <span style="color: green;">✔</span>			0.156s

No test failures

Job summary generated at run-time

Figura 5.60: GitHub Actions CI/CD 2.

Para reforzar la estrategia de pruebas y automatización del desarrollo, se integra AL-Go, una herramienta proporcionada por Microsoft específicamente diseñada para proyectos de desarrollo en Business Central. AL-Go ofrece un conjunto de plantillas y acciones de GitHub que facilitan la configuración y ejecución de procesos de CI/CD, crucial para mantener la calidad y eficiencia del desarrollo.

En el proyecto actual, AL-Go es fundamental para implementar prácticas de CI/CD. Se configuran acciones de GitHub que automáticamente inician flujos de trabajo de AL-Go tras cada 'push' al repositorio, ejecutando pruebas y compilaciones sin necesidad de intervención manual.

Se desarrolla un conjunto específico de datos de prueba para asegurar que el sistema maneja adecuadamente todas las condiciones posibles. Los datos de prueba se almacenan en una base de datos de prueba accesible para el equipo de desarrollo, permitiendo su revisión y reutilización en futuras fases del proyecto.

Las pruebas demuestran que el sistema cumple los requisitos especificados y proporciona una integración robusta y eficiente entre Business Central y SharePoint. La evaluación muestra que las funcionalidades como la creación, gestión y eliminación de enlaces son operativas y eficaces.

## Capítulo 6

# Conclusiones

En términos de resultados del proyecto, la implementación de la gestión de archivos en la nube con SharePoint para Microsoft Dynamics 365 Business Central ha culminado exitosamente, demostrando funcionalidad y adaptabilidad. Se estableció un sistema robusto que integró eficientemente SharePoint, mejorando significativamente tanto la gestión de archivos como la interacción con el ERP. Aunque no todos los objetivos iniciales se cumplieron debido a restricciones de tiempo y recursos, se priorizaron y se añadieron funcionalidades cruciales como la integración del factbox de funciones, lo que enriqueció considerablemente el proyecto.

Profesionalmente, este proyecto proporcionó una inmersión profunda en el mundo de los ERP y la oportunidad de colaborar con una empresa que prioriza la innovación y la eficiencia. La experiencia fue enormemente enriquecedora, ya que desarrollé competencias en consultoría, colaboración en equipo y gestión de clientes. Observar la organización de una empresa tecnológica y participar en reuniones y planificaciones basadas en propuestas solidificó mi entendimiento de un ambiente corporativo efectivo.

Personalmente, el proyecto fue un desafío significativo que me forzó a salir de mi zona de confort y adaptarme a un entorno empresarial lleno de nuevas caras y tecnologías. Representó un hito en mi carrera, marcando la transición de la universidad al mundo profesional. La acogida y el apoyo de la empresa fomentaron un ambiente cálido que facilitó esta transición, mientras que actividades de integración fortalecieron mi sentido de pertenencia al equipo.

A nivel formativo, adquirí conocimientos en tecnologías y herramientas avanzadas, incluyendo el lenguaje AL y técnicas de migración de datos, esenciales para la interacción efectiva con herramientas de Microsoft. La especialización en ERP, un campo con alta demanda y baja oferta, abre amplias oportunidades profesionales futuras, prometiendo una trayectoria ascendente.

A pesar de que la implementación completa de la funcionalidad de carga masiva no se logró, las mejoras implementadas superaron las expectativas iniciales, demostrando la capacidad del proyecto para adaptarse y evolucionar. El resultado final, aunque desafiante, ha validado el esfuerzo y la dedicación invertidos, culminando en una oferta para continuar mi desarrollo profesional con futuros proyectos en la empresa.



# Bibliografía

- [1] GitHub. Github. <https://github.com/>. [Consulta: 03 de Mayo de 2024].
- [2] Innova. Innova advanced consulting. [https://www.innovaconsulting.es/?gclid=CjwKCAjw88yxBhBWEiwa7cm6pUHfI--kXXgrY3ZFQEL-Z23m\\_zpAIHATHGauZCxiQ5Be7ce-B5bqIhoCU6wQAvD\\_BwE](https://www.innovaconsulting.es/?gclid=CjwKCAjw88yxBhBWEiwa7cm6pUHfI--kXXgrY3ZFQEL-Z23m_zpAIHATHGauZCxiQ5Be7ce-B5bqIhoCU6wQAvD_BwE). [Consulta: 03 de Mayo de 2024].
- [3] Lucid. Lucidchart. [https://www.lucidchart.com/pages/landing?utm\\_source=google&utm\\_medium=cpc&utm\\_campaign=\\_chart\\_ol\\_allcountries\\_mixed\\_search\\_brand\\_exact\\_&km\\_CPC\\_CampaignId=12085501855&km\\_CPC\\_AdGroupID=117687297058&km\\_CPC\\_Keyword=lucidchart&km\\_CPC\\_MatchType=e&km\\_CPC\\_ExtensionID=&km\\_CPC\\_Network=g&km\\_CPC\\_AdPosition=&km\\_CPC\\_Creative=491660231431&km\\_CPC\\_TargetID=kwd-33511936169&km\\_CPC\\_Country=9047055&km\\_CPC\\_Device=c&km\\_CPC\\_placement=&km\\_CPC\\_target=&gad\\_source=1&gclid=Cj0KCQjwltKxBhDMARIsAG8KnqVaaPSRTEkqc45P9\\_JXa0Y\\_0tfnWBuh82x0a7as16knoZ55UerwdpYaAnR1EALw\\_wcB](https://www.lucidchart.com/pages/landing?utm_source=google&utm_medium=cpc&utm_campaign=_chart_ol_allcountries_mixed_search_brand_exact_&km_CPC_CampaignId=12085501855&km_CPC_AdGroupID=117687297058&km_CPC_Keyword=lucidchart&km_CPC_MatchType=e&km_CPC_ExtensionID=&km_CPC_Network=g&km_CPC_AdPosition=&km_CPC_Creative=491660231431&km_CPC_TargetID=kwd-33511936169&km_CPC_Country=9047055&km_CPC_Device=c&km_CPC_placement=&km_CPC_target=&gad_source=1&gclid=Cj0KCQjwltKxBhDMARIsAG8KnqVaaPSRTEkqc45P9_JXa0Y_0tfnWBuh82x0a7as16knoZ55UerwdpYaAnR1EALw_wcB). [Consulta: 03 de Mayo de 2024].
- [4] Microsoft. Arquitectura de la base de datos. <https://learn.microsoft.com/en-us/dynamics365/business-central/dev-itpro/deployment/product-and-architecture-overview>. [Consulta: 03 de Mayo de 2024].
- [5] Microsoft. Business central pricing. <https://dynamics.microsoft.com/es-es/pricing/>. [Consulta: 03 de Mayo de 2024].
- [6] Microsoft. Microsoft dynamics 365 business central. <https://www.microsoft.com/es-es/dynamics-365/products/business-central>. [Consulta: 03 de Mayo de 2024].
- [7] Microsoft. Sharepoint online. <https://www.microsoft.com/es-es/microsoft-365/sharepoint/collaboration>. [Consulta: 03 de Mayo de 2024].
- [8] Microsoft. Sharepoint pricing. <https://www.microsoft.com/es-es/microsoft-365/sharepoint/compare-sharepoint-plans>. [Consulta: 03 de Mayo de 2024].
- [9] Oracle. Erp. <https://www.oracle.com/es/erp/what-is-erp/>. [Consulta: 03 de Mayo de 2024].
- [10] Visual Paradigm. Visual paradigm. <https://www.visual-paradigm.com/>. [Consulta: 03 de Mayo de 2024].
- [11] VSC. Visual studio code. <https://code.visualstudio.com/>. [Consulta: 03 de Mayo de 2024].