



GRADO EN INGENIERÍA INFORMÁTICA

TRABAJO DE FINAL DE GRADO

**Desarrollo de una aplicación en Power Apps
para la gestión de inventario con la
integración de Microsoft Dynamics 365
Business Central.**

Autor:
Pablo ROMERO RECATALÁ

Supervisor:
Sergi VILAR DOMENECH
Tutor académico:
Ricardo CHALMETA ROSALEÑ

Fecha de lectura: 22 de Junio de 2023
Curso académico 2022/2023

Resumen

El presente documento ofrece una descripción detallada de la memoria del trabajo de final de grado realizado por el estudiante Pablo Romero Recatalá durante su período de prácticas en la empresa *Innova Advanced Consulting*. El enfoque principal del proyecto es la creación de una aplicación que facilite la organización de los productos de la empresa.

La creación de esta aplicación ha implicado un minucioso análisis de los requisitos de los clientes. Con el fin de satisfacer dichos requisitos, se ha desarrollado una aplicación utilizando herramientas de Microsoft, específicamente Power Apps en conjunto con Business Central.

La funcionalidad de la aplicación se divide en dos grupos principales. El primero se centra en la capacidad de modificar la cantidad de stock físico, es decir, la cantidad real de productos existentes en una ubicación específica del almacén. Por otro lado, la segunda funcionalidad permite a la persona encargada del almacén llevar a cabo la reclasificación de productos.

Para implementar estas funcionalidades, se ha realizado modificaciones en el código utilizando Visual Studio, el cual posteriormente se ha ejecutado en Power Apps mediante Power Automate. De esta manera, se ha logrado integrar estas capacidades en la aplicación, brindando a los usuarios las herramientas necesarias para gestionar el stock físico y realizar reclasificaciones de manera eficiente.

Palabras clave

ERP, Business Central, reclasificación, inventario físico, producto, almacén, Power Apps.

Keywords

ERP, Business Central, reclassification, physical inventory, product, location, Power Apps.

Índice general

1	Introducción	7
1.1	Contexto y motivación del proyecto	7
1.2	Objetivo del proyecto	8
1.3	Objetivo y alcance del producto	8
1.3.1	Alcance funcional	8
1.3.2	Alcance organizativo	9
1.3.3	Alcance informático	9
1.4	Descripción detallada del producto	9
1.5	Estructura de la memoria	10
2	Planificación del proyecto	11
2.1	Metodología	11
2.2	Planificación	12
2.3	Seguimiento del proyecto	12
2.4	Costes	15
2.4.1	Recursos humanos	15
2.4.2	Software	16
2.4.3	Hardware	17
2.4.4	Costes Totales	18
2.5	Riesgos	18
2.5.1	Identificación de riesgos	18
2.5.2	Análisis de riesgos	19
3	Análisis	23
3.1	Definición de requisitos	23
3.2	Análisis de requisitos	24
4	Diseño	29
4.1	Diseño de la base de datos	29
4.2	Diseño de la arquitectura del producto	30
4.3	Diseño de las interfaces	32
5	Implementación	39
5.1	Descripción técnica de la implementación	39
5.2	Verificación y validación	46
6	Conclusiones	51

Índice de figuras

1.1	Módulos ERP. [15]	7
2.1	Diagrama de Gantt.	12
3.1	Diagrama de casos de uso.	25
4.1	Arquitectura de la base de datos.[11]	29
4.2	Diseño base de datos relacional.	30
4.3	Arquitectura tres capas.[14]	31
4.4	Arquitectura del producto.[10]	31
4.5	Página Inicial.	32
4.6	Menú de la aplicación.	33
4.7	Página del inventario físico preparado.	33
4.8	Funcionalidad de modificar el Stock Físico.	34
4.9	Mensaje de confirmación.	34
4.10	Página de reclasificación.	35
4.11	Formulario de reclasificación.	36
4.12	Notificación de error de cantidad.	36
4.13	Interfaz de lectura de código QR.	37
4.14	Página con datos rellenados mediante la lectura del código QR.	38
4.15	Página de reclasificación rellenada.	38
5.1	Página API principal.	40
5.2	Llamada a las funciones de la Codeunit desde la API <i>Item Journal Line</i> .	41
5.3	Páginas API complementarias para la reclasificación.	42
5.4	Llamada a las funciones de la Codeunit desde la API <i>Item Journal Line</i> .	43
5.5	Llamada a las funciones de la Codeunit desde la API <i>Item Journal Line</i>	44
5.6	Acción de Power Automate	45
5.7	Código QR con información de producto y almacén.	45
5.8	Código que envía todo el formulario de reclasificación a la colección.	46
5.9	Página de Diario de Inventario Físico de Business Central.	46
5.10	Página de Diario de Reclasificación del Producto de Business Central.	47
5.11	Página de Disponibilidad del producto por Almacén de Business Central.	47
5.12	Página de los test de Business Central.	49
5.13	Función local que calcula la cantidad	49
5.14	Página de Business Central para comprobar el resultado de los test	50

Índice de cuadros

2.1	Costes de recursos humanos.	16
2.2	Costes de software.	17
2.3	Costes de hardware	18
2.4	Costes totales del proyecto.	18
2.5	Tipo de riesgos.	19
3.1	Definición caso de uso CU01	25
3.2	Definición caso de uso CU02	26
3.3	Definición caso de uso CU03	26
3.4	Definición caso de uso CU04	27

Capítulo 1

Introducción

1.1. Contexto y motivación del proyecto

La empresa en la que se ha desarrollado la estancia de prácticas es Innova Advanced Consulting, empresa situada en Castellón concretamente en el CEEI, especializada en el sector del ERP. El ERP es un software de gestión y planificación de recursos empresariales, el cual está organizado en diferentes módulos como pueden ser el módulo de Ventas, Compras y Manejo de inventario entre otros, módulos que se puede observar en la figura 1.1 [15].

Innova cuenta con diversos premios de Microsoft Partner Gold, certificación que se da a la empresa debido a los números conseguidos, estos números incluyen cantidad de clientes, soluciones software a medida creados y finalmente capacidades y competencias de los trabajadores entre otros. Innova es uno de los 11 centros “ISV Development Center” reconocido por Microsoft a nivel mundial (único de habla hispana), también ofrecen un servicio de preparación y formación en el sector.



Figura 1.1: Módulos ERP. [15]

La principal motivación del proyecto es desarrollar una nueva aplicación para la empresa *Kurago* [8], con sede en el País Vasco y especializada en soluciones de software para el sector de la chapa. La empresa ha solicitado a *Innova* la creación de una aplicación que permita organizar todos los productos resguardados en sus distintos depósitos.

La aplicación debe contar con una funcionalidad sencilla y rápida, de modo que el personal encargado no pierda tiempo tratando de entender el software. El objetivo de la empresa cliente es proporcionar al inventarista un listado de productos dentro de su sector, con la posibilidad de realizar modificaciones en la cantidad real de los mismos. Además, si el responsable mueve productos entre diferentes almacenes, debe poder registrar esa información en la aplicación para tener un registro completo dentro de la empresa. A lo largo de la memoria, se brindan explicaciones más detalladas para comprender completamente la funcionalidad requerida.

1.2. Objetivo del proyecto

El objetivo del proyecto parte con la finalidad de solucionar un problema. Este problema era el de la organización y control de cantidades de los diferentes productos que tiene en posesión la empresa. El proyecto pretendía desarrollar una aplicación que facilitará esta gestión comentada anteriormente mediante un producto, en este caso una aplicación para el móvil o tablet, que se le proporcionará al inventarista.

1.3. Objetivo y alcance del producto

La aplicación tiene un uso intuitivo para el inventarista ya que esta consiste en un menú con diferentes opciones para resolver los problemas citados anteriormente. Primeramente, se requiere hacer un control de cantidades, para eso la aplicación recibe datos directamente calculados, permitiendo al inventarista de un almacén en concreto modificar la cantidad física, si esta es incorrecta.

Por otra parte, si dicha persona responsable del almacén pretende hacer movilización de producto, es decir seleccionar un producto y situarlo en otro almacén, esto se le facilita mediante una funcionalidad ligada directamente con las tecnologías utilizadas. Todo esto facilita la actualización de datos de los productos por parte del inventarista en la empresa. Toda esta explicación es ampliada en el punto 1.4

1.3.1. Alcance funcional

Desde el punto de vista funcional el proyecto tiene diferentes objetivos los cuales son:

- Facilitar el stock calculado de cada producto al inventarista
- Facilitar al inventarista la inserción del stock físico de cada producto
- Reinsertar los datos modificados en Microsoft Dynamics 365 BC
- Facilitar al inventarista la opción de reclasificación del producto.
- Facilitar los diferentes almacenes y ubicaciones existentes para hacer la reclasificación.
- Proporcionar un listado con los productos para que así se puedan reclasificar.

1.3.2. Alcance organizativo

Desde el punto de vista organizativo toda empresa que es gestionada desde Business Central debería poder acceder para así poder beneficiarse de la gestión del stock, así como la posterior reclasificación.

1.3.3. Alcance informático

Desde el punto de vista informático la aplicación debe poder conectarse con Microsoft Dynamics 365 BC, para poder recibir el stock calculado y posteriores actualizaciones tanto del stock físico como de la reclasificación.

1.4. Descripción detallada del producto

Ampliando el objetivo mencionado anteriormente cabe indicar las tecnologías que se utilizaron han sido tecnologías y herramientas de Microsoft. Como principal tecnología se ha utilizado el ERP de Business Central, herramienta que ha permitido toda esta funcionalidad. Por otra parte, la base de la aplicación se ha creado en Power Apps, herramienta de Microsoft para realizar las interfaces, así como los botones que generan dicha funcionalidad.

Ahora entrando en detalle en la funcionalidad el producto se ha creado con la finalidad de agilizar el movimiento de actualización del inventario, así como posterior reclasificación. El proyecto pretende aprovecharse de la funcionalidad de “cálculo del inventario físico” que proporciona Dynamics 365 Business Central para obtener la cantidad de unidades que se deberían observar en el inventario (Stock calculado), esta cantidad se le pasa al inventarista (persona encargada de monitorizar la entrada y salida de stock del almacén) para que rellene el campo de la cantidad real (stock físico) del producto. Teniendo ya todos los productos recontados, el inventarista pulsa un botón de confirmación que actualiza estos datos en Business Central y registra en la sección de “diario de inventario físico” del ERP de Microsoft.

La otra funcionalidad mencionada que se contempla en el proyecto es el hecho de reclasificar los productos, es decir, poder seleccionar un producto de un almacén y una ubicación y poder reenviarlo a otra ubicación de otro almacén u otra ubicación del mismo almacén. Esta funcionalidad se ha realizado añadiendo otra pestaña en la aplicación que permita al inventarista seleccionar el producto, posteriormente el almacén donde se encuentra dicho producto y el almacén donde se pretende mover. A todo esto, se unen unas ubicaciones, que son diferentes clasificaciones que puede tener, o no, un almacén. Finalmente se selecciona la cantidad a reclasificar y se pulsa el botón de confirmación. Cabe especificar que se pueden tener diferentes cantidades de un mismo producto en diferentes almacenes, por eso cabe especificar la cantidad a mover de una localización a otra.

1.5. Estructura de la memoria

Esta sección se utiliza para realizar una anotación de la organización que se ha tenido en la realización del documento.

- En el **capítulo 1** como se ha podido observar, se encuentra la empresa donde se realizan las prácticas, así como una explicación de los objetivos y alcance del proyecto.
- En el **capítulo 2** se encuentra la planificación del proyecto donde se puede observar las diferentes metodologías seleccionadas, así como los diferentes costes desglosados y los diferentes riesgos que se han encontrado.
- En el **capítulo 3** se encuentran los diferentes requisitos, desde su definición hasta su posterior análisis.
- En el **capítulo 4** se encuentra el diseño del sistema. Este diseño agrupa diferentes partes, como puede ser el diseño del software y el diseño de la interfaz.
- Finalmente el **capítulo 5** se encuentra la estructura del código y una posterior verificación y validación del producto final.

Capítulo 2

Planificación del proyecto

2.1. Metodología

La metodología utilizada en este proyecto es la metodología en cascada, también conocida como secuencial o ciclo de vida. Esta metodología consiste en ordenar las tareas de tal manera que el inicio de una tarea se lleva a cabo una vez finalizada su predecesora. Se optó por utilizar esta metodología debido a que cada tarea requería conocimientos o funcionalidades previas para poder ser realizada.

Las tareas o fases que componen el proyecto son las siguientes:

- **Formación:** Al inicio del proyecto, para familiarizarse mejor con el mundo del ERP se realizó un estudio sobre Dynamics 365 BC concretamente su versión actual, así como las diferencias con sus versiones anteriores, seguidamente junto con la ayuda de trabajadores experimentados, se recibieron diferentes clases para entender la extensa funcionalidad que almacena este ERP.
- **Planificación:** Terminada ya la formación se realizó un estudio del proyecto para así poder definir más concretamente los objetivos el alcance del mismo y paralelamente empezar con la redacción de la propuesta técnica, una parte fundamental para entender de manera gráfica es la realización del diagrama de Gantt que observamos en la figura 2.1.
- **Análisis:** En Esta fase se quiere entender todo lo que se va a realizar y sobre todo hacer una gestión de los diferentes requisitos a cumplir por el producto.
- **Diseño y desarrollo:** Terminadas las fases anteriores se empezó con el diseño y desarrollo del producto, en esta fase empezamos a utilizar Power Apps, Visual Studio y Business Central para realizar la aplicación.
- **Pruebas:** Finalmente, terminado ya el desarrollo de la app queremos hacer unas test-app para comprobar si el funcionamiento de la aplicación es correcto.

Para que toda tarea se proclamase como finalizada, se necesitaba de la aprobación del

supervisor del proyecto de prácticas. Indicando, si no fuese el caso de la aprobación, los motivos por los cuales dicha tarea no se consideraba apta.

2.2. Planificación

En esta sección se va a comentar como se ha planificado el proyecto. Para representarlo de manera gráfica hay dos opciones, hacer un diagrama de Gantt o realizar una tabla con las tareas y su coste temporal, es decir, su duración. Debido a que Gantt siempre es un modelo más intuitivo para observar todo coste temporal, así como sus fechas de realización.

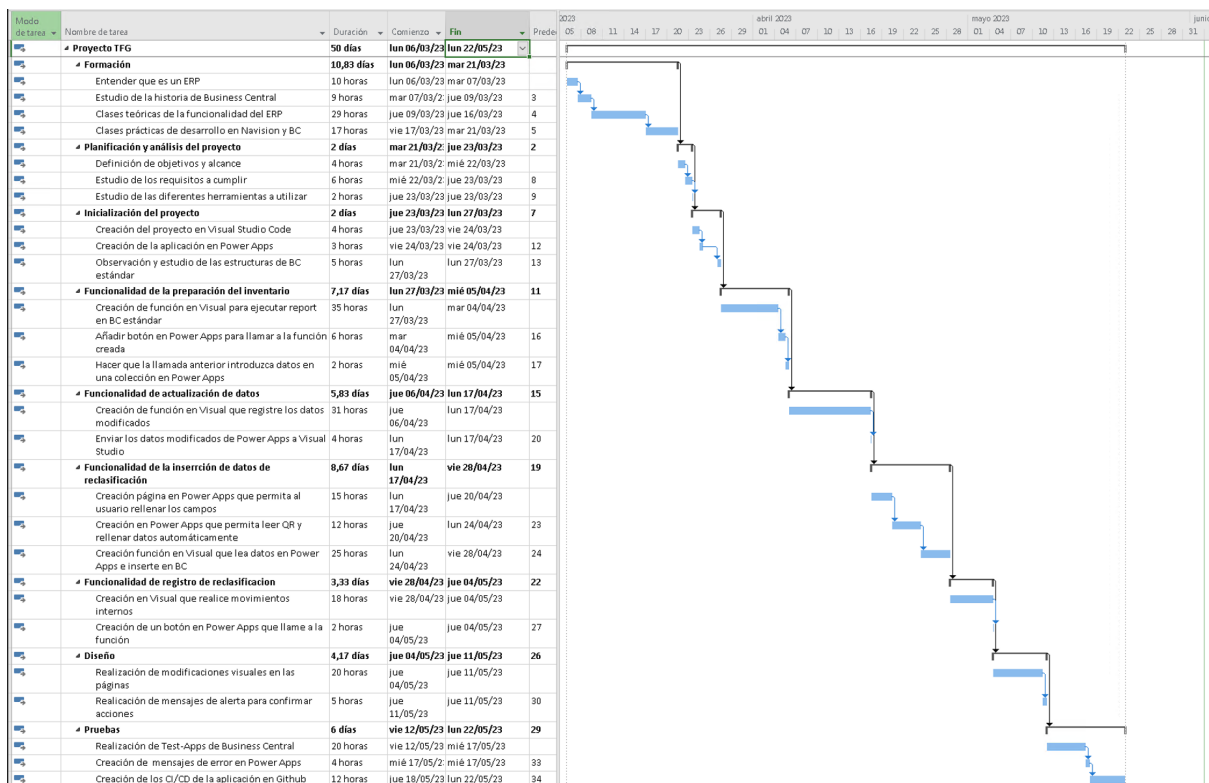


Figura 2.1: Diagrama de Gantt.

2.3. Seguimiento del proyecto

Una de las partes cruciales en un proyecto es llevar a cabo un seguimiento adecuado para asegurarse de que todas las tareas se estén realizando de manera correcta y dentro de los plazos establecidos. Con este propósito, se han elaborado informes quincenales concisos para proporcionar información al tutor de prácticas. Además, el supervisor del proyecto ha estado supervisando de cerca los avances del proyecto, realizando visitas cada dos días para verificar el progreso, abordar cualquier duda o error que pudiera surgir y ofrecer asistencia en su resolución.

Asimismo, se han llevado a cabo diversas presentaciones para que el supervisor pueda evaluar el funcionamiento de la aplicación y verificar si se ha logrado la interfaz esperada.

En cuanto al proceso de seguimiento, los informes quincenales se han elaborado de manera detallada y los supervisores han brindado orientación constante durante sus visitas regulares. Esto ha permitido una supervisión exhaustiva del proyecto, asegurando que se cumplan los objetivos establecidos y se mantenga un progreso constante hacia la finalización del proyecto.

Primera quincena

Con la primera quincena se completó el periodo de formación donde se llevó a cabo un estudio para entender de manera concreta qué era un “Enterprise Resource Planning” y se documentó claramente sobre el ERP que se iba a utilizar, en nuestro caso, Business Central, una herramienta de Microsoft. Para demostrar que se había entendido todo de manera correcta, se presentaron los conocimientos adquiridos ante los integrantes de la empresa, y la persona evaluada se ayudó de unas diapositivas que había realizado él mismo. Una vez finalizado el trabajo, la empresa proporcionó clases adicionales para profundizar en todas las extensas funcionalidades que ofrece la herramienta.

En estas clases, se explicaron diferentes módulos, entre los que se puede destacar el módulo de **Almacén**, en el cual se podía comprobar todo el inventario y realizar diferentes movimientos de reclasificación, crear diferentes almacenes con sus características propias y añadir o crear ubicaciones extra para cada almacén. También se puede destacar el módulo de **CompraVentas**, donde se observaron las diferencias entre albarán, factura y pedido que administra Business Central, así como los diferentes tipos de facturación que la herramienta puede realizar, ya sea facturación mensual, al contado, giro bancario, entre otros. También se pudo modificar los diferentes porcentajes que se querían pagar en cada mes.

Finalmente, se explicó el módulo más denso, el módulo de **Contabilidad**, donde se detallaron todos los diferentes tipos de IVA, así como la explicación de lo que significa que algo vaya al “debe” o al “haber”, términos importantes dentro de la contabilidad de una empresa. Se explicó todo el plan de cuentas, donde se pueden observar de manera clasificada los diferentes datos contables que la empresa debe tener en cuenta, con códigos que comienzan por 1* para la financiación básica, códigos 3* para los movimientos del inventario, códigos 4* para los diferentes tipos de IVA, entre otros. Para asimilar todos estos datos, se realizaron ejercicios después de la clase recibida.

Una vez finalizada la parte teórica, se llevaron a cabo ejercicios prácticos en el departamento de desarrollo de la empresa. Se realizó una instalación de Docker, que básicamente es una plataforma de software que permite probar aplicaciones rápidamente [1]. En Docker, se llevaron a cabo diferentes pruebas y pruebas con la versión de Navision, es decir, lo que antes era Business Central antes de ser adquirida por Microsoft en 2002 y posteriormente renombrada en 2018. Concretamente, se instaló la versión de Navision de 2015. Durante esta fase, se realizaron diferentes desarrollos y se crearon objetos, como páginas, tablas, entre otros.

Segunda quincena

En la segunda quincena, se continuó con la parte práctica de la formación, donde se realizaron los mismos ejercicios en la versión actual del ERP, Navision, para aprender a crear diferentes páginas y “codeunits”, entre otros elementos, y familiarizarse con el lenguaje de programación .AL utilizado por Microsoft.

Una vez finalizada la familiarización con la parte práctica del ERP en las diferentes versiones, se procedió a la planificación y análisis del proyecto. Se definieron claramente los objetivos del proyecto y se entendieron los requisitos exactos solicitados por el cliente. Estos aspectos, estrechamente asociados, permitieron iniciar la inicialización del proyecto, donde se creó y preparó todo en Visual Studio, junto con la creación de la aplicación de Power Apps. En esta etapa, se comenzó creando las páginas API donde se agregarían los datos necesarios para la aplicación de Power Apps.

Para inicializar el proyecto, se llevaron a cabo diferentes acciones. En primer lugar, se descargó Visual Studio y luego se instalaron varias extensiones como “AL CodeAction, AL Language Tools, AL Navigator, AZ AL Dev Tools y GitLens”. Después, se presionó el botón F1 y se escribió “AL:GO”, lo que generó un nuevo proyecto. A continuación, se modificó el archivo launch.json para configurar y sincronizar el servidor de Business Central con Visual Studio. Luego, se volvió a presionar F1 para realizar la función de “Download Symbols” y así configurar el estándar y sincronizarlo con las extensiones instaladas en el servidor.

Con los datos ya integrados, se comenzó a trabajar en los informes de Business Central que cumplirían con la primera funcionalidad de preparación del inventario físico. Esta parte se explicará más detalladamente en el apartado 5.1.

Con los datos ya integrados, se empieza a realizar los reports de Business Central que realicen la primera funcionalidad de preparación del inventario físico, todo esta parte se explicara más detenidamente en el apartado 5.1.

Tercera quincena

En esta quincena, se finalizó el primer informe y se empezó a conectar la funcionalidad del informe con Power Apps, utilizando Power Automate. Se utilizaron botones en la aplicación para llamar al informe y ejecutar su funcionalidad en el ERP de Business Central. También se creó un informe encargado de registrar todos los datos modificados de la aplicación, así como las llamadas realizadas. Al concluir la semana, se comenzó a trabajar en el informe de la primera parte de la segunda funcionalidad mencionada, que consiste en la reclasificación de datos, es decir, la inserción de datos dentro de Business Central.

Quarta quincena

Con la cuarta quincena, se terminó la funcionalidad de inserción de datos y se agregó una nueva funcionalidad para quitarle trabajo al inventarista. Se proporcionó una opción para que

pudiera leer códigos QR y rellenar automáticamente los campos. Todas estas funcionalidades, junto con la de registrar los datos de la reclasificación, se implementaron de manera que solo se permitiera reclasificar datos si eran correctos.

Quinta quincena

Finalmente, en esta última quincena se realizó y pulió el diseño de la interfaz de la aplicación y se hicieron unas test apps con Visual Studio que permitieron comprobar que el funcionamiento de la app era el correcto. Todo esto, junto con las pruebas que se pudieron realizar al hacer uso de la aplicación. Además, se llevó a cabo un CI/CD en github (integración continua y despliegue continuo, respectivamente).

2.4. Costes

En esta sección se pretende hacer una estimación de los costes y los recursos que se han necesitado para completar la estancia en prácticas.

2.4.1. Recursos humanos

Entrando en detalle en el coste de las personas integrantes del proyecto, se puede empezar indicando las personas y los roles que se les ha asignado a cada participante. Primeramente, tenemos al alumno de prácticas que ejerce el rol de programador, concretamente programador junior, posteriormente se encuentra el supervisor del estudiante dentro de la empresa el cual ejercería el rol de analista.

El total de horas ejercidas por el programador junior en el proyecto es un total de 300 horas a un precio aproximado de 6€ la hora. Por otra parte, el analista ha realizado un total de 100 horas, debido a que ha estado primeramente entendiendo los requisitos y traduciéndolos para que el programador junior los entendiese de manera más correcta y seguidamente ha estado revisando que el proyecto se desarrollara de forma correcta. También se tienen que contar las horas de los dos empleados de la empresa que realizaron las clases teóricas de las funcionalidades, por una parte, de compra-venta, inventario y reclasificación y por otra parte de facturación y términos económicos utilizados en el ERP.

Para entender de manera gráfica los costes de recursos humanos se ha realizado el cuadro 2.1. La información de la columna del sueldo se ha obtenido de Glassdoor [7]. Se ha calculado todo este coste multiplicado con las horas, para saber el coste total invertido en la mano de obra.

Costes de recursos humanos			
Rol	Sueldo/hora (€)	Horas	Coste total (€)
Programador junior	6	300	1.800
Analista	15,83	100	1.583
Empleados	14	33	462
Total			3845

Cuadro 2.1: Costes de recursos humanos.

Conociendo los costes de recursos humanos, debemos tener en cuenta también los dos tipos de sobrecostes que hay ligados. Estos dos serían los costes de contratación y los costes indirectos.

Costes de contratación

Estos costes incluyen los impuestos de la seguridad social los cuales representan un porcentaje entre el 20-30% de los costes totales de RRHH. Los gastos de contrataciones incluyen diferentes gastos, entre ellos la publicidad, los honorarios de la agencia y la posterior formación.

Costes indirectos

Finalmente cabe mencionar que estos costes son los costes que no están relacionados directamente con una tarea específica del proyecto. Se pueden nombrar diferentes ejemplos de estos costes como puede ser el alquiler de instalaciones, mantenimiento de equipo o incluso los seguros.

2.4.2. Software

Respecto al software utilizado, cabe resaltar que la mayoría de las herramientas que se han utilizado son de uso gratuito, pero hay de otras que necesitan de la adquisición de una licencia o suscripción, en este caso de Microsoft, para poder ser utilizadas como es el caso de Business Central.

Entrando en detalle en las herramientas gratuitas que se han utilizado. Para la comunicación con los diferentes trabajadores de la empresa y proyecto, se ha utilizado el Teams, herramienta de Microsoft para hacer videollamadas, llamadas sin imagen o simplemente chat de texto. Se utiliza Visual Studio Code para el desarrollo del código en lenguaje .AL, que es el que se utiliza en Business Central. Para la comunicación con la sección de recursos humanos tenemos el Outlook.

Explicando de manera más concreta la licencia, se puede concretar que la empresa de Innova Advanced Consulting tiene una licencia de BC la cual asciende a un precio de 93€ mensuales. Esta suscripción es necesaria para la creación y personalización de diferentes estructuras de

datos dentro de Business Central (No se puede modificar el estándar del software), es decir, para extender su funcionalidad. También necesitamos la licencia para la utilización Power Apps y de Power Automate. Power Automate es una herramienta que se utiliza en Power Apps para crear un flujo. Este flujo es una unión de Power Apps con Visual Studio Code, indirectamente Business Central. Gracias a Power Automate se pueden llamar a las funciones que hemos creado en Visual Studio desde Power Apps. A todo esto, se une que para publicar la aplicación se necesita tener publicada dicha aplicación en Azure.

Dentro de Business Central se ha utilizado una empresa y un entorno de prueba al principio, los cuales tenían el nombre de CRONUS y Demo respectivamente. Posteriormente se creó una empresa y un entorno de prueba desde cero, personalizado para el estudiante que realizaba las prácticas. Para agrupar estos datos se ha realizado la tabla 2.2 con datos obtenidos de diferentes páginas, como es el caso de las licencias de BC [5] por otra parte la Licencia de 365 E3 [12] incluyendo la última herramienta de Microsoft que se ha obtenido información de [13] y finalmente la última licencia de pago es la suscripción mensual de github que se ha consultado en la página. [6].

Costes de Software		
Producto	Coste/mensual (€)	Coste total (€)
Licencia BC	93,60	280,8
Outlook	0	0
Licencia Microsoft 365 E3	25,10	75,3
Microsoft Teams	0	0
Licencia Microsoft Power Apps	4,70	14,1
Microsoft Power Automate Free	0	0
Visual Studio Code	0	0
Extensiones Visual Studio Code	0	0
Github	17,94	53,82
Total		424,02

Cuadro 2.2: Costes de software.

2.4.3. Hardware

Respeto al coste del hardware se va a nombrar todo el material proporcionado, primeramente el dispositivo más utilizado es el ordenador, que en este caso es un portátil de HP, concretamente el **HP 255 G7 Notebook PC** con diferentes características de las cuales podemos destacar su procesador Ryzen 5 y sus 8 GB de memoria RAM. Se proporcionó un combo de ratón y teclado inalámbricos muy cómodos, de la marca **seenda**, en referente al modelo es el **IWG-ZXK26TZ**.

En nuestro espacio de trabajo se encuentra un monitor de 24 pulgadas de la marca **Acer V226HQL** junto con un soporte de Amazon para poder colocar el portátil y el monitor a altura y tener la mesa libre. Todos estos costes se observan de manera detallada en el cuadro 2.3. Para contemplar de manera justa, dicho coste de hardware debería dividirse entre los meses de la estancia en prácticas ya que los productos que se han comprado van a seguir utilizándose.

Costes de hardware	
Producto	Coste (€)
Monitor	169,41
Combo Teclado/Ratón	49,99
Soporte portátil-monitor	46,99
Portátil	439,67
Total	706,06

Cuadro 2.3: Costes de hardware

2.4.4. Costes Totales

Realizando un sumatorio de los diferentes costes totales de cada subsección se puede obtener lo que sería el coste total del producto, como se puede observar en el cuadro 2.4.

Costes totales	
Diferentes costes	Coste total (€)
Coste RRHH	3845
Coste Hardware	706,06
Costes de contratación	961,25
Coste indirectos	961,25
Coste Software	424,02
Total	6.897,58

Cuadro 2.4: Costes totales del proyecto.

2.5. Riesgos

En esta sección vamos a detectar los riesgos y dificultades que nos hemos encontrado a lo largo de la realización del proyecto.

2.5.1. Identificación de riesgos

Esta sección es en la que nombramos los diferentes riesgos que nos podemos encontrar.

- R01: Falta de permisos suficientes para realizar ciertas tareas.
- R02: Falta de experiencia en el lenguaje .AL
- R03: Falta de experiencia en la migración de datos desde Dynamics 365 y PowerApps.
- R04: Falta de tiempo.
- R05: No aceptación de la aplicación por parte de los clientes.

- R06: Falta de entendimiento de los requisitos.

2.5.2. Análisis de riesgos

En este apartado se quiere analizar con detalle los riesgos, indicando primeramente lo grave o leve que puede ser, es decir su magnitud. También se quiere describir más detalladamente en que consiste ese riesgo, y el hecho de que, si sucede, a que parte del proyecto y producto afectaría. Finalmente hemos clasificado también el tipo de riesgo (producto/proyecto). Toda esa información se ha condensado en el cuadro 2.5.

Id	Análisis del riesgo
R01	<p>Magnitud: bastante alta.</p> <p>Descripción: para poder realizar funciones específicas se necesita tener unos permisos especiales que nos tiene que dar la cuenta con rol de admin.</p> <p>Impacto: podría originar limitaciones para realizar la funcionalidad.</p> <p>Indicadores: limitación a la hora de realizar funciones.</p> <p>Tipo: Riesgo de producto.</p>
R02	<p>Magnitud: variable dependiendo de la fase donde aparezca.</p> <p>Descripción: se desconoce el lenguaje y hay poca información (manuales).</p> <p>Impacto: puede suponer que el proyecto tarde más tiempo en finalizar.</p> <p>Indicadores: —</p> <p>Tipo: riesgo del producto/proyecto.</p>
R03	<p>Magnitud: media</p> <p>Descripción: el programador no había realizado nunca migración de datos con BC.</p> <p>Impacto: podrían perderse datos al migrar estos datos.</p> <p>Indicadores: —</p> <p>Tipo: riesgo del producto/proyecto.</p>
R04	<p>Magnitud: media</p> <p>Descripción: El proyecto tiene unos tiempos establecidos y estos se tienen que cumplir.</p> <p>Impacto: podría haber una pérdida de dinero.</p> <p>Indicadores: —</p> <p>Tipo: riesgo del proyecto.</p>
R05	<p>Magnitud: alta.</p> <p>Descripción: al cliente no le gusta el resultado final.</p> <p>Impacto: se tendrían que cambiar cosas para que le guste al cliente.</p> <p>Indicadores: reunión final con el cliente.</p> <p>Tipo: riesgo del producto/proyecto.</p>
R06	<p>Magnitud: Variable dependiendo de la fase en la que aparezca.</p> <p>Descripción: los requisitos son la base del proyecto para desarrollar la aplicación.</p> <p>Impacto: Se puede estar desarrollando una aplicación errónea.</p> <p>Indicadores: —</p> <p>Tipo: riesgo del producto.</p>

Cuadro 2.5: Tipo de riesgos.

Para dar por terminada la parte de los riesgos, se ha de entender cómo se pueden solucionar los riesgos, mejor dicho, como hacer planes de prevención y contingencia para que así estos riesgos comentados anteriormente no causen tantos problemas. Siempre se hará más inciso en los problemas que sean más perjudiciales para el resultado final del proyecto.

R01

- **Plan de prevención:** Realizar un estudio exhaustivo de todas las funciones que vamos a tener que utilizar tanto de Business Central como de Power Platform (agrupación de Power Apps y Power Automate).
- **Plan de contingencia:** Pedir al administrador que proporcione los permisos necesarios para realizar la perfecta ejecución de toda la funcionalidad.

R02

- **Plan de prevención:** Realizar una formación intensa en la sintaxis del lenguaje de forma autónoma siempre reforzando lo aprendido mediante los conocimientos de algún integrante de la empresa.
- **Plan de contingencia:** Realizar una pausa en el desarrollo del proyecto para acabar de adquirir los conocimientos necesarios. Esto sería un riesgo que puede proporcionar una pérdida de tiempo.

R03

- **Plan de prevención:** Realizar unas explicaciones de cómo se realiza y los riesgos y dificultades que se puede encontrar a la hora de realizarlo.
- **Plan de contingencia:** Realizar desde cero otra vez la migración para que todos los datos se agrupen de forma correcta.

R04

- **Plan de prevención:** Realizar desde el primer día un plan de gestión de tiempos adaptado a las necesidades del cliente.
- **Plan de contingencia:** Recortar en funcionalidad y solo realizar la funcionalidad básica para llegar a tiempo a las entregas programadas.

R05

- **Plan de prevención:** Realizar “mockups” para que el cliente pueda observar cómo está quedando la aplicación y como está organizado la gestión de botones y la interfaz en general.
- **Plan de contingencia:** Rehacer la interfaz y la gestión del producto, esto se puede hacer realizando una reunión con el cliente y comentando los errores y diferencias que se han observado intentando así arreglarlo.

R06

- **Plan de prevención:** Realizar una lectura más completa de la descripción y reunirse con el analista para concretar los requisitos del cliente.
- **Plan de contingencia:** En las primeras fases del proyecto podemos realizar todas las modificaciones necesarias para incorporar los cambios. Por otra parte en las últimas fases del mismo, habrá que analizar la importancia de las actualizaciones y tomar la decisión más adecuada.

Riesgos experimentados

Terminada ya la aplicación cabe resaltar la experiencia obtenida con los riesgos. El riesgo que más dificultad nos supuso fue el de los permisos debido que fue al principio del proyecto y el administrador de licencias, la persona con el rol más importante en la empresa, la cual otorga permisos, no se encontraba en las oficinas, y al principio resultó limitante. Cuando volvió tuvo una fácil solución este riesgo.

Otro riesgo que supuso dificultades fue el del poco conocimiento del programador sobre el lenguaje de programación utilizado en el proyecto y la poca información que hay en las redes sobre dicho lenguaje, esta solución fue un poco más progresiva, ya que el conocimiento mejoraba con el paso de los días y la prueba y error del programador con respecto a la escritura del código.

Capítulo 3

Análisis

3.1. Definición de requisitos

Los requisitos que tiene que cumplir la aplicación se pueden especificar mediante historias de usuario. Esta manera de especificar es muy correcta para esta aplicación debido a que la gran parte del proyecto es *frontend*. Para entender aún más los requisitos, se ha dividido en dos apartados que hacen referencia a la funcionalidad.

Funcionalidad de modificación del inventario físico

HU01 - Preparación del inventario

- Como cliente de la aplicación.
- Quiero poder consultar todos los productos calculados así como su stock.
- Para no tener que acordarme de memoria de todos los productos.

HU02 - Modificación del inventario

- Como cliente de la aplicación.
- Quiero poder modificar la cantidad de stock físico (cantidad real) de los productos calculados.
- Para poder editar la cantidad del producto si esta no es correcta.

HU03 - Registrar los cambios realizados

- Como cliente de la aplicación.

- Quiero poder registrar todos los datos modificados.
- Para completar la tarea de modificación del inventario físico.

Funcionalidad de reclasificación del producto

HU04 - Insertar datos

- Como cliente de la aplicación.
- Quiero poder realizar una reclasificación.
- Para hacer modificación de la localización del producto.

HU05 - Visualizar los datos

- Como cliente de la aplicación.
- Quiero poder modificar la cantidad de stock físico (cantidad real) de los productos calculados.
- Para no tener que acordarme de memoria de todos los almacenes ni de las ubicaciones relacionadas.

HU06 - Registrar los cambios realizados

- Como cliente de la aplicación.
- Quiero poder registrar todos los datos modificados.
- Para completar la tarea de reclasificación del producto.

3.2. Análisis de requisitos

Entendiendo la funcionalidad, se ha realizado un diagrama de casos de uso 3.1 pero una realización básica debido a que se observa una funcionalidad sencilla debido a que simplemente es cálculo y modificación del stock por una parte y reclasificación de un producto entre diferentes almacenes por otra parte.

En el diagrama se pueden apreciar tres actores, uno de los cuales es externo. Por otra parte, está el usuario que en este caso sería el inventarista que es el actor donde recae todo el peso de la aplicación. El único caso de uso que realizaría el admin es el de registrar datos si fuese necesario.[4].

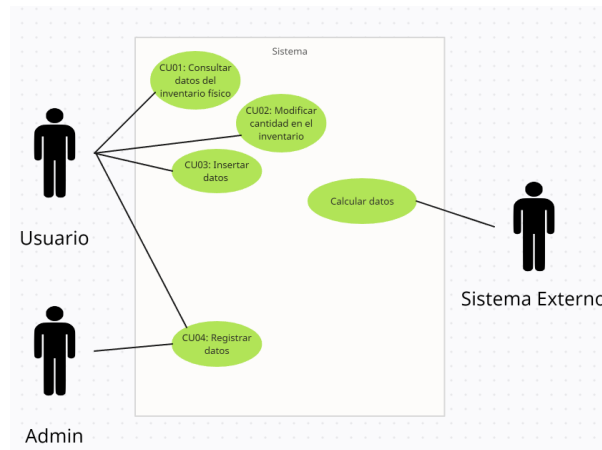


Figura 3.1: Diagrama de casos de uso.

A continuación, basándonos en la figura anterior, se presentarán diferentes tablas desde 3.1 hasta 3.4 que describen la funcionalidad de los diferentes casos de uso y la secuencia de interacciones correspondientes.

Identificador	CU01
Nombre:	Consultar datos del inventario físico
Versión:	1.0
Fuente:	Pablo Romero Recatalá
Actor principal:	Usuario
Descripción:	Este caso de uso representa la funcionalidad de consultar los datos de Business Central.
Secuencia normal:	<ol style="list-style-type: none"> 1. El usuario entra en la aplicación 2. El usuario pulsa el botón de inventario 3. El usuario puede observar los datos de los productos 4. El usuario pulsa el botón de la reclasificación 5. El usuario puede seleccionar los diferentes datos de cada almacén
Excepciones:	<ol style="list-style-type: none"> 1. No existan estos datos

Cuadro 3.1: Definición caso de uso CU01

Identificador	CU02
Nombre:	Modificar cantidad en el inventario
Versión:	1.0
Fuente:	Pablo Romero Recatalá
Actor principal:	Usuario
Descripción:	Este caso de uso representa la funcionalidad de poder modificar los datos, previamente preparados.
Secuencia normal:	<ol style="list-style-type: none"> 1. El usuario entra en la aplicación 2. El usuario pulsa el botón de inventario 3. El usuario puede observar los datos de los productos 4. El usuario selecciona la cantidad a modificar
Excepciones:	<ol style="list-style-type: none"> 1. No existan estos datos

Cuadro 3.2: Definición caso de uso CU02

Identificador	CU03
Nombre:	Insertar datos
Versión:	1.0
Fuente:	Pablo Romero Recatalá
Actor principal:	Usuario
Descripción:	Este caso de uso representa la funcionalidad de insertar datos dentro de la reclasificación.
Secuencia normal:	<ol style="list-style-type: none"> 1. El usuario entra en la aplicación 2. El usuario pulsa el botón de reclasificación 3. El usuario puede observar los diferentes desplegables 4. El usuario pulsa selecciona los datos 5. El usuario pulsa el botón de insertar
Excepciones:	—

Cuadro 3.3: Definición caso de uso CU03

Identificador	CU04
Nombre:	Registrar datos
Versión:	1.0
Fuente:	Pablo Romero Recatalá
Actor principal:	Usuario y administrador
Descripción:	Este caso de uso representa la funcionalidad de registrar todos los datos modificados en los diferentes CU.
Secuencia normal: <i>Usuario</i>	<ol style="list-style-type: none"> 1. El usuario entra en la aplicación 2. El usuario pulsa el botón de reclasificación o inventario 3. El usuario modifica los valores que desee 4. El usuario pulsa el botón de registrar 5. Todos los cambios realizados se actualizan en Business Central
Secuencia normal: <i>Administrador</i>	<ol style="list-style-type: none"> 1. El administrador inicia sesión en Business Central 2. El administrador entra en la página donde quiere realizar modificaciones 3. El administrador hace las inserciones y modificaciones necesarias y actualiza
Excepciones:	<ol style="list-style-type: none"> 1. No se modifiquen datos

Cuadro 3.4: Definición caso de uso CU04

Capítulo 4

Diseño

4.1. Diseño de la base de datos

En esta sección se explicará de manera detallada donde se almacenan todos los datos de la aplicación. Business Central en su formato *SaaS* (software basado en la nube). Contando que es una herramienta de Microsoft conectada con Azure, se aprovecha el **Azure SQL Database**. Se puede observar la arquitectura y el flujo de datos mediante la figura 4.1 [11].

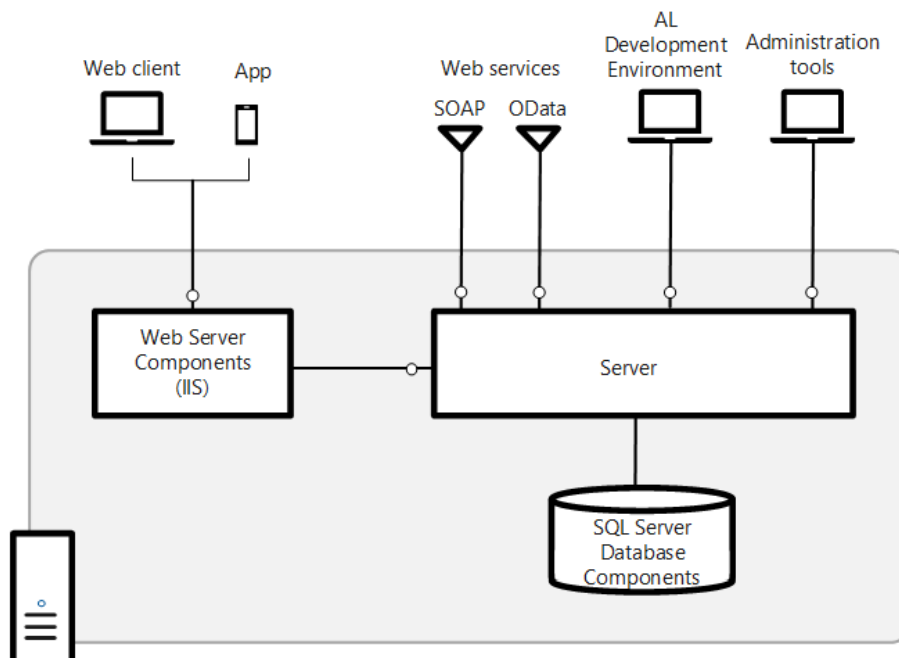


Figura 4.1: Arquitectura de la base de datos.[11]

Por otro lado, los datos en Power Apps se han almacenado mediante unas estructuras de datos que se llaman colecciones y galerías. Mediante un fragmento de código que se explicara en la sección de 5.1 se puede rellenar una colección con los datos de las funciones de la aplicación.

Esta colección almacena la información de manera temporal, posteriormente es llamada por la galería para así poder mostrar dicha información por la página de la aplicación. La galería sería como una tabla expandible es decir puedes ir insertando datos de manera sencilla. Finalmente en Business central la base de datos se ha distribuido relacionando la tabla de *Item Journal Line* que almacena los datos necesarios para realizar las dos funcionalidades, junto con la tabla de **Item** para conseguir toda información de los productos y la tabla de **Location**. Para realizar los cálculos de la cantidad de producto que hay en un almacén se ha utilizado la tabla *Item Ledger Entry*. Toda colección y galería de Power Apps, puede obtener datos de estas tablas mediante las APIs mostradas en la sección 5.1.

En la figura 4.2 se ha realizado mediante el Lucidchart [9] una figura con los datos que se ha utilizado de las tablas, debido a que las tablas tienen muchísima más información y datos que no se han considerado necesarios.

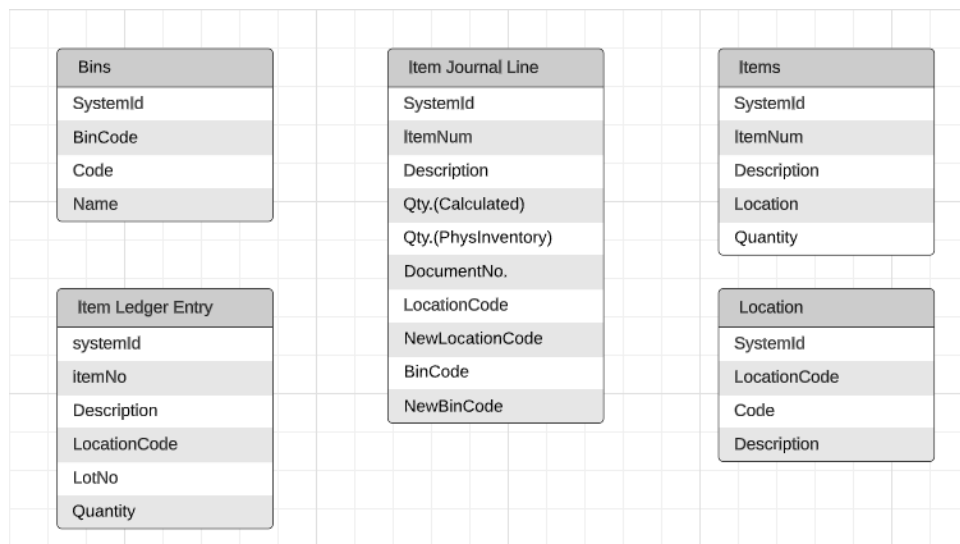


Figura 4.2: Diseño base de datos relacional.

4.2. Diseño de la arquitectura del producto

Como se observa en la figura 4.3 se puede definir la arquitectura del producto como una arquitectura de tres capas, que como bien se sabe indica que tiene una capa de presentación, en este caso sería la interfaz de Power Apps donde el inventarista observa los datos, es decir, es la comunicación de los datos con el usuario. Seguidamente tenemos la capa de negocio, aquí es donde se realizan todos los cálculos, donde se encuentra todo el backend. Finalmente tenemos la capa de acceso a los datos que une los cálculos a realizar en la capa anterior junto con la base de datos donde se almacenan estos.

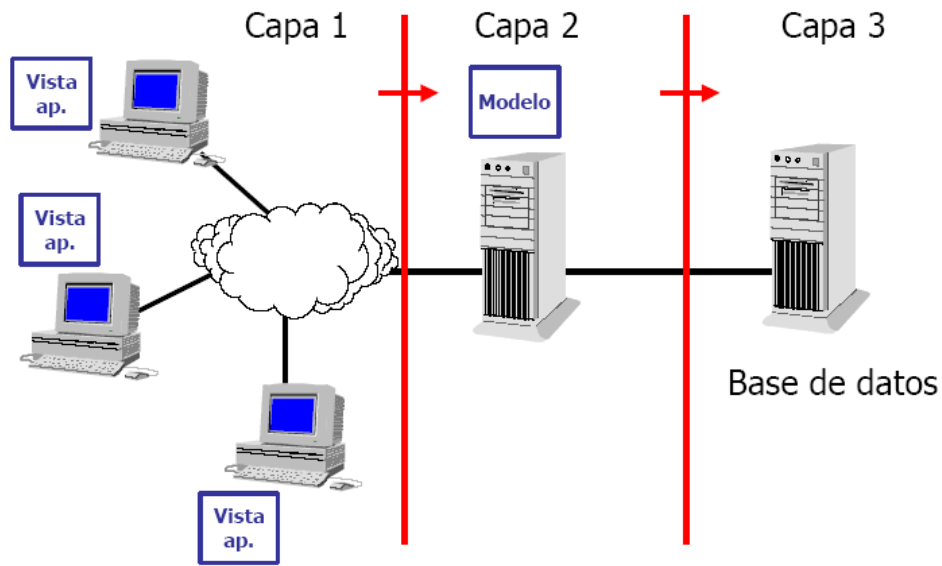


Figura 4.3: Arquitectura tres capas.[14]

Como se puede apreciar en la figura 4.4[10], se establece un flujo ordenado de tareas numeradas. En primer lugar, el desarrollador genera el proyecto y a continuación procede a desarrollar la aplicación utilizando Power Platform. Una vez completada esta etapa, se crea un repositorio en Azure para almacenar los cambios que se realicen posteriormente. El siguiente paso consiste en que el desarrollador cree otro proyecto específicamente para realizar las pruebas de Power Platform. Por último, el desarrollador publica la aplicación de Power Apps y utiliza la herramienta Azure Application Insights para monitorear y evaluar el rendimiento de dicha aplicación.

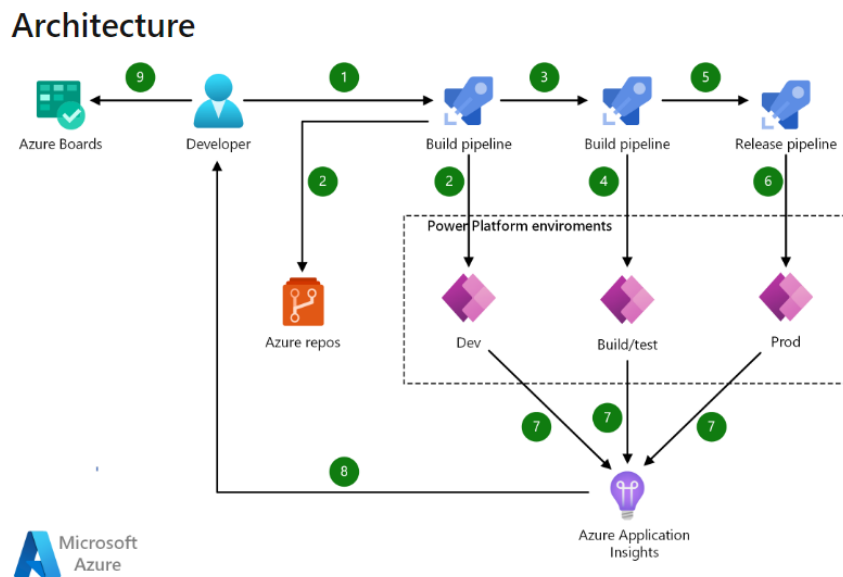


Figura 4.4: Arquitectura del producto.[10]

4.3. Diseño de las interfaces

A continuación, se va a observar cómo sería el apartado gráfico de la aplicación. Se ha pensado que la mejor manera de entender la aplicación sería con un flujo/demostración de como utilizaría el inventarista la aplicación.

En la figura 4.5 el inventarista se encontraría con la página inicial, donde se observa el nombre y logo de la empresa de **Innova** junto con un botón en forma de casa. Este botón haría las primeras llamadas a Business Central y prepararía el inventario, ya que es el cálculo más costoso.



Figura 4.5: Página Inicial.

En la figura 4.6 se encuentra el menú donde se observan dos botones con forma de carpeta que indican la funcionalidad que se puede realizar en la aplicación, por una parte se tiene el botón de inventario físico donde están preparados todos los datos de los productos y por otra parte la reclasificación. También existe el botón de recargar para volver a preparar el inventario.



Figura 4.6: Menú de la aplicación.

En la figura 4.7 se puede observar los diferentes datos separados en cuatro columnas, el código del producto, su nombre, el stock que ha calculado Business Central y el stock que el inventarista deberá modificar si es necesario.

Item	Nombre	Stock Calc.	Stock Físico
1896-S	Escritorio ATENAS	4	4
1906-S	Pedestal móvil ATENAS	5	5
1908-S	Silla giratoria LONDRES	3	3
1920-S	Mesa de conferencia AV	5	5
1920-S	Mesa de conferencia AV	1	1
1920-S	Mesa de conferencia AV	4	4
1928-S	Lámpara ÁMSTERDAM	8	8
1936-S	Silla invit. BERLÍN, amar.	100	100
1953-W	Sección invitados 1	-49	-49
1960-S	Silla inv. ROMA, verde	2	2
1964-S	Silla inv. TOKIO, azul	4	4
1965-W	Paquete de conferencia	-81	-81
1968-S	Silla giratoria MÉXICO, n	3	3
1968-S	Silla giratoria MÉXICO, n	6	6
1968-S	Silla giratoria MÉXICO, n	1	1
1969-W	Conjunto de conferenci	-7	-7

Figura 4.7: Página del inventario físico preparado.

En la figura 4.8 podemos observar cómo se modificaría el Stock Físico, este campo solo aceptaría números, para que no hubiese un colapso en el posterior registro de los datos.

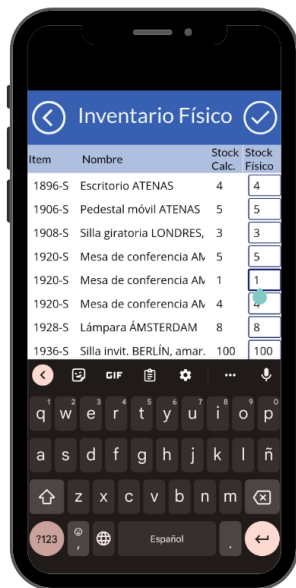


Figura 4.8: Funcionalidad de modificar el Stock Físico.

Por otra parte, cuando se termina de modificar todos los campos, le damos al botón de confirmar, para registrar los datos. Para confirmar dicha acción saldría un mensaje con unos botones para ejecutar la acción. En la figura 4.9 se puede observar el mensaje que saldría (parte izquierda) y como quedaría el inventario al registrarlo todo(derecha), es decir, que borraría todo registro.



Figura 4.9: Mensaje de confirmación.

En la figura 4.10 se puede observar que enseñaría la aplicación justo después de pulsar el botón con forma de carpeta y nombre reclasificación que aparece en el menú. En esta página

cabe destacar el botón con forma de signo de suma, el cual redirigiría al formulario donde se permitiría añadir los datos.

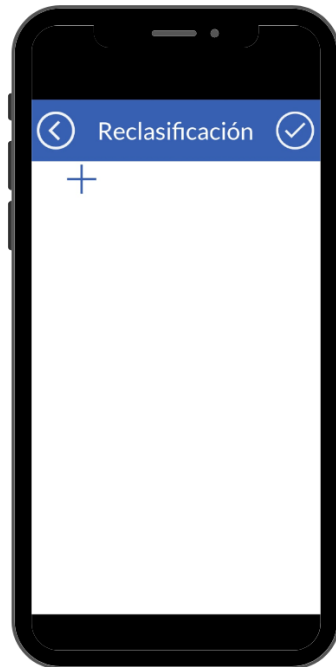


Figura 4.10: Página de reclasificación.

Para poder rellenar los datos de manera correcta, se ha insertado un desplegable con todos los productos que existen en el entorno de Business Central donde se trabajaría como podemos observar en la figura 4.11. Añadiendo características, cabe destacar que el desplegable de la ubicación, está relacionada directamente con el almacén al que pertenece, si en el desplegable de almacén seleccionas un almacén, en el desplegable de ubicaciones saldrían las ubicaciones que existen en ese almacén.



Figura 4.11: Formulario de reclasificación.

Cuando se selecciona el producto y el almacén de origen, se calcula la cantidad total del producto seleccionado que hay en ese almacén en concreto. Como se puede observar en la figura 4.12 se puede observar el campo de la cantidad máxima, por otra parte se encuentra el campo editable de la cantidad, es decir, que cantidad de ese producto queremos reclasificar. Si se indican unos valores incorrectos, se notificaría mediante un mensaje para que insertemos una cantidad correcta es decir una cantidad entre uno y la cantidad máxima calculada.



Figura 4.12: Notificación de error de cantidad.

Cuando se selecciona el botón de la cámara que se puede observar en las figuras anteriores, la aplicación enciende directamente la cámara del móvil o tablet que el inventarista estaría utilizando para así poder realizar una lectura automática de un código QR como se observa en la figura 4.13.



Figura 4.13: Interfaz de lectura de código QR.

En la figura 4.14 se muestra la página de reclasificación, justo después de leer el código QR de la figura anterior. Se han rellenado los campos del producto y del almacén de origen. Esta funcionalidad está pensada para que el inventarista simplemente con leer el código que estaría en ese lugar del almacén, pueda hacer la reclasificación de una manera más rápida.



Figura 4.14: Página con datos rellenos mediante la lectura del código QR.

Finalmente en la figura 4.15 se puede observar la pantalla de reclasificación pero esta vez rellena con datos. Justo cuando se le da al botón de confirmar del formulario de reclasificación, se van añadiendo los datos dentro de esta página, para que cuando se tenga claro toda la reclasificación, se pulse el botón de confirmar de arriba a la derecha y se haga.



Figura 4.15: Página de reclasificación rellena.

Capítulo 5

Implementación

5.1. Descripción técnica de la implementación

En esta sección se va a explicar mediante imágenes, todo el código utilizado en Visual Studio el cual realiza toda la funcionalidad. Primeramente observando la figura 5.1 se encuentra una página tipo API que recibe los datos de la tabla *Item Journal Line* del estándar de Business Central. Esta API se ha realizado para poder obtener los datos e información desde Power Apps, como se puede observar se han añadido diferentes *fields* que son los campos, como puede ser el código del ítem, su descripción la cantidad, entre otros. Para realizar las capturas y que el lenguaje se quedase más visible se ha utilizado la página web de [2].

Seguidamente en la figura 5.2 se encuentran las funciones a ejecutar, todas estas funciones están dentro de la misma página API de la imagen anterior. Explicando las funciones desde el principio, primero se tiene un trigger que se ejecuta siempre que se abre la página API, es decir, si desde Power Apps se llama a la API, ejecutarán la primera función. Seguidamente están las diferentes funciones que se explicaran en figuras posteriores, ya que en esta imagen solo se puede observar la inicialización de la variable tipo *Codeunit* y la posterior llamada a la función. Cabe recalcar que para poder llamar las funciones desde Power Automate se tiene que poner las dos características de [**ServiceEnable**] y [**Scope('Cloud')**] es decir, justo antes de la creación de la función se tiene que escribir esas dos características.

```

1 page 50105 "Item Journal Line API"
2 {
3   SourceTable = "Item Journal Line";
4   PageType = API;
5   Caption = "Item Journal Line API";
6   APIPublisher = 'innova';
7   APIGroup = 'custom';
8   APIVersion = 'v1.0';
9   EntityName = 'apiItemJournalLineWS';
10  EntitySetName = 'apiItemJournalLinesWS';
11  DelayedInsert = true;
12  SourceTableView = sorting("Line No.") where("Journal
Template Name" = const('INVENTARIO'), "Journal Batch Name" =
const('GENERICO'));
13
14  layout
15  {
16    area(Content)
17    {
18      repeater(ListadoCampos) //Siempre se utiliza un
repeater para listar en las páginas tipo API
19      {
20        field(systemId; Rec.SystemId)
21        {
22          ApplicationArea = All;
23        }
24
25        field(itemNum; Rec."Item No.")
26        {
27          ApplicationArea = All;
28        }
29
30        field(productName; Rec.Description)
31        {
32          ApplicationArea = All;
33        }
34
35        field(qtyCalculated; Rec."Qty. (Calculated)")
36        {
37          ApplicationArea = All;
38        }
39
40        field(qtyPhysic; Rec."Qty. (Phys.
Inventory)")
41        {
42          ApplicationArea = All;
43        }
44
45        field(quantity; Rec.Quantity)
46        {
47          ApplicationArea = All;
48        }
49
50      }
51    }
52  }
53
54 }
55

```

```

//-----Reclasificación-----
//
field(docNo; Rec."Document No.")
{
  ApplicationArea = All;
}
field(warehouse; Rec."Location Code")
{
  ApplicationArea = All;
}
field(destinationWarehouse; Rec."New Location
Code")
{
  ApplicationArea = All;
}
field(ubication; Rec."Bin Code")
{
  ApplicationArea = All;
}
field(newUbication; Rec."New Bin Code")
{
  ApplicationArea = All;
}
}
}

```

Figura 5.1: Página API principal.


```

89 //-----Inventario Fisico-----//
90
91
92 trigger OnOpenPage()
93 var
94     calcPhyInv: Codeunit CalcPhysInventory;
95 begin
96     calcPhyInv.calcPhysInventory(); //Función para
97     preparar el inventario fisico
98 end;
99
100 [ServiceEnabled]
101 [Scope('Cloud')]
102 procedure RegisterItemJournalLine()
103 var
104     regItem: Codeunit CalcPhysInventory; //Codeunit para
105     registrar el inventario fisico
106 begin
107     regItem.regisPhysInventory();
108 end;
109 //-----Reclasificación-----//
110 [ServiceEnabled]
111 [Scope('Cloud')]
112 procedure InsertLineRec(item: Code[20]; qty: Decimal;
113     loc: Code[10]; newLoc: Code[10]; line: Integer);
114 var
115     prepItem: Codeunit ItmJnlReclas; //Función para
116     insertar líneas en la reclasificación
117 begin
118     prepItem.prepReclasification(item, qty, loc, newLoc,
119     line);
120 end;
121 [ServiceEnabled]
122 [Scope('Cloud')]
123 procedure RegisterReclasification()
124 var
125     recItem: Codeunit ItmJnlReclas;
126 begin
127     recItem.regisReclasification(); //Función para
128     registrar las líneas insertadas
129 end;
130
131
132
133

```

Figura 5.2: Llamada a las funciones de la Codeunit desde la API *Item Journal Line*.

En la figura 5.3 se encuentran las diferentes tres páginas tipo API que completan la obtención de datos de las tres diferentes tablas de todos los almacenes, ubicaciones y las entradas del diario de contabilidad. Todas estas tablas permiten hacer la reclasificación de una manera más sencilla.

<pre> 1 page 50112 "API Bins WS IN BHS" 2 { 3 ODataKeyFields = SystemId; 4 SourceTable = Bin; 5 PageType = API; 6 Caption = 'API Bins WS'; 7 APIPublisher = 'innova'; 8 APIGroup = 'custom'; 9 APIVersion = 'v1.0'; 10 EntityName = 'apiBinWS'; 11 EntitySetName = 'apiBinsWS'; 12 DelayedInsert = true; 13 14 layout 15 { 16 area(Content) 17 { 18 group(GroupName) 19 { 20 field(systemId; Rec.SystemId) 21 { 22 ApplicationArea = All; 23 } 24 field(LocationCode; Rec."Location Code") 25 { 26 ApplicationArea = Location; 27 } 28 field(code; Rec.Code) 29 { 30 ApplicationArea = All; 31 } 32 field(description; Rec.Description) 33 { 34 ApplicationArea = All; 35 } 36 } 37 } 38 } 39 } </pre>	<pre> 1 page 50111 "Locations WS API" 2 { 3 ODataKeyFields = SystemId; 4 SourceTable = Location; 5 PageType = API; 6 Caption = 'API Locations WS'; 7 APIPublisher = 'innova'; 8 APIGroup = 'custom'; 9 APIVersion = 'v1.0'; 10 EntityName = 'apiLocationsWS'; 11 EntitySetName = 'apiLocationsWS'; 12 DelayedInsert = true; 13 14 layout 15 { 16 area(Content) 17 { 18 repeater(GroupName) 19 { 20 field(systemId; Rec.SystemId) 21 { 22 ApplicationArea = All; 23 } 24 field(code; Rec.Code) 25 { 26 ApplicationArea = All; 27 } 28 field(name; Rec.Name) 29 { 30 ApplicationArea = All; 31 } 32 } 33 } 34 } 35 } </pre>
--	---

```

1 page 50113 "Ledger Entry API"
2 {
3     ODataKeyFields = SystemId;
4     SourceTable = "Item Ledger Entry";
5     PageType = API;
6     Caption = 'API Ledger Entry WS';
7     APIPublisher = 'innova';
8     APIGroup = 'custom';
9     APIVersion = 'v1.0';
10    EntityName = 'apiLedgerEntryWS';
11    EntitySetName = 'apiLedgerEntryWS';
12    DelayedInsert = true;
13
14    layout
15    {
16        area(Content)
17        {
18            repeater(GroupName)
19            {
20                field(itemNo; Rec."Item No.")
21                {
22                    ApplicationArea = All;
23                }
24
25                field(description; Rec."Description")
26                {
27                    ApplicationArea = All;
28                }
29
30                field(location; Rec."Location Code")
31                {
32                    ApplicationArea = All;
33                }
34
35                field(lotNo; Rec."Lot No.")
36                {
37                    ApplicationArea = All;
38                }
39
40                field(quantity; Rec."Quantity")
41                {
42                    ApplicationArea = All;
43                }
44            }
45        }
46    }
47 }

```

Figura 5.3: Páginas API complementarias para la reclasificación.

En las siguientes dos figuras se explica la funcionalidad que ejecutan las diferentes funciones, o como se llaman en el lenguaje *.AL procedures*. Primeramente en la figura 5.4 se observa dentro del apartado **var** todas las variables que se utilizarán en la función. Ahora la funcionalidad primeramente lo que se tiene que hacer es filtrar todos los datos de la tabla por el *Template*(plantilla) y el *Batch*(grupo) con el **.SetRange** teniendo ya todo eso filtrado, primeramente eliminamos los datos que podría haber anteriormente, para que no se acumulen los diferentes bloques de cálculo del inventario físico.

Teniendo ya filtrada la tabla, se llama al Report del estándar que calcula y prepara el inventario físico. Se hace una llamada a la página *UseRequestPage* con su propiedad a false, para que no salga la opción de confirmar, que sea un “*click to function*” es decir, que cuando pulsemos un botón realice la función.

La segunda función realiza el registrado de datos y tiene una fácil explicación, ya que solo filtra como lo comentado anteriormente, para obtener la tabla filtrada con los datos, si esta tabla tiene datos, inicializa la *Codeunit Item Jnl. -Post Batch* para poder registrar los datos dentro de Business Central.

```

1 codeunit 50149 "CalcPhysInventory"
2 {
3     procedure calcPhysInventory() //Función que prepara el
      diario de inventario físico
4         var
5             itemJnlLine: Record "Item Journal Line";
6             itemJnlBatch: Record "Item Journal Batch";
7
8             calcQty: Report "Calculate Inventory";
9             NoSeriesMgt: Codeunit NoSeriesManagement;
10            docNo: Code[20];
11            itemsNotOnInvnt: Boolean;
12            includeItemWithNoTransaction: Boolean;
13
14        begin
15            itemJnlLine.SetRange("Journal Template Name",
16            'INVENTARIO');
17            itemJnlLine.SetRange("Journal Batch Name",
18            'GENERIC0');
19            itemJnlLine.DeleteAll();
20            itemJnlLine.Init();
21            itemJnlLine.Validate("Journal Template Name",
22            'INVENTARIO');
23            itemJnlLine.Validate("Journal Batch Name",
24            'GENERIC0');
25            calcQty.SetItemJnlLine(itemJnlLine);
26            itemJnlBatch.Get('INVENTARIO', 'GENERIC0');
27            docNo := NoSeriesMgt.GetNextNo(itemJnlBatch."No.
28            Series", Today, false);
29            calcQty.InitializeRequest(Today, docNo,
30            itemsNotOnInvnt, includeItemWithNoTransaction);
31            calcQty.UseRequestPage(false);
32            calcQty.Run();
33        end;
34
35    procedure regisPhysInventory() //Función que registra el
      diario de inventario físico
36        var
37            itemJnlLine: Record "Item Journal Line";
38
39        begin
40            itemJnlLine.SetRange("Journal Template Name",
41            'INVENTARIO');
42            itemJnlLine.SetRange("Journal Batch Name",
43            'GENERIC0');
44            if itemJnlLine.FindSet() then
45                CODEUNIT.Run(CODEUNIT::"Item Jnl.-Post Batch",
46                itemJnlLine);
47            end;
48        }
49    }

```

Figura 5.4: Llamada a las funciones de la Codeunit desde la API *Item Journal Line*.

Por otra parte en la figura 5.5 se observa la *Codeunit* de reclasificación, esta tiene dos funciones, la segunda no haría falta explicarla ya que tiene un funcionamiento similar a la del registrado de la Codeunit anterior pero con un filtrado diferente. Haciendo hincapié en la primera función, se observa que se le pasan diferentes parámetros a la función, estos parámetros los recibe desde el formulario de reclasificación de Power apps. En esta función se inicializa el registro de la tabla con la función **.Init** y mediante la función **.Validate** vamos insertando los parámetros dentro de la tabla filtrada. Teniendo ya insertados todos los parámetros, se almacenan los datos en la tabla, para la posterior llamada a la función de registrar.

```

1 codeunit 58138 "ItemJnlReclas"
2 {
3     procedure prepReclasificacion(item: Code[20]; qty:
4         Decimal; Loc: Code[10]; newLoc: Code[10]; Line: Integer)
5     var
6         ItemJnlLine: Record "Item Journal Line";
7     begin
8
9         //Bases bases
10        ItemJnlLine.Init();
11        ItemJnlLine.Validate("Journal Template Name",
12            'TRANSFEREN');
13        ItemJnlLine.Validate("Journal Batch Name",
14            'GENERICO');
15        ItemJnlLine.Validate("Posting Date", Today);
16        ItemJnlLine.Validate("Document No.", 'T01001');
17        //Bases inventarios
18        ItemJnlLine.Validate("Item No.", item);
19        ItemJnlLine.Validate("Entry Type", ItemJnlLine."Entry
20            Type"::"Transfer");
21        ItemJnlLine.Validate("Quantity", qty);
22        if loc = 'VACIO' then //Para que no de error en el
23            PowerAutomate.Run que pesarte en valor = 0430
24            ItemJnlLine.Validate("Location Code", '')
25        else
26            ItemJnlLine.Validate("Location Code", loc);
27        if newLoc = 'VACIO' then
28            ItemJnlLine.Validate("New Location Code", '')
29        else
30            ItemJnlLine.Validate("New Location Code", newLoc);
31        ItemJnlLine.Validate("Inventory Posting Group",
32            'MERCADERIA');
33        ItemJnlLine.Validate("Source Code", 'RECLAS.JNL');
34        ItemJnlLine.Validate("Line No.", line);
35        ItemJnlLine.Insert();
36    end;
37
38    procedure regisReclasificacion()
39    var
40        ItemJnlLine: Record "Item Journal Line";
41    begin
42        ItemJnlLine.SetRange("Journal Template Name",
43            'TRANSFEREN');
44        ItemJnlLine.SetRange("Journal Batch Name",
45            'GENERICO');
46
47        if ItemJnlLine.FindSet() then
48            CODEUNIT.Run(CODEUNIT::"Item Jnl.-Post Batch",
49                ItemJnlLine);
50        end;
51    }
52 }

```

Figura 5.5: Llamada a las funciones de la Codeunit desde la API *Item Journal Line*

Entrando ahora en la parte del código de Power Apps, cabe resaltar que cada icono tiene diferentes propiedades, como puede ser el *OnSelect* o el *OnChange* dentro de esa propiedad hemos realizado diferentes acciones. Para que se comprenda toda la funcionalidad, así como su dificultad se ha pensado que la mejor manera de explicarlo es mediante el flujo de datos. Es decir, como el inventarista utilizaría la aplicación cosa que se puede observar mediante las diferentes fotografías de la sección de la interfaz comentada anteriormente. Primeramente al darle al botón de la casa se ejecuta el siguiente comando = `CLEARCOLLECT (PHYSI-TEM;'APIITEMJOURNALLINESWS(INNOVA/CUSTOM/V1.0)');`; `SET(COUNTER;10000) ;`; `NAVIGATE(HOME)`.

Analizando este código tenemos el ClearCollect el cual crea una colección borrando los datos, si tenía anteriormente. En este código selecciona los datos, aprovechando el Trigger **OnOpenPage** de la página API de Item Journal Line, comentada anteriormente. Inicializa una variable “counter” que se utilizara posteriormente y con el Navigate para moverse al menú.

Después el inventarista se encontraría en el menú principal, y los dos botones tienen la misma funcionalidad, inicializar una variable a false que es la transición del botón de confirmar que esta visible en la figura 4.9.

Si el inventarista entra en la opción de calcular inventario físico, se encuentra una galería (comentada en el diseño de la base de datos) donde se muestran todos los valores al darle al botón de confirmar, la variable del botón se vuelve true, para dar visibilidad al mensaje de confirmación. Al darle al botón de confirmar se ejecuta la acción de Power Automate que podemos observar en la figura 5.6.

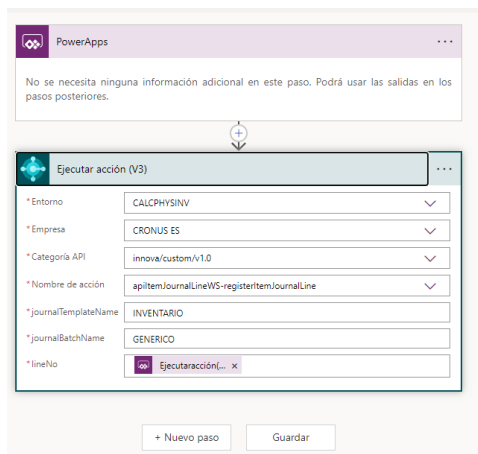


Figura 5.6: Acción de Power Automate

Por otra parte si el inventarista entra en la opción de reclasificación y le da al botón de la cámara, en la propiedad OnScan que tiene el lector de códigos de Power Apps, se ha indicado que realice la acción de `CLEARCOLLECT(TEST; BARCODEREADER.BARCODES); SET(PRUEBA; CONCAT(BARCODEREADER.BARCODES; VALUE; "—")); NAVIGATE(RELLENADOFORMULARIO); CLEARCOLLECT(UNO; FILTER('APILEDGERENTRYWS (INNOVA/CUSTOM/V1.0)'; LOCATION = LAST(TRIMENDS (SPLIT (PRUEBA; "—"))).VALUE ; ITEMNO = FIRST (TRIMENDS (SPLIT(PRUEBA; "—"))).VALUE))`. Todo este código simplemente lo que hace es que lee el QR de la figura 5.7 que se ha realizado con la página [3] y rellena la reclasificación.



Figura 5.7: Código QR con información de producto y almacén.

Para calcular la cantidad máxima que se observa en la página de reclasificación se utiliza la página API de *Entry Ledger Entry* filtrando la relación de almacén con el producto seleccionado y hacer un `SUM(UNO;QUANTITY)`. Se tiene que realizar la suma sobre una colección debido a que haciéndolo directamente con la página API proporcionaba unos errores de solapamiento de datos.

Finalmente, cuando se pulsa el botón de confirmar en el formulario de la reclasificación se ha tenido que coger uno a uno todos los valores del formulario y pasárselo a la colección de la página de la reclasificación. Para observarlo de forma más clara se ha pasado la captura 5.8. La variable correcto es una variable *Booleana* que indica si la cantidad es correcta o no. Esta variable se actualiza en la propiedad **OnChange** del campo del formulario referente a la cantidad.

```

If(correcto;
Collect(
  rec;
  {prod: Prod.SelectedText.Value;
  desc: Desc.SelectedText.Value;
  almacen: AlmacenOrigen.SelectedText.Value;
  almacenNuevo: AlmacenDestino.SelectedText.Value;
  ubi: UbicacionOrigen.SelectedText.Value;
  ubiNueva: UbicacionDestino.SelectedText.Value;
  cant: Cantidad;
  line: LineaPrueba
  }
);;Set(Counter; Counter + 10000);;
Navigate(Reclasificacion);;
Reset(AlmacenOrigen);;Reset(AlmacenDestino);;
Reset(UbicacionOrigen);;Reset(UbicacionDestino);;
Notify("Debes seleccionar una cantidad correcta")

```

Figura 5.8: Código que envía todo el formulario de reclasificación a la colección.

5.2. Verificación y validación

Finalmente para la sección de verificación y validación del producto, se han realizado diferentes Test, unos en Power Apps por parte del programador junior proyectando a su supervisor la funcionalidad de la aplicación, intentando hacer diferentes pruebas como seleccionar un almacén vacío ya que esto dentro de Business Central se puede hacer, ya que un producto no tiene por qué estar ligado directamente a un almacén o por ejemplo seleccionar una cantidad incorrecta de valores o insertar letras en lugar de números en el apartado de modificación del inventario. Todas estas acciones han sido tratadas para que el inventarista no pueda introducir datos incorrectos.

También se ha comprobado mediante doble pantalla, que los datos que se insertan en Power Apps, son posteriormente actualizados dentro de Business Central. Esto se puede observar mediante las figuras posteriores donde se observa en el ERP el antes y el después de ejecutar las funciones dentro de Power Apps.

Primeramente se tiene la figura 5.9 donde se encuentra la página de Business Central después de pulsar el botón con forma de casa de la página de inicio de Power Apps, aquí se observa el cálculo y posterior rellenado de los datos en la tabla.

Fecha registro	Tipo movimiento	Nº documento	Nº producto	Descripción	Cód. almacén	Cód. vendedor/co...	Stock calculado	Cart. Stock físico	Cantidad	Precio unitario	Importe
31/05/2023	Ajuste posi...	DF0271	1896-S	Escritorio ATENAS			4	4	0	784.60	0.00
31/05/2023	Ajuste posi...	DF0271	1906-S	Pedestal móvil ATENAS			5	5	0	339.90	0.00
31/05/2023	Ajuste posi...	DF0271	1908-S	Silla gimnasia LONDRES, azul			3	3	0	148.90	0.00
31/05/2023	Ajuste posi...	DF0271	1920-S	Mesa de conferencia AMBERES	AMARILLO		5	5	0	508.00	0.00
31/05/2023	Ajuste posi...	DF0271	1920-S	Mesa de conferencia AMBERES	ESTE		1	1	0	508.00	0.00
31/05/2023	Ajuste posi...	DF0271	1920-S	Mesa de conferencia AMBERES	OESTE		4	4	0	508.00	0.00
31/05/2023	Ajuste posi...	DF0271	1928-S	Lámpara ÁMSTERDAM			8	8	0	43.10	0.00
31/05/2023	Ajuste posi...	DF0271	1936-S	Silla invit. BERLÍN, amar.			100	100	0	151.10	0.00
31/05/2023	Ajuste posi...	DF0271	1953-W	Sección invitados 1			-49	-49	0	0.00	0.00

Figura 5.9: Página de Diario de Inventario Físico de Business Central.

Seguidamente se observa en la figura 5.10 la página de Business Central que es modificada con

la inserción de datos de la reclasificación. Para confirmar de manera correcta que está haciendo la reclasificación se tiene que ir a la página de Business Central donde está el producto, seleccionar disponibilidad por almacén y observar como se muestra en la figura 5.11 la modificación de dichos datos. Se tiene que observar la última columna **Inventario** que es la que muestra la cantidad del producto distribuida por almacén. Para que no se pierda el hilo argumental, se han modificado los datos seleccionados en la figura 4.15.

Fecha requisito	Nº documento	Nº producto	Descripción	Cód. almacén	Cód. ubicación	Cód. almacén destino	Nueva ubicación	Cantidad	Cód. unidad medida	Liq. por nº orden	Dpto Código	Nue Cód
01/06/2023	T01001	1920-S	Mesa de conferencia AMBERES	AMARILLO	OESTE	OESTE	ESTE	2	UDS	0		
	T01001	1920-S	Mesa de conferencia AMBERES	OESTE		ESTE		1	UDS	0		

Figura 5.10: Página de Diario de Reclasificación del Producto de Business Central.

Código	Nombre	Necesidades brutas	Recepción programada	Recep. planif.	Inventario
→	(Ubicación sin especificar)	0	0	0	0
AMARILLO	Almacén amarillo	0	0	0	5
BLANCO	Almacén blanco	0	0	0	0
ESTE	Almacén del este	0	0	0	1
LOG. PROP.	Logística propia	0	0	0	0
LOG. SUBC.	Logística subcontr.	0	0	0	0
OESTE	Almacén del oeste	0	0	0	4

Código	Nombre	Necesidades brutas	Recepción programada	Recep. planif.	Inventario
→	(Ubicación sin especificar)	0	0	0	0
AMARILLO	Almacén amarillo	0	0	0	3
BLANCO	Almacén blanco	0	0	0	0
ESTE	Almacén del este	0	0	0	2
LOG. PROP.	Logística propia	0	0	0	0
LOG. SUBC.	Logística subcontr.	0	0	0	0
OESTE	Almacén del oeste	0	0	0	5

Figura 5.11: Página de Disponibilidad del producto por Almacén de Business Central.

Entrando en detalle en los Test realizados en código, se puede observar la figura 5.12 con todo lo que se ha tenido que realizar en Visual Studio para poder realizar los test. Primeramente para poder unir los test con el proyecto, se ha tenido que ejecutar otra vez las acciones de **AL:GO y Download Symbols** así generando otro proyecto, al cual creamos una carpeta con nombre Test, para que el código AL lo detecte (Se observa como el formato de la carpeta cambia y se vuelve roja). Seguidamente se inicia otra vez el proyecto donde se almacena toda la funcionalidad, es decir el proyecto anterior, en el cual se tiene que seleccionar la opción de añadir una nueva carpeta al *Workspace* es decir al espacio de trabajo. Seleccionamos la carpeta

de los test nombrada anteriormente y dentro del fichero **app.json** se indica dentro del apartado de dependencias, la relación con el proyecto principal. Se tiene que indicar nombre, id y versión del proyecto.

Con todo esto ya configurado, se tiene que subir una extensión que se llama Library Assert, al *Extension Management* de Business Central, ya instalada la extensión, se procede a ir a la carpeta de Test y se hace otra vez el *Download Symbols* se realiza una nueva modificación a las dependencias dentro del fichero app.json y se empieza a configurar lo nombrado en la figura 5.12.

```
1 codeunit 50112 "MyTestCodeunit"
2 {
3     Subtype = Test;
4
5     [Test]
6     procedure TestPreparacionInventario();
7     var
8         ItemJnlLine: Record "Item Journal Line";
9         AssertThat: Codeunit "Library Assert";
10        prepPhys: Codeunit CalcPhysInventory;
11        NotExpectedValue: Variant;
12        ActualValue: Variant;
13        IfErrorTxt: Text;
14
15    begin
16        // [Scenario]
17
18        // [Given] Setup:
19        ItemJnlLine.SetRange("Journal Template Name", 'INVENTARIO');
20        ItemJnlLine.SetRange("Journal Batch Name", 'GENERIC');
21
22        // [When] Exercise:
23        prepPhys.calcPhysInventory();
24        // [Then] verify:
25        NotExpectedValue := 0;
26        ActualValue := ItemJnlLine.Count;
27        IfErrorTxt := 'Tendria que rellenarse la tabla';
28        AssertThat.AreNotEqual(NotExpectedValue, ActualValue, IfErrorTxt);
29    end;
30
31
32
33    [Test]
34    procedure TestRegistroInventario();
35    var
36        ItemJnlLine: Record "Item Journal Line";
37        AssertThat: Codeunit "Library Assert";
38        prepPhys: Codeunit CalcPhysInventory;
39        ExpectedValue: Variant;
40        ActualValue: Variant;
41        IfErrorTxt: Text;
42
43    begin
44        // [Scenario]
45
46        // [Given] Setup:
47        ItemJnlLine.SetRange("Journal Template Name", 'INVENTARIO');
48        ItemJnlLine.SetRange("Journal Batch Name", 'GENERIC');
49
50        // [When] Exercise:
51        prepPhys.calcPhysInventory(); //Primero insertamos en el inventario datos
52        prepPhys.regisPhysInventory(); //Luego registramos el inventario
53        // [Then] verify:
54        ExpectedValue := 0;
55        ActualValue := ItemJnlLine.Count;
56        IfErrorTxt := 'Tendria que vaciarse la tabla';
57        AssertThat.AreEqual(ExpectedValue, ActualValue, IfErrorTxt);
58    end;
59
```



```

61 [Test]
62 procedure TestInsertarReclasificacion();
63 var
64     ItemJnlLine: Record "Item Journal Line";
65     AssertThat: Codeunit "Library Assert";
66     prepRec: Codeunit ItmJnlReclas;
67     NotExpectedValue: Variant;
68     ActualValue: Variant;
69     IFErrorTxt: Text;
70
71
72 begin
73     // Scenario
74
75     // [Given] Setup:
76     ItemJnlLine.SetRange("Journal Template Name", 'TRANSFEREN');
77     ItemJnlLine.SetRange("Journal Batch Name", 'GENERIC0');
78
79     // [When] Exercise:
80     prepRec.prepReclasificacion('1928-S', 1, 'AZUL', 'VERDE', 10000);
81     // [Then] Verify:
82     NotExpectedValue := 0;
83     ActualValue := ItemJnlLine.Count;
84     IFErrorTxt := 'Tendria que rellenarse la tabla';
85     AssertThat.AreNotEqual(NotExpectedValue, ActualValue, IFErrorTxt);
86 end;
87
88 [Test]
89 procedure TestRegistrarReclasificacion();
90 var
91     AssertThat: Codeunit "Library Assert";
92     reclasificacion: Codeunit ItmJnlReclas;
93     blueBefore: Integer;
94     blueAfter: Variant;
95     greenBefore: Integer;
96     greenAfter: Variant;
97
98     IFErrorTxt: Text;
99
100
101 begin
102     // Scenario
103
104     // [Given] Setup:
105     blueBefore := CalcularValorInventarioFiltrado('1928-S', 'AZUL'); //Tendria que ser 38
106     greenBefore := CalcularValorInventarioFiltrado('1928-S', 'VERDE'); //Tendria que ser
107
108     // [When] Exercise:
109     reclasificacion.regisReclasificacion();
110
111     // [Then] Verify:
112
113     blueAfter := CalcularValorInventarioFiltrado('1928-S', 'AZUL'); //Tendria que ser 37
114     greenAfter := CalcularValorInventarioFiltrado('1928-S', 'VERDE'); //Tendria que ser
115
116     IFErrorTxt := 'No se han movido de forma correcta los valores';
117     AssertThat.AreEqual(blueBefore - 1, blueAfter, IFErrorTxt);
118     AssertThat.AreEqual(greenBefore + 1, greenAfter, IFErrorTxt);
119 end;
120

```

Figura 5.12: Página de los test de Business Central.

Esta función de la figura 5.13 simplemente es una ayuda para comprobar que el cálculo es el correcto, hace un filtrado dentro de Business Central y calcula la cantidad total del producto a reclasificar que existe en ese almacén.

```

122 local procedure CalcularValorInventarioFiltrado(item: Code[20]; loc: Text): Decimal;
123 //Calcula el valor del inventario filtrado por item y localizacion
124 var
125     itemLed: Record "Item Ledger Entry";
126
127 begin
128     itemLed.SetFilter("Item No.", item);
129     itemLed.SetFilter("Location Code", loc);
130     itemLed.CalcSums(Quantity);
131
132     exit(itemLed.Quantity);
133 end;
134
135 }

```

Figura 5.13: Función local que calcula la cantidad

Para concluir con la verificación de la funcionalidad del producto, se pueden observar los test programados anteriormente mediante el apartado de Test AL de Business Central, ya que al estar conectado mediante el launch.json se publica el código dándole al botón **F5**. Para observar cómo querían los test se puede observar la figura 5.14.

AI Test Tool | Fecha de trabajo: 23/01/2025

Suite Name: DEFAULT Code Coverage Tracking: Disabled

Test Runner Codeunit: 130450 - Test Runner - Isol. Codeunit Code Coverage Track All Sessions:

Administrar ▶ Run Tests ▶ Run Selected Tests ▶ Get Test Codeunits ▶ Get Test Codeunits by Range ▶ Update Test Methods ▶ Delete Lines

Line Type	Codeunit ID	Name	Run	Result	Error Message
Codeunit	50112	MyTestCodeunit	<input checked="" type="checkbox"/>	Success	--
Function	50112	TestPreparacionInventario	<input checked="" type="checkbox"/>	Success	--
Function	50112	TestRegistrarInventario	<input checked="" type="checkbox"/>	Success	--
Function	50112	TestInsertarReclasificacion	<input checked="" type="checkbox"/>	Success	--
Function	50112	TestRegistrarReclasificacion	<input checked="" type="checkbox"/>	Success	--

Figura 5.14: Página de Business Central para comprobar el resultado de los test

Respeto a la validación del trabajo realizado, se han realizado las pertinentes comprobaciones para observar que el producto final, cumple con los requisitos impuestos por el cliente. Para realizar dichas comprobaciones se realizó una entrega a espera de *feedback* del cliente. Para que la validación esté completada al 100% se realizó una presentación que agrupaba cada una de las historias de usuario definidas en el proyecto.

Capítulo 6

Conclusiones

Para concluir la memoria se comentarán las opiniones y sensaciones encontradas relacionados con la realización del proyecto juntamente con su memoria. Para la más correcta explicación se dividirán estas opiniones y sensaciones en diferentes ámbitos.

Primeramente hablando del **ámbito formativo** cabe resaltar el aprendizaje de tecnologías que no se habían visto en profundidad en la carrera. Iniciando con el tema del software de un ERP que solo se había realizado una asignatura en este último año y en esta formación se han aprendido conceptos más concretos. Indicar también el aprendizaje de un nuevo lenguaje de programación como es el lenguaje .AL , las diferentes acciones que se pueden realizar en Business Central *SaaS* y el lenguaje y acciones que se pueden realizar en las diferentes aplicaciones de Power Platform. Para concluir con este ámbito cabe resaltar el aprendizaje de migración de datos, conexión entre las diferentes herramientas de Microsoft y la evaluación de los test.

Seguidamente se comenta el **ámbito profesional** ya que era la primera experiencia laboral dentro del sector informático. Las anteriores experiencias habían sido en el sector hostelero. Cabe destacar que ha sido muy interesante saber cómo está organizada la empresa. En la empresa se ha de indicar que hay dos departamentos separados en dos pisos, en esta conclusión se va a comentar la organización del departamento de BC. En este departamento se realizan reuniones cada cierto tiempo, dependiendo del proyecto que están realizando. La manera de trabajar de la empresa consiste en recibir una propuesta por parte de una empresa, esta propuesta la analiza el analista y evalúa la cantidad de horas que puede costar en función de los conocimientos de los trabajadores. Teniendo ya clara esta acción se concreta una fecha periódica para ir observando el adelanto del proyecto.

El último ámbito que se va a comentar es el **ámbito personal** en el cual solo hay palabras de gratitud y comodidad. Desde el primer día se respira un aire familiar y siempre se interesan en las actividades a realizar y si pueden ayudar siempre lo van a hacer. También les gusta realizar actividades conjuntas como ir a almorzar los viernes para reforzar los vínculos interempresariales. Lo que más me ha gustado es que el hecho de realizar la formación en un sector como es el del ERP me permite especializarme en un sector en el que hay mas demanda que oferta, es decir, que se necesitan especialistas en el sector. Esto potencia un futuro profesional prospero.

Finalmente a **nivel de proyecto** se han cumplido todos los objetivos incluso se han ampliado funcionalidades como viene ser el lector de códigos QR y posterior relleno del mismo. También se había propuesto mejoras a futuro como pueden ser la creación de un repositorio en github que utilice la plantilla de AL:GO para que se pueda instalar directamente desde VisualStudio y que realice los TEST correspondientes si posteriormente el estándar se modifica.

Para acabar con la conclusión el resultado del proyecto ha resultado tener una dificultad elevada debido a la poca información que hay sobre el sector, es más, justo una semana anterior a terminar el proyecto, los de Microsoft hicieron un ejemplo de aplicación que se relacionase con Business Central, teniendo ya terminado todo el producto. Hablando de la estancia en prácticas ha sido muy gratificante, tanto es así por las dos partes que se ha proporcionado una oferta de seguir en la empresa para posteriores proyectos.

Bibliografía

- [1] Amazon. Precio de la utilización de docker. <https://aws.amazon.com/es/docker/>. [Consulta: 12 de Abril de 2023].
- [2] Chalk.ist. Crea bonitas imagenes sobre el código de programación. <https://chalk.ist/>. [Consulta: 01 de Junio de 2023].
- [3] CodeGenerator. Código qr para rellenar la reclasificación. <https://es.qr-code-generator.com/>. [Consulta: 30 de Mayo de 2023].
- [4] Creately. Creación de diagrama de casos de uso. <https://app.creately.com/d/gx9BG5MKKSB/edit>. [Consulta: 22 de Mayo de 2023].
- [5] Fortón. Coste licencia business central iws. <https://www.triangle.es/que-licencias-existen-para-dynamics-365-business-central/>. [Consulta: 12 de Abril de 2023].
- [6] Github. Precio de la suscripción team de github. <https://github.com/pricing>. [Consulta: 12 de Abril de 2023].
- [7] Glassdoor. Búsqueda de empleo en glassdoor. <https://www.glassdoor.es/>. [Consulta: 30 de Abril de 2023].
- [8] Kurago. Empresa kurago. <https://kurago.software/es/>. [Consulta: 01 de Junio de 2023].
- [9] Lucidchart. Creación de diagrama de casos de uso. https://www.lucidchart.com/pages/es/landing?utm_source=google&utm_medium=cpc&utm_campaign=_chart_es_tier1_mixed_search_brand_exact_&km_CPC_CampaignId=1487565747&km_CPC_AdGroupID=56488588774&km_CPC_Keyword=lucidchart&km_CPC_MatchType=e&km_CPC_ExtensionID=&km_CPC_Network=g&km_CPC_AdPosition=&km_CPC_Creative=284415987027&km_CPC_TargetID=kwd-33511936169&km_CPC_Country=9047055&km_CPC_Device=c&km_CPC_placement=&km_CPC_target=&gclid=CjwKCAjwyeujBhA5EiwA5WD7_ZQLjhBDN32F8Kb_EcWr983hHIAxDrlZwhWXajDGnVZiUm9_sm2xvhoCs2EQAvD_BwE. [Consulta: 18 de Mayo de 2023].
- [10] Microsoft. Arquitectura azure junto con power apps. <https://learn.microsoft.com/es-es/azure/architecture/solution-ideas/articles/azure-devops-continuous-integration-for-power-platform/>. [Consulta: 12 de Abril de 2023].

- [11] Microsoft. Azure sql database. <https://learn.microsoft.com/en-us/dynamics365/business-central/dev-itpro/deployment/product-and-architecture-overview/>. [Consulta: 12 de Abril de 2023].
- [12] Microsoft. Coste licencia 365 e3. https://www.microsoft.com/es-es/microsoft-365/enterprise/compare-office-365-plans?ef_id=_k_Cj0KCQjwmtGjBhDhARIsAEqfDef_evHnnh8AF_yqgtNz-YnbcRPaJueCXeZTOAbmd12ErhmQcRm_b00aAsfwEALw_wcB_k_&OCID=AIDcmmppzjhjayb_SEM__k_Cj0KCQjwmtGjBhDhARIsAEqfDef_evHnnh8AF_yqgtNz-YnbcRPaJueCXeZTOAbmd12ErhmQcRm_b00aAsfwEALw_wcB_k_&gclid=Cj0KCQjwmtGjBhDhARIsAEqfDef_evHnnh8AF_yqgtNz-YnbcRPaJueCXeZTOAbmd12ErhmQcRm_b00aAsfwEALw_wcB. [Consulta: 12 de Abril de 2023].
- [13] Microsoft. Coste licencia power apps. <https://powerapps.microsoft.com/es-es/pricing/>. [Consulta: 12 de Abril de 2023].
- [14] oness.sourceforge.net. Arquitectura tres capas. <https://oness.sourceforge.net/proyecto/html/ch03s02.html>. [Consulta: 01 de Junio de 2023].
- [15] XelCron . Los diferentes módulos de un erp. <https://www.xelcron.mx/que-son-los-modulos-erp/>. [Consulta: 23 de Febrero de 2023].