

Universidad de Alcalá

Escuela Politécnica Superior

Grado en Ingeniería de Computadores

Trabajo Fin de Grado

SISTEMA PARA LA GESTIÓN INFORMATIZADA DE STOCK
DE PRODUCTOS

Autor: Adrián Gálvez Gómez

Tutor: Noelia Hernández Parra

Cotutor: Javier Lorenzo Díaz

2022



Universidad
de Alcalá

Escuela Politécnica Superior

GRADO EN INGENIERÍA DE COMPUTADORES

Trabajo Fin de Grado

SISTEMA PARA LA GESTIÓN INFORMATIZADA DE STOCK
DE PRODUCTOS

Autor: Adrián Gálvez Gómez

Tutor: Noelia Hernández Parra

Cotutor: Javier Lorenzo Díaz

TRIBUNAL:

Presidente: D. Ignacio Parra Alonso

Vocal 1º: D. Antonio Guerrero Baquero

Vocal 2º: Dña. Noelia Hernández Parra

FECHA: 09/2022

Agradecimientos

Con este trabajo pongo fin a una de las etapas de mi vida donde más me he exigido a mi mismo, donde más he sufrido e incluso llorado, pero también reído y disfrutado.

Gracias a mis compañeros de carrera. Se ha forjado una amistad más allá de las aulas que difícilmente se romperá, donde cada uno ha puesto algo de su parte para que cada etapa de esta carrera fuese más fácil de llevar.

Gracias también a todos los profesores que he tenido a lo largo de estos años. En una carrera como esta, la paciencia a la hora de enseñar es clave, y todos y cada uno de ellos han sabido demostrar lo profesionales y comprometidos que son con esta labor, creando un sentimiento de interés por las asignaturas cada vez mayor. Especialmente a ti, Noelia, por la paciencia que has tenido durante este trabajo, con todas las circunstancias e inconvenientes que nos han ido rodeando durante su desarrollo.

Gracias a mi familia, que me ha apoyado en todo momento desde el primer día, y por haberme brindado la educación y los valores de los que hoy puedo presumir.

Gracias por último a mis amigos de toda la vida, que sin saber exactamente con lo que me podía encontrar durante cada curso, han sabido estar siempre a la altura.

Índice general

Índice general	I
Índice de figuras	V
Resumen	VII
Abstract	IX
Resumen extendido	XI
1 Introducción	1
1.1 Motivación	1
1.2 Objetivos	3
1.3 Organización de la memoria	3
2 Base teórica	5
2.1 Funcionamiento del almacén.	5
2.2 Herramientas utilizadas.	9
3 Desarrollo del sistema	11
3.1 Relación entre las tablas de la base de datos	11
3.2 Tabla Usuarios	11
3.3 Tabla Productos	13
3.4 Tabla Almacén	14
3.5 Tabla Pedidos Pendientes	15
3.6 Tabla Historial de Pedidos	16
3.7 Tabla Control Stock	17

4	Interfaz	19
4.1	Interfaz de la visión del almacén	20
4.2	Interfaz de la gestión principal	21
4.2.1	Pantalla Login	21
4.2.2	Pantalla Nuevo Producto	22
4.2.3	Pantalla Control de Stock	22
4.2.4	Pantalla Dar Entrada	23
4.2.5	Pantalla Ver Reposición	24
4.2.6	Pantalla Sugerir Reposición	25
4.2.7	Pantalla Ubicar	26
4.2.8	Pantalla Generador de Pedidos	27
4.2.9	Pantalla Simular Pedido	29
4.2.10	Pantalla Iniciando Sesión como Usuario	31
5	Casos de Uso	35
5.1	CU-01 - Iniciar Sesión	35
5.2	CU-02 - Limpiar BBDD Pedidos Pendientes/Historial	36
5.3	CU-03 - Sugerir Reposición	37
5.4	CU-04 - Abrir Ventana Ubicar	38
5.5	CU-04-1 - Ubicar	39
5.6	CU-05 - Abrir Ventana Ver Reposición de Fija	40
5.7	CU-05-1 - Ver Reposición de Fija	41
5.8	CU-06 - Abrir Ventana Introducir Nuevo Producto	42
5.9	CU-06-1 - Introducir Nuevo Producto	43
5.10	CU-07 - Abrir Ventana Dar Entrada	44
5.11	CU-07-1 - Dar Entrada	45
5.12	CU-08 - Actualizar Control de Stock	46
5.13	CU-09 - Abrir Ventana Generador de Pedidos	47
5.14	CU-09-1 - Generador de Pedidos	48
5.15	CU-10 - Abrir Ventana Simular Pedido	49
5.16	CU-10-1 - Simular Pedido	50
5.17	CU-11 - Abrir Ventana Ubicar Como Usuario	51
5.18	CU-11-1 - Ubicar Como Usuario	52

6 Conclusiones y líneas futuras	53
6.1 Conclusiones	53
6.2 Líneas Futuras	54
7 Bibliografía	55

Índice de figuras

2.1	Reparto de los pasillos	6
2.2	Reparto ubicaciones de un pasillo	6
2.3	Reparto completo del almacén	6
2.4	Nomenclatura de cada ubicación	7
3.1	Relación entre las tablas del sistema	12
3.2	Tabla usuarios	12
3.3	Tabla usuarios en base de datos	13
3.4	Tabla productos	13
3.5	Tabla productos en base de datos	13
3.6	Tabla almacén	14
3.7	Tabla pedidos pendientes	15
3.8	Tabla historial pedidos	16
3.9	Tabla control stock	17
4.1	Visión almacén	20
4.2	Pantalla login	21
4.3	Pantalla menú	21
4.4	Pantalla introducir producto	22
4.5	Pantalla dar entrada.	23
4.6	Pantalla ver reposición inicial.	24
4.7	Pantalla ver reposición si no existe fija.	24
4.8	Pantalla ver reposición si no hay mercancía.	24
4.9	Pantalla ver reposición con éxito.	25
4.10	Pantalla sugerir reposición inicial.	25
4.11	Pantalla sugerir reposición con éxito.	25
4.12	Pantalla sugerir reposición sin mercancía.	25

4.13	Pantalla ubicar inicial.	26
4.14	Pantalla ubicar inicial, no existe ubicación.	26
4.15	Pantalla ubicar inicial donde el destino tiene ya mercancía.	26
4.16	Pantalla ubicar inicial donde hay error en la ubicación origen.	27
4.17	Pantalla generador pedidos.	27
4.18	Código generador pedidos.	28
4.19	Código generador pedidos por cada nivel elegido.	28
4.20	Código que elige la cantidad de cada producto en un pedido.	29
4.21	Pantalla simular pedido sin pedidos pendientes.	29
4.22	Vista pedidos en base de datos.	29
4.23	Pantalla simular pedido con éxito.	30
4.24	Pantalla simular pedido cogiendo un producto.	30
4.25	Vista pedidos en base de datos con pedido ya comenzado.	30
4.26	Pantalla simular pedido con ubicación vacía.	30
4.27	Pantalla simular pedido esperando más producto en fija.	31
4.28	Pantalla menú como usuario	32
4.29	Pantalla sugerir reposición al usuario con éxito.	32
4.30	Pantalla sugerir reposición al usuario con error.	33
4.31	Pantalla sugerir reposición al usuario completo.	33
5.1	Contraseña o usuario erróneo.	35
5.2	Sugiere la reposición con éxito.	37

Resumen

El objetivo principal de este proyecto es conseguir crear un sistema que simule la gestión de un almacén, tanto la parte de almacenamiento de material y entrada de productos nuevos, como la simulación de realizar pedidos donde se solicitará material que es almacenado en el almacén y donde deberemos controlar el stock.

El proyecto se centrará en dos grandes partes:

- El almacenamiento de los datos del material que se encuentra en el almacén, la información que van a contener las bases de datos sobre estos, y el control del stock donde se informará en todo momento del stock disponible y reservado para pedidos.
- La simulación de realizar pedidos donde reservaremos material disponible para los pedidos, asegurándonos que no se podrá reservar más de la cantidad disponible para que todos los pedidos puedan completarse con éxito.

Abstract

The main objective of this project is to create a system that simulates the management of a warehouse, both the material storage part and the entry of new products, as well as the simulation of placing orders where the material that is stored in the warehouse will be requested and where we will have to control the stock.

The project will focus on two main parts:

- The storage of the data of the material that is in the warehouse, the information which the databases will contain about these and the control of the stock. It will always be necessary to inform about the availability of products as well as those that have been kept for orders.
- The simulation of placing orders where we will keep available material for orders, making sure that no more than the available quantity can be reserved so that all orders can be completed successfully.

Resumen extendido

Aunque día a día vemos evolucionar la tecnología a pasos agigantados, no crece por igual este desarrollo tecnológico en todos los sectores de nuestra sociedad. Abordando el sector que nos interesa en nuestro proyecto que sería el industrial, conocemos miles de grandes empresas que han evolucionado de manera sorprendente y en las que, en algunas de ellas, cuesta encontrar trabajadores físicos moviéndose por sus almacenes. Sin embargo, no en todas es así, ya que también podemos encontrar el otro extremo, pequeñas o medianas empresas de las que se conocían como “familiares”, aquellas donde trabajan los hijos de otros trabajadores o familiares muy cercanos y donde no se empleaba apenas presupuesto en la evolución tecnológica, y en las que, hasta hace muy poco se ha llevado un control de stock o datos importantes de manera física, apuntándose en algún lugar común o directamente por escrito.

Toda empresa busca optimizar su tiempo y dinero, tratando de mejorar todos sus procesos o tareas de manera que se aproveche al máximo el tiempo de la jornada de cada empleado. La gran evolución tecnológica en los almacenes se consiguió con la creación de sistemas de gestión y almacenamiento de datos los cuales podían mostrarnos de manera rápida y sencilla el contenido total de un almacén, los datos de cada uno de los productos que allí se almacenaban, o simplemente saber un lote o fecha específica de una ubicación.

El objetivo de este proyecto es crear un sistema que facilite la adaptación de estos almacenes a la digitalización, no sólo ofreciéndoles a los usuarios un producto rápido y efectivo, sino también fácil de manipular, de entender, de saber utilizar por las personas menos habituadas al uso de las tecnologías, y en definitiva, de todos los trabajadores que lo usen.

Este sistema tratará de una interfaz de fácil uso para los usuarios, donde puedan realizar la totalidad de las acciones asignadas a su rol de la manera más intuitiva posible, y que les agilice sus tareas dentro del almacén. Como objetivos principales, el sistema nos proporcionará en todo momento un control del stock del almacén con sus datos correspondientes almacenados y donde se realizarán comprobaciones para cada acción, unido con la gestión de la simulación de creación de pedidos, cantidades reservadas para estos, nunca superando las cantidades disponibles en el sistema, y otras actividades como cambios de ubicación, nuevas entradas de material, etc.

Capítulo 1

Introducción

1.1 Motivación

Debido a las últimas catástrofes que estamos viviendo, como una pandemia mundial, la crisis energética, de materias... ha tocado adaptarse en todos los aspectos de la vida cotidiana a nuevas normas o formas de actuar, y con esto, las grandes empresas han valorado cambiar su manera de trabajar ante muchas situaciones, y una de las más afectadas han sido los negocios, grandes o pequeños. Muchos de estos últimos no han sido capaces de sobrevivir a toda esta situación y se han visto en la obligación de desaparecer, y otros no tan pequeños han evaluado resultados de ventas durante y después de la pandemia y han optado por cerrar algunas de sus tiendas físicas puesto que no les merecía la pena. Esto ha sido debido a que la vida ha cambiado mucho estos últimos años y nos hemos acostumbrado a que podemos tener prácticamente al día siguiente en la puerta de nuestra casa cualquier producto que decidamos comprar con un solo “click” en nuestros dispositivos, sin la necesidad de movilizarnos a una tienda, esperar que haya stock, compararlo, etc. Cada vez estas empresas nos lo ponen más sencillo con comparaciones entre productos, envíos rápidos y gratuitos, y comodidades diferentes para pagos. Y en estos detalles es donde se está buscando mejorar diariamente, en un mundo donde un cliente puede elegir tu servicio o no solo por la velocidad del envío, la calidad tanto de este como del producto o las comodidades que le puedas ofrecer, la lucha por mejorar cualquier mínimo detalle es clave para ser finalmente la empresa elegida.

Ahora que ya sabemos la importancia que tiene para un negocio los tiempos necesarios para ejecutar cada una de las tareas, podemos profundizar en nuestro tema y la buena optimización que debe tener este, la gestión informatizada del stock de productos en un negocio o almacén.

Quizás para una pequeña tienda que venda productos contados al día no sea tan complicado llevar un control de stock incluso manual, apuntando las ventas, los productos importados, y haciendo cálculos el responsable para prever la duración de estos y cuando comprar más. Sin embargo, cuando se va aumentando el número de trabajadores, de ventas, de espacio, o si se trata de una fábrica o almacén los cuales van a mover grandes volúmenes de material, se necesita tener todo informatizado en un sistema que haga automáticamente recuentos, cálculo de cantidades reales, ayude a aprovisionar las futuras entradas, en definitiva, tener control total del stock. El objetivo de este proyecto es aprovechar las nuevas tecnologías para poder optimizar el trabajo de las personas que lo necesiten, ya que muchas

empresas aún no poseen sistemas actualizados donde sacarle todo el provecho posible y continúan con los primeros sistemas de gestión, robustos y lentos, y actualmente sin apenas mantenimiento.

1.2 Objetivos

Este proyecto se basa en crear un sistema que gestione la información completa de la mercancía de un almacén, abarcando tanto la gestión de la base de datos, sus relaciones, y la interfaz con la que el usuario pueda manipular esta. El objetivo principal del proyecto es facilitar las tareas a cualquier persona que quiera hacer una operación con estos datos, modificar alguno de ellos o consultar información, por lo que se ha buscado conseguir en todo momento una interfaz muy sencilla y fácil de entender.

Alguno de los objetivos principales del proyecto eran:

- Estudio y desarrollo de técnicas de programación en SQL.
- Desarrollo de técnicas de programación donde a través de Python se pudiesen realizar las peticiones a la base de datos.
- Familiarizarse con librerías de creación de interfaces y contenido visual.
- Conceder al almacén una lógica donde se consiga un flujo de actividades sin conflictos, como la creación de ubicaciones fijas, aportando ideas que puedan facilitar el uso a las personas encargadas de utilizarlas.
- Añadir mejoras al proyecto que vayan surgiendo y que se estudie su beneficio tanto para usuarios como para el propio sistema.

1.3 Organización de la memoria

La memoria se encuentra dividida en cinco capítulos:

1. **Base teórica:** En él se detalla tanto el funcionamiento del almacén, organización física de este, contenido de los pasillos, alturas, etc, y también las herramientas utilizadas, lenguajes de programación, tecnologías, software, etc.
2. **Desarrollo del sistema:** En este capítulo se explica la relación entre las tablas de la base de datos, los datos que comparten, de la manera que lo hacen, y posteriormente cada una de ellas por separado y la finalidad que tienen en el sistema.
3. **Interfaz:** En este se explica cada una de las pantallas que el usuario va a poder encontrarse durante su uso, las opciones a las que se van a enfrentar dependiendo de los datos que se introduzcan, y los diferentes mensajes que estas mostrarán.
4. **Casos de uso:** En él se detalla cada una de las situaciones a las que se va a poder enfrentar la interfaz, viendo los casos positivos y negativos, buscando el éxito de la aplicación y el fallo y como se comporta ante estos.
5. **Conclusiones y líneas futuras:** Por último en este capítulo se expresan las ideas finales que ha generado este proyecto y como se podría mejorar en el futuro.

Capítulo 2

Base teórica

En este apartado vamos a distinguir dos grandes partes de las que vamos a detallar a continuación: por un lado la base teórica de cómo se gestiona un almacén, de cómo se reparte física y digitalmente, de que datos necesitamos, de cuales van a ser de vital importancia, y cómo se va a estructurar en el sistema. Por otro lado, trataremos la base teórica de las herramientas utilizadas para la creación del sistema y su mantenimiento.

2.1 Funcionamiento del almacén.

Cada país, cada ciudad, cada empresa y cada almacén es un mundo, uno totalmente distinto al de al lado y cuyos objetivos y sus maneras de trabajar, sin embargo, pueden ser comunes. En esta parte se va a explicar la disposición del almacén que se ha creado para este sistema, escogida por su sencillez, la facilidad de ser entendida por el usuario, de gestionar por las bases de datos, de explicar en un documento por escrito, y de documentarse para que cualquier persona que quiera comenzar a utilizarla no tenga problemas en ello.

El almacén se ha creado con 90 ubicaciones predefinidas desde un inicio, con la totalidad de estas vacías de manera inicial, y repartida en distintos pasillos, alturas y posiciones, que ahora veremos con detalle.

Una ubicación o posición será cada uno de los huecos existentes en el almacén para colocar en su interior un “pallet”, y del cual tendremos información detallada durante todo el tiempo que se encuentre dentro de este. Solo podremos tener un producto por ubicación, del tipo que sea, sin mezclas de productos ni cantidades.

Con esta información se dispondrá de una base de datos con 90 filas creadas previamente vacías, con su número de ubicación real añadido. Este número de ubicación real va a ser la manera de identificar cada uno de los huecos de nuestro almacén.

Se dividirá el almacén en 3 pasillos distintos, ya que se ha considerado un tamaño correcto para las pruebas de este proyecto y para una mejor visualización. Para un uso real, el almacén dispondrá de más ubicaciones seguramente, pero la manera de gestionar las bases de datos es exactamente la misma. Cada uno de estos pasillos contendrá 30 ubicaciones, siendo estos totalmente iguales.

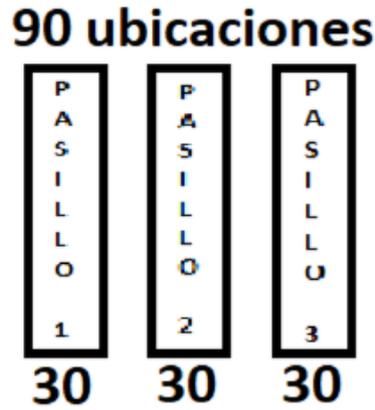


Figura 2.1: Reparto de las 90 ubicaciones del almacén en pasillos.

Cada uno de los pasillos tendrá 10 ubicaciones a lo largo, con 3 alturas distintas, la altura cero que será el suelo, la altura uno que estará encima de esta, y la altura dos, encontrándose las tres totalmente de manera vertical. Esto se repite con los diez huecos que se encuentran de manera horizontal, pegados unos a otros.



Figura 2.2: Reparto de las 30 ubicaciones dentro de un pasillo.

De esta manera podemos apreciar el siguiente esquema de como quedaría estructurado el sistema de control de ubicaciones.

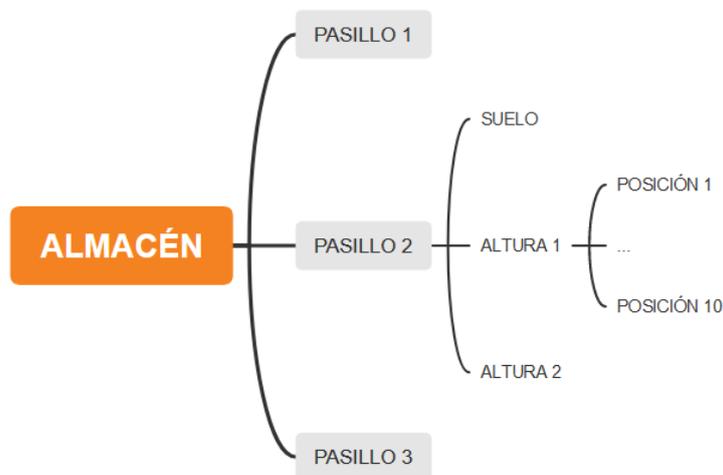


Figura 2.3: Reparto detallado de todas las ubicaciones.

A estas ubicaciones se le debe asignar un sistema de nomenclatura, ya que cada una de ellas debe tener un identificador único. Este identificador se va a tratar de un número de 4 dígitos. El almacén se basa en 3 pasillos distintos, desde el 1 hasta el 3, con 3 alturas cada pasillo (el suelo sería la 0, altura 1 y la 2), y con 10 posiciones cada altura, desde la posición 1 a la 10. La posición va a ser siempre un número de dos cifras, siendo la 01 la primera y terminando en la 10. De esta manera, se gestiona los valores de las ubicaciones como un número entero con estructura mostrada posteriormente:



Figura 2.4: Explicación de la nomenclatura de cada ubicación.

De esta manera se va a obtener una primera ubicación del sistema del tipo 1001, donde se indica el pasillo 1, la altura 0, y la posición 01. Continuando ese pasillo terminaremos en la ubicación 1010, la última del pasillo 1, altura 0, con número de posición 10. Poniendo un ejemplo de la parte media del almacén, el pasillo 2, con altura 1 y posición 5 se gestionaría como la posición con identificador 2105. Finalmente la última posición del sistema sería el pasillo 3, altura 2, posición 10, con un valor de 3210.

Por lo tanto, toda esta parte relacionada con la identificación de las ubicaciones reales del sistema sirven para simular el espacio físico que poseíamos en un almacén real. De esta manera sabremos en todo momento el producto que hay en un hueco real del almacén, o en qué ubicación se encuentra un pallet específico que necesitamos.

La siguiente parte teórica que debemos entender de nuestro sistema es que cada producto dado de alta en este debe tener asignada una ubicación fija, la cual contendrá siempre ese mismo producto o nada en caso de no haber más material de ese, pero nunca otro producto distinto. Esta ubicación fija se utiliza para que los trabajadores puedan tener localizados ciertos productos en ciertas ubicaciones, de manera que se familiaricen con ellos y puedan optimizar la velocidad a la hora de realizar pedidos, sabiendo que van a encontrar un mismo producto en un mismo lugar las veces que sean necesarias. Las ubicaciones fijas serán las alturas cero de todo el almacén, de modo que no sea necesaria maquinaria para llegar a ellas. Para lograr entenderlo vamos a utilizar un ejemplo dentro del sistema. Imaginemos

que en nuestro almacén vamos a dar entrada al producto “pan”. En ese instante se le asigna una ubicación fija en el suelo, y a partir de ese momento todos los pedidos que necesites este producto lo solicitará de esa ubicación. En ella no va a poder encontrarse ningún otro producto, de manera que cuando se acabe la mercancía de un pallet de ese producto, se repondrá con otro de ese mismo, enviándonos siempre a esa ubicación y no a otras donde haya producto.

Esto se hace así para evitar pallets abiertos o incompletos en alturas, ya que un escenario real físico, puede causar problemas por riesgos de caídas, de pérdidas, o de desajustes en el sistema.

Por lo que hemos visto, las ubicaciones fijas solo se podrán dar en el suelo, por lo que en este almacén de 90 ubicaciones, solo se dispondrá de 30 productos distintos con sus 30 ubicaciones fijas correspondientes. Esto se podría cambiar, permitiendo que cualquier ubicación pudiese ser fija, sin embargo no es recomendable ya que siempre que un pedido tenga un producto con una ubicación fija en altura, se pedirá a los empleados que recojan el producto de ella, necesitando máquinas continuamente y dificultando la velocidad al realizar pedidos. Por ello es recomendable elegir como fijas primero todas aquellas ubicaciones de suelo hasta completarlas.

Para explicar la diferencia entre ambos tipos distintos de ubicaciones, llamaremos ubicación real a cada hueco físico del almacén, que tendrá un número de pasillo, una altura y una posición en este. En cada una de estas ubicaciones se puede almacenar cualquier producto y se irá intercambiando los datos en el caso de que se ubique un pallet de una altura en otra. Sin embargo, la ubicación fija es aquella ubicación en el suelo que se le asigna a un producto al darle entrada en el sistema y que siempre va a poseer cantidad de este mientras haya en el almacén, y de la cual se cogerá la cantidad de los pedidos realizados, no de ninguna otra.

Con esta base ya interiorizada, podemos ver detalles antes de adentrarnos en las explicaciones de las clases y bases de datos. Cuando nosotros ubicamos un pallet en otra ubicación distinta de la que procedía, lo que vamos a hacer en la base de datos es coger toda la información de la fila origen, copiarlo en la destino exceptuando la ubicación real que será la nueva, y eliminar los datos de la fila origen, dejando solo la ubicación real donde ahora aparecerá como vacía, sin pallet, sin datos. Durante todos los cambios de ubicaciones que sufra un mismo pallet, podrá cambiar infinitud de veces la ubicación real en la que se encuentra, pero su fija será siempre la misma, se convierte en irremplazable, y será donde acabará ubicado para poder ser utilizado para pedidos.

2.2 Herramientas utilizadas.

El sistema de gestión de bases de datos utilizado para este proyecto ha sido SQL, más concretamente MariaDB, una bifurcación de MySQL creada en 2009 que cuenta con el modelo de base de datos relacional. Ambas comunidades de desarrolladores publican en sus páginas respectivas las pruebas de referencia para instalaciones con diferentes motores de base de datos, y en estas pruebas se ha podido apreciar que el rendimiento de una base de datos depende fuertemente de la medida en que la implementación del software se ha ajustado a unas necesidades concretas. En nuestro caso vamos a necesitar peticiones a nuestra base de datos muy básicas, como crear un producto que vamos a incluir, añadir un producto nuevo que ha entrado al almacén, eliminar otro que quitamos o simplemente descontar una cantidad del sistema, por lo que teniendo claras nuestras necesidades, nuestros objetivos y estudiando la dificultad que una petición puede suponer para estos, se ha elegido MariaDB. Uno de los factores que ha hecho que se elija MariaDB es que sigue siendo completamente de código abierto, mientras que MySQL tiene ahora módulos de código cerrado. MariaDB ofrece un mejor rendimiento, es más rápido y más ligero que MySQL gracias a sus 12

nuevos motores de almacenamiento y a sus más de 200.000 conexiones, lo que agilizará el sistema bastante al ejecutarse desde una Raspberry Pi. MariaDB cuenta con un thread pool dinámico, que permite al servidor optimizar sus recursos retirando los hilos inactivos, por lo que se consigue una velocidad considerablemente mejorada, una replicación mejorada y actualizaciones más rápidas.

La herramienta de administración de bases de datos utilizada ha sido DBeaver, que es una aplicación de software cliente de SQL. Proporciona un editor que soporta el autocompletado de código y el resaltado de sintaxis. Proporciona una arquitectura de plugins que permite a los usuarios modificar gran parte del comportamiento de la aplicación para proporcionar funcionalidad o características específicas de la base de datos que son independientes de la base de datos. Las características más importantes por lo que se utiliza DBeaver han sido:

- Ejecución de consultas SQL.
- Navegador/editor de datos con una gran cantidad de funciones.
- Resaltado de sintaxis y compleción automática de SQL.
- Navegar y editar la estructura de base de datos (metadatos).
- Gestión de scripts SQL.
- Exportación/migración de datos.
- Importación, exportación y backup de datos (MySQL y PostgreSQL).

El proyecto se basaba más en la parte de gestión de sistema y base de datos que en la interfaz del usuario, por lo que no se ha empleado ninguna herramienta de alto nivel de dificultad o con acabados visuales profesionales. La interfaz se ha creado con la biblioteca Tkinter, para el lenguaje de programación Python. Se considera un estándar para la interfaz gráfica de usuario para Python. Para la creación de interfaces con esta biblioteca, debemos seguir varios pasos: la creación del marco donde se ubicará la interfaz, la configuración de los atributos, y la asignación de funciones a diferentes elementos de esta.

Capítulo 3

Desarrollo del sistema

Una vez explicado el sistema utilizado y el por qué de su elección, vamos a detallar el reparto de esta base de datos, la relación entre sus tablas, y el significado de los atributos de cada una de las clases. Vamos a comenzar explicando las relaciones entre cada una de las tablas, las dependencias existentes, y el flujo total de una ejecución, y tras ello, se detallará cada una de las tablas individualmente.

3.1 Relación entre las tablas de la base de datos

Antes de detallar cada una de las tablas que forman nuestro sistema, conviene conocer la diferencia entre los dos “mini sistemas” que forman la parte de almacén y la de pedidos, y los valores de cada una, donde podemos apreciar de manera más completa todas las relaciones existentes y la dependencia que se tiene a la hora de ejecutar acciones.

- De manera diferenciada tenemos la tabla “usuarios”, la cual no depende de ninguna otra y solo afecta a la parte de interfaz, como luego veremos, para mostrarnos distintas opciones según el rol.
- Por la parte de pedidos, tenemos *pedidos pendientes* que almacena todos los pedidos generados, comprueba si hay alguno inicializado para elegir este el primero, añade los completos a *historial pedidos*, y aporta información de los productos reservados para pedidos a la tabla *control stock*, para poder llevar un control real de los productos y del stock restante.
- Por la parte del almacén físico, tenemos *almacen* que contendrá la información del estado de un almacén real, sus ubicaciones tanto vacías como llenas, en caso de contener material se tendrán los datos de este producto, obtenidos en parte de la tabla “productos”, y se aporta información también a *control stock* para calcular las cantidades físicas reales y poder gestionar el stock.

3.2 Tabla Usuarios

La primera tabla a la que vamos a acceder en una ejecución real, y por lo tanto la primera que vamos a explicar, va a ser la tabla usuarios. La primera pantalla con la que vamos a encontrarnos al

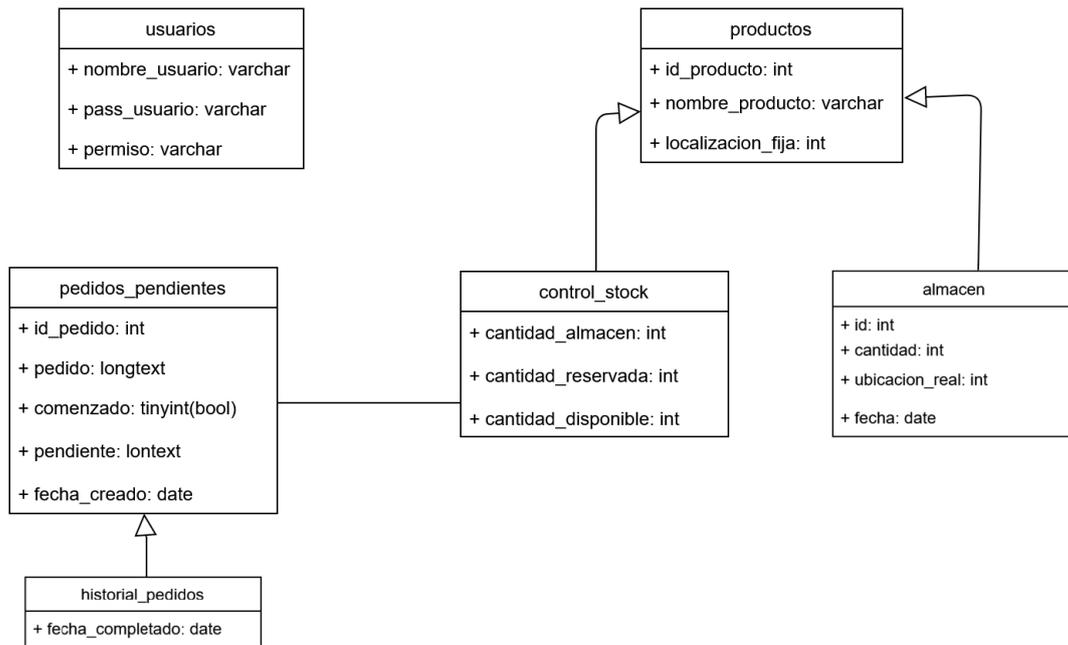


Figura 3.1: Relación entre las tablas del sistema.

iniciar la aplicación va a ser un menú donde debemos iniciar sesión. Nuestra tabla va a contener los distintos usuarios que se den de alta en el sistema, los cuales dispondrán de un nombre de usuario, una contraseña, y su rol asignado. Dependiendo de este rol, el usuario que inicie sesión va a tener unos permisos u otros, y por lo tanto le cargará una pantalla diferente con las opciones que a este le sea permitido utilizar.

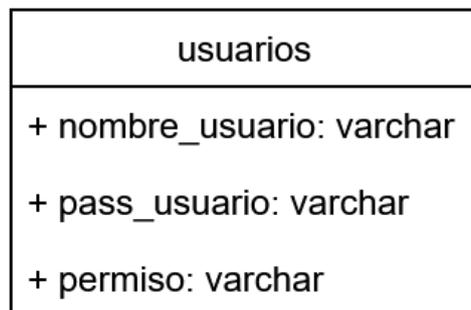


Figura 3.2: Declaración de la tabla usuarios de la base de datos.

En el caso del proyecto, hemos creado dos permisos: administrador y usuario. Más adelante, cuando expliquemos la interfaz, veremos las diferencias entre estos permisos y las opciones que tienen disponibles los usuarios a diferencia de los administradores. Para el caso práctico sólo hemos creado un usuario de cada tipo, y en la siguiente imagen se muestra el contenido de dicha tabla:

Solo existen esos dos permisos y el sistema no está preparado para soportar otros distintos, lo que se sí puede aceptar es un número sin límite de usuarios creados, ya que tan solo puede iniciar sesión uno por ejecución.

	ABC nombre_usuario ▼↑	ABC pass_usuario ▼↑	ABC permiso ▼↑
1	admin	admin123	admin
2	user	user123	user

Figura 3.3: Tabla usuarios de la base de datos.

3.3 Tabla Productos

Una vez superado el proceso de inicio de sesión, pasamos a la parte más importante del proyecto y la pieza sobre la que gira este, el stock del almacén. En esta primera clase nos centramos solo en la información básica de los productos que se encuentran en este momento en el almacén. En esta tabla se podrá añadir un producto nuevo con su id y ubicación correspondiente, pero no necesariamente añadir una cantidad de este producto al almacén, solo darlo de alta para futuras recepciones de este.

productos
+ id_producto: int
+ nombre_producto: varchar
+ localizacion_fija: int

Figura 3.4: Declaración de la tabla usuarios de la base de datos.

Esta tabla se basa en un id, el nombre del producto ligado a este id, y una localización fija para cada producto. Todos estos datos son únicos, no pudiéndose repetir ningún nombre ni ninguna localización.

En la parte de base teórica se ha explicado cómo funcionan tanto las localizaciones reales del almacén como las fijas a las que se le asigna un producto. Con esta información, y volviendo a los datos de esta tabla, la ubicación fija va a ser aquella en la que siempre vamos a encontrar el producto aquí referido y ningún otro.

	123 id_producto ▼↑	ABC nombre_producto ▼↑	123 localizacion_fija ▼↑
1	1	pan	1.001
2	2	agua	1.002
3	3	pasta	1.003
4	4	fruta	1.004
5	5	verdura	1.005
6	6	legumbres	1.006

Figura 3.5: Tabla productos de la base de datos.

Para explicar el ejemplo de esta imagen, un producto, por ejemplo “pan”, siempre que disponga de unidades en el almacén, va a tener que ser en esa ubicación. Si viene mercancía de este producto repartida en 10 ubicaciones, la más antigua (se seguirá el orden de fabricación) se ubicará en su fija

(en este caso 1001), y las demás repartidas por las alturas del almacén. De esta manera se respetarán las ubicaciones de suelo para productos diferentes, para intentar que cada uno de ellos tenga acceso desde el suelo, sin otras máquinas. A la hora de realizar pedidos, como veremos posteriormente, se cogerá producto de su ubicación fija hasta agotarlo, y se ubicará el siguiente contenedor al suelo por orden de antigüedad.

3.4 Tabla Almacén

Conociendo el sistema de gestión de los productos, vamos a profundizar en la clase que simula nuestro almacén. Esta contiene las noventa ubicaciones mencionadas anteriormente, y además un conjunto de datos que veremos a continuación.

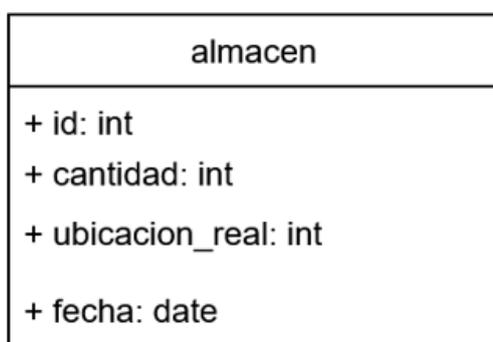


Figura 3.6: Declaración de la tabla almacén de la base de datos.

En esta clase, el valor que más información nos va a aportar y del cual va a depender el almacén va a ser la *ubicación real*. En nuestra base de datos se crea esta clase con las noventa ubicaciones antes mencionadas de manera previa al inicio de la ejecución, de manera que simule el funcionamiento de un almacén real, existiendo sus ubicaciones vacías hasta que se empiece a trabajar con él. Esta clase mantendrá el estado actual del almacén mientras la base de datos exista, y en ella se podrán añadir productos en ubicaciones, se podrán mover mercancías existentes de una posición a otra, y descontar unidades de una ubicación cuando se realice un pedido. Se detalla más adelante la manera de procesar los datos en los casos justo antes mencionados, por el momento vamos a dejar explicados los valores que esta clase guarda para cada ubicación. Las ubicaciones que se encuentren vacías sólo contendrán un id de tipo entero, y la *ubicación real*, que contendrá un valor de los explicados en el punto anterior, dependiendo del pasillo, altura y posición. En aquellas ubicaciones que sí contengan producto, se podrá apreciar además el id del producto almacenado, su nombre, y su localización fija, correspondientes con el punto anteriormente explicado en el apartado de productos, y junto a ello se almacenará la *cantidad* de ese producto en esa ubicación, y la fecha en la que esa cantidad fue dada de alta en el sistema.

Con esta parte daríamos por explicada la parte del almacenamiento de los productos en el sistema, sus cantidades, y su localización. Pero los almacenes tienen como objetivo guardar material durante un tiempo hasta que ese producto es vendido o movido a otro lugar, por lo que, a parte del sistema de almacenamiento, también vamos a generar un sistema de control de stock, que simule pedidos, que simule la retirada parcial de producto de sus ubicaciones, que simule empleados realizando un

trabajo de “picking” donde cogen una cantidad dada de distintas ubicaciones y en el que vamos a tener en tiempo real el control de las cantidades actuales en el sistema, vamos a saber la cantidad guardada para pedidos y la disponible para pedidos futuros, y así poder prever las faltas de stock, evitándose y consiguiendo una mayor eficiencia de entregas. Con todo ello, vamos a adentrarnos en la parte de la gestión de los pedidos y su control de stock.

3.5 Tabla Pedidos Pendientes

Antes de explicar la tabla *control stock* que está relacionada directamente con toda la parte anterior, nos interesa ver primero la parte de los pedidos para poder llegar a ella sabiendo de dónde viene cada dato y que todas las tablas tengan la relación esperada.

pedidos_pendientes
+ id_pedido: int
+ pedido: longtext
+ comenzado: tinyint(bool)
+ pendiente: lontext
+ fecha_creado: date

Figura 3.7: Declaración de la tabla pedidos pendientes de la base de datos.

Posteriormente veremos la manera en la que se generan los pedidos, quién puede hacerlo, cómo se tratan y cómo se realizan, pero por el momento vamos a explicar sólo los datos que se almacenan de estos una vez se crean.

- El “id” del pedido es único e irremplazable, y es la única manera de poder identificar un pedido, recuperarlo o moverlo a otras tablas.
- El “pedido” va a ser un conjunto de datos similar a un diccionario, el cual contenga su clave, que en este caso será el id del producto a retirar, y su valor, el cual contenga la cantidad. De manera más visual, un ejemplo de un pedido sería algo así:

{pan: 5, agua: 3}

donde se nos solicita 5 unidades del producto “pan”, y 3 unidades de “agua”. Sin embargo, para el sistema es más fácil manipular datos por su id que por su nombre, por lo que un ejemplo real de este pedido sería:

{1: 5, 2: 3}

donde el 1 representaría el id del “pan” y el 2 el del “agua”.

- El valor “comenzado” será un tipo booleano el cual nos indicará si el pedido se encuentra iniciado o no, es decir, ya se habría cogido alguna cantidad de alguna ubicación. Mostraremos

más adelante el funcionamiento a la hora de simular pedidos, pero para esta explicación, en caso de haber un pedido comenzado, va a ser el primero en simularse sin importar otros datos. También mencionar que, debido a que se va a ejecutar siempre que lo haya uno que esté como “comenzado”, sólo puede haber uno con este valor a “True” de manera simultánea. Es imposible encontrar dos pedidos distintos como “comenzado”.

- El valor “pendiente” contendrá la parte del pedido que queda por realizar. Como hemos explicado, no puede haber dos pedidos distintos como “comenzado”, por lo tanto el valor “pendiente” será igual al pedido completo, menos en el caso de haber uno ya comenzado, que nos mostrará la parte de pedido que falta por recoger.
- La fecha en la que se generó el pedido se almacena en el valor “fecha creado”.

3.6 Tabla Historial de Pedidos

Esta es una tabla que contiene el historial de los pedidos completos. En ella no se almacena ningún pedido pendiente ni comenzado sin terminar, por lo que nos servirá para tener un registro para alguna futura consulta, o en caso de necesitar algún dato tanto nosotros como cliente.

historial_pedidos
+ fecha_completado: date

Figura 3.8: Declaración de la tabla historial pedidos de la base de datos.

Esta tabla contendrá:

- La “fecha completado” donde encontraremos la fecha en la que se finalizó el pedido, y por lo tanto, en la que se incluyó en esta tabla.
- Por otro lado, los datos principales heredados de la tabla pedidos pendientes, como sería:
 - Un “id pedido” único, que nos facilitará su búsqueda e identificación.
 - El “pedido” que contendrá el diccionario previamente explicado y donde se podría revisar su contenido en un futuro.
 - La “fecha creado” donde se almacenará la fecha en la que se generó el pedido, independientemente del tiempo que haya pasado hasta finalizarlo.

3.7 Tabla Control Stock

Tras haber visto el almacenamiento y la gestión de los pedidos en la base de datos, podemos relacionar todos los elementos para el control de stock del que hablábamos. Esta tabla va a ser el “puente” de información entre los dos grupos diferenciados hasta ahora, el almacén y los pedidos. Para conseguir esta unión y estas dependencias, la tabla “control stock” va a pedir información antes de cada consulta o cambio a varias tablas, comprobando disponibilidades, comparando valores y cantidades, y muchas otras acciones.

control_stock
+ cantidad_almacen: int
+ cantidad_reservada: int
+ cantidad_disponible: int

Figura 3.9: Declaración de la tabla control stock de la base de datos.

Así podemos apreciar que la tabla va a contener heredados todos los datos de la tabla “productos”, junto a tres datos nuevos, todos con fin de almacenar cantidades. Esta va a ser un punto de unión constante entre la tabla “almacen” y la tabla “pedidos pendientes”, por lo que las opciones era crear estos atributos en alguna de las dos, o tener esta tabla separada para agilizar las consultas y los movimientos, y hacerlo también más visible. Los atributos nuevos que contendrá esta tabla serán:

- El valor “cantidad almacen” va a contener la cantidad real de cada producto que se encuentra en el almacén. Esta tabla accede a la tabla “almacen”, filtra por el id de cada producto sin importar la ubicación en la que se encuentre, suma todas las cantidades de estos, y las almacena. Así se consigue la “cantidad almacen”.
- El valor “cantidad reservada” va a depender de la tabla “pedidos pendientes”, ya que va a comprobar dentro de estos la cantidad de cada producto que está reservada para un pedido. Así vamos a poder obtener de cada elemento la cantidad bloqueada pendiente de coger.
- El valor “cantidad disponible” hace la diferencia entre los dos valores anteriores, restándole a la cantidad física que poseemos la cantidad reservada para pedidos, y así poder catalogarla como “disponible”.

Una duda que puede resultar aquí es si a la hora de generar pedidos se puede hacer de una cantidad mayor a la que el sistema dispone, y la respuesta es no. El sistema está desarrollado de manera que compruebe esta tabla antes de realizar cualquier acción. Se explicará posteriormente más detalladamente todas las comprobaciones realizadas previas a cada acción, pero en manera de resumen vamos a explicar cómo evitamos que nos falle esta tabla. A la hora de generar un pedido, este comprueba en la tabla primero la cantidad de ese producto en el almacén y después la de los pedidos ya generados. Si tras este cálculo vemos que queda cantidad disponible, el generador de

pedidos va a elegir una cantidad como máximo igual a la disponible, nunca superándose, evitando así el primer error. Cada vez que se añade algún producto al almacén, se actualiza inmediatamente “control stock”, elevando así la cantidad disponible para la próxima generación de pedidos. De la misma manera, cada vez que se genera un pedido, se actualiza en el momento la tabla para reducir la cantidad disponible y evitar conflictos.

Capítulo 4

Interfaz

Hasta ahora nos hemos centrado en toda la parte relacionada con la base de datos, el almacenamiento y la gestión de los datos, y ahora nos toca profundizar en la parte interfaz, en las opciones que tiene un usuario de la aplicación disponibles en la pantalla, lo que hace cada una de ellas, los movimientos de datos que genera en la base de datos, los permisos que se necesitan, como afectan al sistema cada una de estas acciones. . . En definitiva, el funcionamiento de nuestra aplicación.

Como introducción a esta parte visual del proyecto, este se iba a basar en un principio en una única interfaz, la cual iba a contener todas las opciones disponibles y desde la que íbamos a poder acceder a todo sin necesidad de más interfaces. Esta interfaz existe y será posteriormente explicada, sin embargo, no ha sido la única realizada. Durante el desarrollo de la aplicación te encuentras en innumerables ocasiones con la necesidad de realizar pruebas, comprobar resultados, si un paso se está comportando como debe o si se ha recibido el resultado esperado. Inicialmente, con peticiones SQL, se podían comprobar datos o funcionamientos, pero esta solución era poco eficiente y, aunque no tan importante, poco visual también. Así surgió la interfaz añadida a la original, una que su objetivo no era hacer cambios en el almacén ni completar pedidos, no era generar nuevas entradas o cambiar ubicaciones de mercancías, su objetivo era mostrarnos el estado del almacén en tiempo real. Esta interfaz mostraría el estado del almacén con dos colores para indicarnos si una ubicación estaba libre u ocupada. Esta idea surgió al realizarse continuamente movimientos dentro del almacén para comprobar el funcionamiento de utilidad como la de ubicar, y se decidió integrar en el proyecto.

Habiendo mencionado la existencia de estas dos interfaces, procedemos a detallar cada una de ellas, su funcionamiento y su distribución, comenzando por la de la visión del almacén en tiempo real.

4.1 Interfaz de la visión del almacén

Esta interfaz nos sirve únicamente para conocer el estado del almacén en el momento de la ejecución.

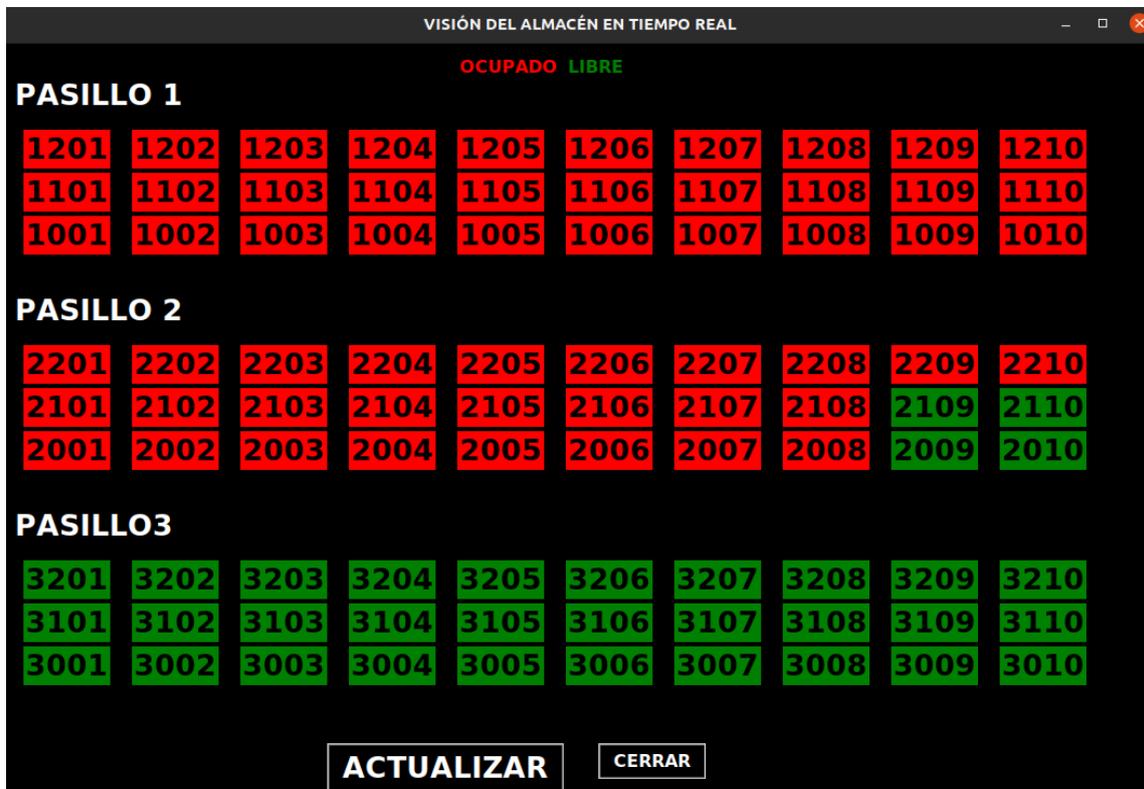


Figura 4.1: Interfaz de la visión del almacén en tiempo real.

En la imagen podemos observar una distribución de números, que hacen referencia a las ubicaciones reales, una leyenda de colores y dos botones. El botón “salir” cerrará la interfaz y el botón “actualizar” refrescará la imagen para mostrarnos el almacén si ha habido algún cambio. Hemos implementado este botón y no hemos hecho que se actualice periódicamente ya que para las pruebas que realizamos no necesitábamos ver los cambios cada pocos segundos sino puntualmente, cuando supiéramos que habíamos realizado algún cambio, y así poder ahorrarnos recursos no teniendo la máquina constantemente actualizando.

Por otra parte, observamos la leyenda de colores donde se nos indica que el color verde significa que esa ubicación está libre, y el color rojo ocupada.

En la parte de las ubicaciones, observamos un ejemplo visual de la explicación que vimos sobre las ubicaciones del almacén. Se divide el almacén en tres pasillos, claramente nombrados. Cada uno de estos contiene 10 posiciones, con 3 alturas distintas. Así el primer número es del pasillo, siendo 1001 y 3001 las dos primeras ubicaciones de los pasillos 1 y 3 respectivamente. El segundo número es la altura, siendo 2101 la posición de altura 1 justo encima de 2001, con altura 0. Por último, la posición 3201 estará al lado de la 3202, en el pasillo 3 y altura 2, pero en posiciones 01 y 02.

4.2 Interfaz de la gestión principal

Una vez entendido el funcionamiento de la interfaz anterior, nos adentramos con la aplicación principal del almacén, el simulador total tanto de gestión del almacén como de generación y simulación de pedidos.

4.2.1 Pantalla Login

Al ejecutarla nos vamos a encontrar en primer lugar con la pantalla de inicio de sesión, donde, como vimos en el apartado de “usuarios”, nos va a pedir iniciar sesión y por el momento solo tenemos dos usuarios creados, uno como administrador y otro como empleado. Para la explicación, vamos a utilizar el usuario “admin” ya que contiene todos los permisos y acciones que los empleados no tienen, y poder explicar todos.



Figura 4.2: Interfaz de la pantalla login del sistema.

Una vez iniciada la sesión como administrador, este es el menú que nos aparece, y ya podemos seguir desarrollando las siguientes acciones.

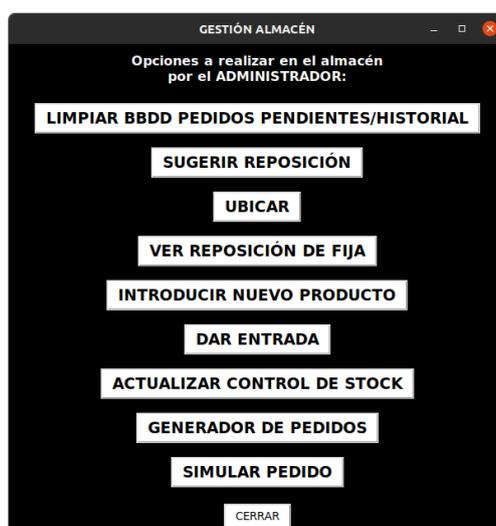
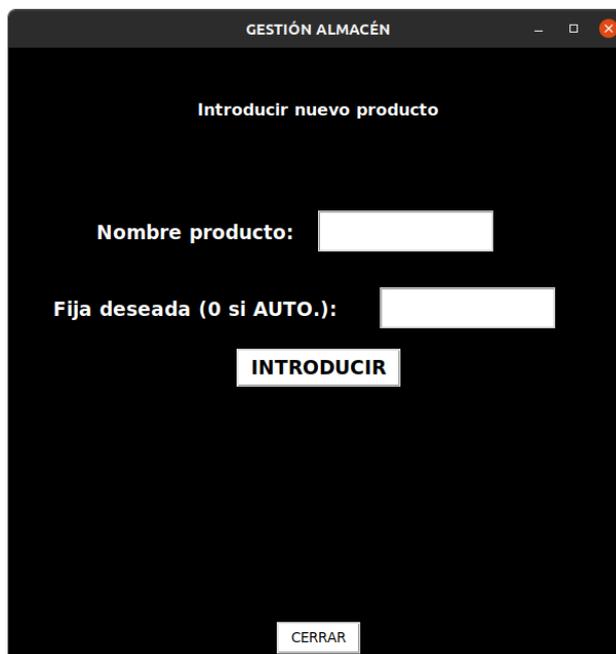


Figura 4.3: Interfaz de la pantalla menú del sistema.

4.2.2 Pantalla Nuevo Producto

En primer lugar vamos a tratar el alta de un nuevo producto que aún no existía en el almacén. Para ellos elegimos la acción “Introducir nuevo producto” donde nos encontraremos con la siguiente pantalla:



The screenshot shows a dark-themed window titled "GESTIÓN ALMACÉN". Inside, the heading "Introducir nuevo producto" is centered. Below it, there are two input fields. The first is labeled "Nombre producto:" and the second is labeled "Fija deseada (0 si AUTO.):". A prominent "INTRODUCIR" button is centered below the second field. At the bottom center, there is a smaller "CERRAR" button.

Figura 4.4: Interfaz de la pantalla introducir producto del sistema.

El nombre elegimos el que deseemos para nuestro nuevo producto, y en fija deseada se va a almacenar la ubicación fija de ese producto, que como hemos visto antes, será la ubicación predefinida de ese producto que será en el suelo siempre que sea posible (altura 0), y que podemos elegir manualmente si sabemos de alguna ubicación vacía en el suelo, o si indicamos el valor 0, nos buscará la siguiente ubicación fija disponible en el sistema y se la asociará. El nombre no sigue ninguna evaluación y puede ser cualquier texto, sin comprobar repeticiones o errores, sin embargo en la entrada de la ubicación es más robusto, no permitiendo ninguna ubicación que no sea altura 0, ni que no esté vacía en ese momento. En caso de elegir el valor cero para que lo elija aleatoriamente y no haya ninguna disponible, también nos mostrará un mensaje de error.

4.2.3 Pantalla Control de Stock

Este botón no tiene una “subinterfaz” al clickar, simplemente acciona el cálculo de nuevo del control de stock, actualizando cantidad real en el almacén, cantidad en pedidos, y cantidad disponible. Esta acción se realiza automáticamente por el sistema cada vez que se realiza una nueva entrada de mercancía, se genera un pedido, o se completa uno existente restando esa cantidad. Sin embargo, se ha añadido ese botón ya que el administrador del sistema puede modificar algún valor directamente desde la base de datos, eliminar datos, añadirlos o cambiar cantidades “a mano”, y utilizando ese botón tras ello, se volverá a recalculer todo el control de stock.

4.2.4 Pantalla Dar Entrada

En esta pantalla tendremos la oportunidad de incluir en el almacén una cantidad de un producto en especial siempre y cuando se haya registrado como producto previamente. Para poder añadir material al almacén se ha debido pasar por la pantalla vista anteriormente “Introducir nuevo producto”, y una vez registrado sin errores, ya se podrá añadir cantidad de este al sistema.



The screenshot shows a web application window titled "GESTIÓN ALMACÉN". Inside the window, the main heading is "Dar entrada de un producto". Below this heading, there are three input fields, each with a label to its left: "Ubicación:", "ID producto:", and "Cantidad:". Each label is followed by a white rectangular input box. At the bottom of the form area, there is a prominent white button labeled "DAR ENTRADA". Below the main form area, centered at the bottom of the window, is a smaller white button labeled "CERRAR". The background of the window is dark, and the text and buttons are white.

Figura 4.5: Interfaz de la pantalla dar entrada del sistema.

En esta pantalla nos encontramos que piden tres valores de entrada, la ubicación donde se va a colocar el material, el id de ese producto, y la cantidad que se va a añadir. La ubicación se debe añadir manualmente y no se puede calcular un hueco vacío ya que, por lo general, un operario va a ser el que coloque esa cantidad en la ubicación donde comprobará que no haya nada físicamente, sin embargo el administrador desde el ordenador no podrá comprobar esto. El sistema va a comprobar en primer lugar que el id del producto exista ya en el sistema, accediendo a la tabla “productos” y verificandolo ahí, y en segundo lugar que la ubicación esté vacía en sistema. La cantidad no se controlará de ninguna manera ya que no podemos saber que material es y si es erróneo o no. Si alguna de las dos comprobaciones falla, se mostrará un mensaje de error por pantalla, y si por el contrario es exitoso, nos lo hará saber también.

4.2.5 Pantalla Ver Reposición

En esta pantalla nos encontramos con una entrada de datos la cual nos pide una ubicación fija de algún producto.



Figura 4.6: Interfaz de la pantalla inicial de la opción ver reposición del sistema.

Con el dato introducido va a realizar una única tarea de comprobación, que es ver si existe esa fija en algún producto en el sistema. En caso de no coincidir, se mostrará un mensaje de error.



Figura 4.7: Interfaz de la pantalla ver reposición del sistema si no existe la ubicación fija.

En caso de existir pero no haber mercancía en el almacén se indicará un mensaje así.



Figura 4.8: Interfaz de la pantalla ver reposición del sistema si no queda mercancía.

Por último, en caso de si haber material de esa ubicación fija única, es decir, de ese producto al que está relacionado, nos mostrará un mensaje como el siguiente indicándonos ubicación real en la que se encuentra, cantidad, y fecha de entrada.



Figura 4.9: Interfaz de la pantalla ver reposición del sistema con éxito.

4.2.6 Pantalla Sugerir Reposición

En los almacenes es habitual que haya una persona encargada de ir bajando material de las alturas al suelo, a sus ubicaciones fijas, y que puedan ver la información que necesitan con una simple opción. En la subinterfaz de esta opción no encontraremos ninguna entrada de datos ni ningún botón, solo un mensaje según la situación del almacén. Tendremos tres posibles escenarios al elegir esta acción:

- Ninguna ubicación fija se encuentra vacía y, por lo tanto, no es necesario rellenarla con otra.

Todas las ubicaciones fijas con producto.

Figura 4.10: Interfaz de la pantalla sugerir reposición inicial del sistema.

- Alguna ubicación fija tiene su cantidad a cero y hay material en el almacén para hacer una reubicación.



Figura 4.11: Interfaz de la pantalla sugerir reposición con éxito del sistema.

- Alguna ubicación fija está vacía pero no hay material en el almacén de este producto.

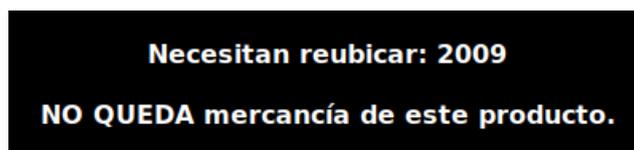


Figura 4.12: Interfaz de la pantalla sugerir reposición del sistema sin mercancía.

4.2.7 Pantalla Ubicar

Al seleccionar esta opción, nos encontraremos con una pantalla que nos solicitará dos datos de entrada para realizar el movimiento en el almacén, la ubicación origen y la ubicación destino.

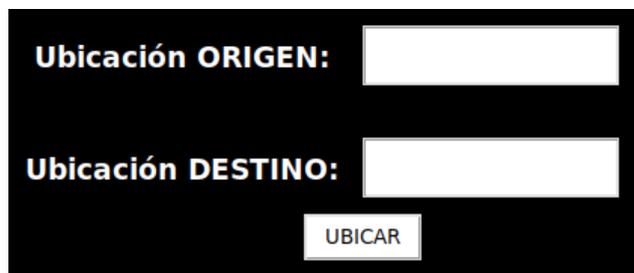


Figura 4.13: Interfaz de la pantalla ubicar inicial del sistema.

Para que se pueda completar la reubicación de una mercancía en otra ubicación, el sistema realizará un conjunto de comprobaciones. En primer lugar, se comprobará que existan en el almacén tanto la ubicación origen como la destino. No se comprobará su contenido en este paso, simplemente que aparezcan en el sistema.

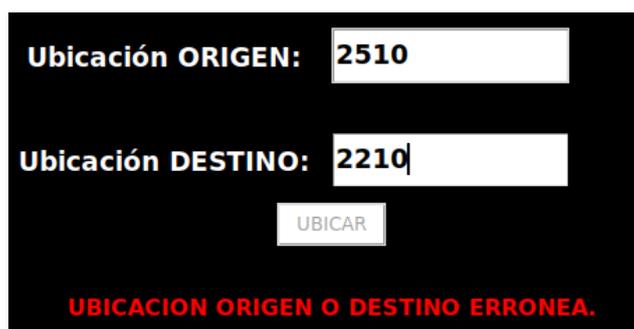


Figura 4.14: Interfaz de la pantalla ubicar del sistema donde no existe alguna ubicación.

En segundo lugar, se comprobará que el destino se encuentre completamente vacío, para así poder hacer el movimiento de mercancía.

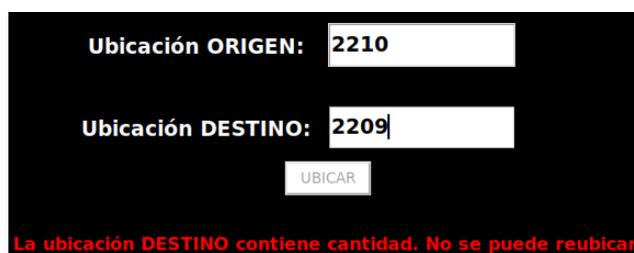


Figura 4.15: Interfaz de la pantalla ubicar del sistema donde el destino tiene ya mercancía.

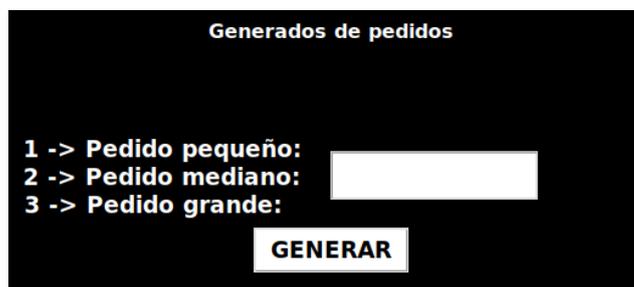
Tras estas dos comprobaciones, si sucede algún error será porque la ubicación origen no contiene mercancía, por lo que se evita actualizar datos de posiciones si ambas están vacías. También se mostrará por pantalla este error.

Problemas con la ubicación ORIGEN.

Figura 4.16: Interfaz de la pantalla ubicar del sistema donde hay error en la ubicación origen.

4.2.8 Pantalla Generador de Pedidos

Vista toda la parte relacionada a la gestión del almacén y sus productos, profundizamos en las acciones cuyo objetivo es gestionar los pedidos. En primer lugar tenemos la opción “Generador de pedidos”, la cual se encarga de crear pedidos nuevos y almacenarlos en su base de datos correspondiente. Para agilizar la simulación de estos procesos, se ha creado este generador automático que genera un tipo de pedido según la entrada.



Generados de pedidos

1 -> Pedido pequeño:
2 -> Pedido mediano:
3 -> Pedido grande:

GENERAR

Figura 4.17: Interfaz de la pantalla generador de pedidos del sistema inicial.

Podría haberse creado una opción en la que el administrador decidiese de que iba a formar parte el pedido, eligiendo este mismo los productos y la cantidad de estos a coger. Sin embargo, para que la simulación sea más sencilla de gestionar, se ha creado esta opción la cual solicita un número de entrada al usuario, entre el 1 y el 3, correspondiendo como se muestra en la imagen el número uno a la generación de un pedido pequeño, el dos a uno mediano, y el tres a uno grande. Cualquier otra entrada que no sea uno de estos valores mostrará un error, por lo que la entrada queda controlada en todo momento. Cada una de estas opciones tiene unos parámetros a seguir que se explicarán a continuación:

- **Opción 1 (Pedido pequeño):** En primer lugar se comprueba en el sistema la cantidad de cada producto que queda disponible en el almacén, y se selecciona de manera aleatoria entre 1 y 3 productos para recoger, y de estos, máximo 3 unidades de cada uno (en caso de que se encuentren en el almacén, si no es así se elegiría entre 1 y el máximo disponible).
- **Opción 2 (Pedido mediano):** Se comprueba en el sistema la cantidad disponible de cada producto, y se selecciona entre 4 y 6 productos en caso de haber existencias, y de cada uno un máximo de 6 unidades, también comprobando que haya esa cantidad disponible, si no es así, entre 1 y el máximo disponible.
- **Opción 3 (Pedido grande):** Todo es aleatorio. Se genera un pedido que puede elegir de manera aleatoria cualquier producto que haya disponible en el almacén, es decir pedidos de entre un producto y el máximo disponible, y de cada uno de ellos, puede elegir entre una unidad o el máximo disponible, pudiendo salir pedidos sencillos de pocos productos o un pedido que pida la totalidad del almacén.

```
#generador de pedidos sobre los productos y cantidad del almacen
#generador_pedidos(1) - pedidos pequeños, (2) - medianos, (3) - random
# 1 -> entre 1-3 productos, maximo 3 unidades por producto
# 2 -> entre 4-6 productos, maximo 6 unidades por producto
# 3 -> todo aleatorio. Maximo de unidades las disponibles en el almacen
def generador_pedido(nivel):
    stock_disp = stock_disponible()
    stock = {x:y for x,y in stock_disp.items() if y!=0}
    pedido = {}
    if not stock or nivel>3 or nivel<1:
        print("No queda material en el almacén o error de entrada.\nImposible generar pedidos.")
        hecho = False
```

Figura 4.18: Código donde se aprecia la manera de actuar para cada tipo de pedido.

En esta imagen se puede apreciar la manera de gestionar este trámite, comprobándose antes de cada ejecución el stock disponible en ese momento, si el número elegido para la opción de pedido está entre los disponibles, y si es así, proceder para cada opción de la manera correcta.

```
if nivel == 1:
    prod = random.sample(set(stock.keys()), random.randint(1, min(3, len(stock))))
    for i in prod:
        pedido[i] = int(cantidad_random(i, 3))
elif nivel == 2:
    prod = random.sample(set(stock.keys()), min(random.randint(4, 6), len(stock)))
    for i in prod:
        pedido[i] = int(cantidad_random(i, 6))
elif nivel == 3:
    prod = random.sample(set(stock.keys()), random.randint(1, len(stock)))
    for i in prod:
        pedido[i] = int(cantidad_random(i, 0))
```

Figura 4.19: Código donde se aprecia según la opción elegida, los resultados que va a elegir.

Según el nivel, se da una manera diferente de actuar.

- En el nivel 1 se elige entre un mínimo de un producto, y un máximo de 3, o menos si no quedan tantos productos disponibles.
- En el nivel 2 se elige entre cuatro y seis productos en caso de haberlos, ya que si no hay disponibles tantos, se elige el número de estos disponibles.
- En el nivel 3 se elige aleatorio entre un producto y el máximo disponible en ese momento en el almacén.

Esta es la manera de elegir el número de productos distintos que van a componer un pedido, sin embargo, para elegir el número de unidades que se van a coger de cada pedido se llama a la función “cantidad random”, la cual se va a encargar de calcular la disponibilidad de cada producto y que vamos a explicar a continuación con más detalle.

A esta función se le pasan por parámetro dos valores, el identificador del producto y la cantidad que se desea. Se hace una petición a la base de datos de la cantidad disponible del producto pasado por parámetro “ident”, y la almacenamos como disponible. Si el parámetro “cant” es cero, significa que no hay límite, podemos elegir cualquier cantidad de ese producto entre uno y lo máximo disponible. Si el valor “cant” es un entero, se elige aleatoriamente entre uno y ese valor, a no ser que sea mayor que lo disponible en el almacén, donde entonces se elegirá esa cantidad límite.

```

#random de la cantidad de un producto entre 0 y:
# la cantidad introducida, máximo disponible si 0
# cantidad_random(identificador, cantidad maxima 0 si maximo)
def cantidad_random(ident,cant):

    conex = db.cursor(buffered=True)
    conex.execute(f'''select cantidad_disponible from control_stock
                    where id_producto={ident}''')
    disponible = conex.fetchone()[0]
    db.commit()
    conex.close()

    if cant != 0:
        return min(random.randint(1,cant),disponible)
    else:
        return random.randint(1,disponible)

```

Figura 4.20: Código que va a elegir la cantidad de cada producto que tendrá cada pedido.

4.2.9 Pantalla Simular Pedido

Con la parte de generación de pedidos ya vista, nos queda ver cómo se ejecuta la simulación de cada uno de ellos. Al tratarse de una simulación donde no estamos dando lugar a errores físicos, hemos creado un orden de preferencias a la hora de simular pedidos, siguiendo siempre un flujo de acciones a realizar una detrás de otra. En primer lugar, y como es de esperar, se comprueba que haya pedidos generados al elegir esta opción, y de no ser así, se mostrará un mensaje que nos lo indique.

No hay pedidos pendientes.

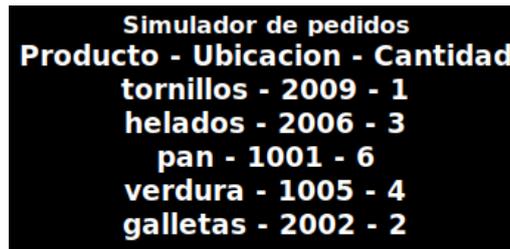
Figura 4.21: Pantalla simular pedido sin pedidos pendientes.

Tras esta comprobación y dando por hecho que existen pedidos en el sistema, se comprueba si hay alguno ya inicializado. Si existe alguno, se elige automáticamente como primer candidato y se continua con él, siguiendo su simulación y guardando sus nuevos estados mientras se encuentre como “no completado”. En caso de no existir ningún pedido inicializado, el pedido elegido como siguiente será aquel que lleve más tiempo en la base de datos, asegurándonos así que no se nos queda ningún pedido “rezagado”. Para verlo a detalle, generamos dos pedidos en el sistema.

id_pedido	pedido	comenzado	pendiente	fecha_creado
1	{21: 1, 16: 3, 1: 6, 5: 4, 12: 2}	0	{21: 1, 16: 3, 1: 6, 5: 4, 12: 2}	2022-08-24
2	{12: 1, 7: 3, 5: 1}	0	{12: 1, 7: 3, 5: 1}	2022-08-24

Figura 4.22: Vista pedidos en base de datos.

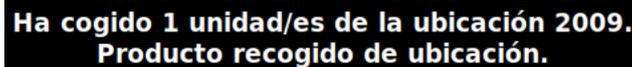
Ambos se encuentran sin inicializar (comenzado con valor a 0), y se crearon en la misma fecha, por lo que, en este caso, se elige el primero de ellos, el pedido con identificador 1. Elegimos la opción de simular pedido y se nos mostrarán los detalles de este, pudiendo observar el producto del que se trata, la cantidad a recoger, y la ubicación fija de la cual deberemos coger esta cantidad.



Simulador de pedidos
Producto - Ubicacion - Cantidad
 tornillos - 2009 - 1
 helados - 2006 - 3
 pan - 1001 - 6
 verdura - 1005 - 4
 galletas - 2002 - 2

Figura 4.23: Pantalla simular pedido con éxito, nos muestra el pedido.

Bajo el contenido del pedido, tenemos una ventana donde elegiremos la ubicación de las recomendadas a la que iremos primero. En este caso nos podemos encontrar con dos posibles situaciones, que la ubicación contenga la cantidad que nos solicita y, por lo tanto, se nos permite sin problemas completar la recogida de ese producto, donde se nos mostrará un mensaje de éxito, y por otro lado que no se encuentre en la ubicación la cantidad total que necesitamos, donde se nos indicará la cantidad recogida, el fin de existencias en esa ubicación, y un mensaje solicitando al usuario que ubique nueva cantidad de ese producto en su ubicación fija para poder continuar. Volviendo a los dos pedidos generados anteriormente, vamos a elegir la ubicación 2009, donde sí tenemos una unidad para recoger por lo que no debería haber problema.



Ha cogido 1 unidad/es de la ubicación 2009.
Producto recogido de ubicación.

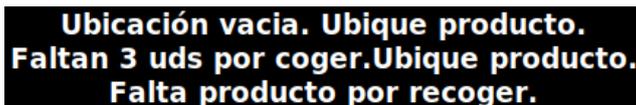
Figura 4.24: Pantalla simular pedido cogiendo un producto.

En este momento nos indica que se ha realizado con éxito y se descuenta esa ubicación en productos pendientes de recoger. En ese momento, se indica en la base de datos que ese pedido está iniciado ya, por lo que al volver a entrar en el sistema en el futuro se elegirá el primero hasta que sea terminado.

id_pedido	pedido	comenzado	pendiente	fecha_creado
1	{21: 1, 16: 3, 1: 6, 5: 4, 12: 2}	1	{16: 3, 1: 6, 5: 4, 12: 2}	2022-08-24
2	{12: 1, 7: 3, 5: 1}	0	{12: 1, 7: 3, 5: 1}	2022-08-24

Figura 4.25: Vista pedidos en base de datos con pedido ya comenzado.

La base de datos también guarda la parte pendiente aún por recoger, para poder continuar con ello al volver a elegirlo. La otra opción que tenemos es que no haya cantidad en la ubicación y el propio sistema nos pida que la repongamos.

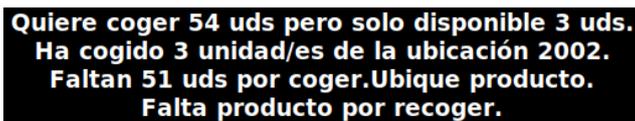


Ubicación vacía. Ubique producto.
Faltan 3 uds por coger. Ubique producto.
Falta producto por recoger.

Figura 4.26: Pantalla simular pedido con ubicación vacía.

La tercera opción que podríamos encontrarnos es gastar todo el producto de una ubicación y seguirnos faltando, donde el mensaje sería el siguiente.

Será extraño encontrarse con pedidos con cantidades tan grandes, pero la aplicación responde a



Quiere coger 54 uds pero solo disponible 3 uds.
Ha cogido 3 unidad/es de la ubicación 2002.
Faltan 51 uds por coger. Ubique producto.
Falta producto por recoger.

Figura 4.27: Pantalla simular pedido esperando más producto en ubicación fija.

todas estas situaciones y mantiene la cantidad restante por coger en todo momento.

- **Pantalla Limpiar Base de Datos de Pedidos Pendientes e Historial.**

La última opción de la que disponemos en el menú principal es la de limpiar la base de datos de los pedidos pendientes. Esta solo puede ejecutarse como administrador, y se encarga de hacer una limpieza general de las bases de datos tanto de pedidos pendientes, estén inicializados o no, como de pedidos terminados. Limpia ambas como si nunca se hubiese realizado uno, y eliminando también la parte de los pedidos que se hayan completado. Es poco recomendable su uso, pero disponemos de ella para poder visualizar pruebas en ejecuciones, o por si todos los pedidos han sido completados y se han almacenado en otro lugar más fiable, reiniciar nuestras bases de datos para mover menos cantidades de datos en el día a día.

4.2.10 Pantalla Iniciando Sesión como Usuario

Estas eran las acciones que el administrador podía realizar desde la interfaz, abarcando prácticamente la totalidad de la funcionalidad. Sin embargo, al iniciar sesión como usuario podemos apreciar un menú inicial distinto, no con funcionalidad nueva pero sí con un menor número de funciones, teniendo disponibles aquellas que no puedan causar un gran problema en el sistema ya que el usuario promedio del sistema no va a tener por qué conocer las acciones más avanzadas de creación de pedidos, historial de datos, etc. Un usuario cualquiera del sistema va a tener una única misión al utilizar la interfaz y esta va a ser realizar pedidos, por lo tanto la opción principal que este va a poseer va a ser la de simular pedidos. En este proceso se puede encontrar con un único inconveniente que va a ser la falta de stock en la ubicación fija del producto solicitado. Para ello, al usuario se le van a proporcionar las herramientas fundamentales para solventarlo y poder completar su trabajo sin parones ni ayudas de técnicos ni administradores.

Las opciones de simular pedidos y de sugerir reposiciones funcionan exactamente igual que en la interfaz del administrador vista previamente. La opción de ver reposición fija también es igual que la vista previamente para el administrador, pero en este caso sí es importante comentar la importancia de esta opción para los usuarios en detalle. En la introducción a este proyecto hablamos de la importancia de dominar el mercado, de buscar mejorar la velocidad, el servicio y el producto, por lo que tratar de seguir mejorando como empresa continuamente y ser cada vez más efectivos. Hemos recalcado varias veces que el objetivo principal de los usuarios del sistema en un almacén es realizar pedidos, los máximos posibles y de manera correcta, y esto nos hace pensar que solo existen los encargados de preparar esos pedidos. Sin embargo, suelen existir en estos lugares personas encargadas de colocar mercancía continuamente, de facilitar el trabajo de los que realizan los pedidos teniendo siempre material en el suelo para que tengan que realizar el menor número de pausas

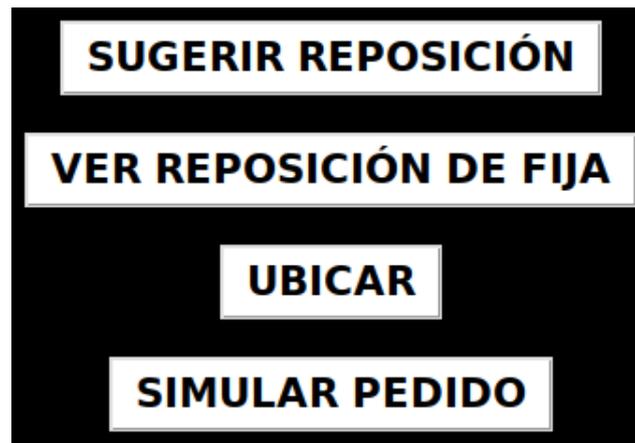


Figura 4.28: Pantalla menú como usuario.

posibles, observando las ubicaciones donde la cantidad de producto es escasa para tratar de prever la reposición en ellas, y cuyo objetivo es continuamente ese, proporcionar material a otros compañeros que van a retirarlo de manera continua.

La opción de sugerir reposición tiene la peculiaridad de mostrarte las ubicaciones fijas que tienen la cantidad de producto cero, es decir, que no poseen nada de material, para poder reemplazarlo por otra ubicación con contenido. Esta opción se usa cuando el usuario gasta la totalidad de la ubicación en su pedido y necesita más para continuar. Aquí es donde destaca la opción ver reposición fija. El encargado de reponer material aún no terminado puede elegir la ubicación fija por la que esté circulando y vea que queda poca cantidad o que muchos usuarios van a coger material de ella, pudiendo anticiparse a faltas de stock en el suelo y haciendo que no es “estancuen” varios usuarios esperando a este reabastecimiento.

La única opción diferente que dispone el usuario será la acción de ubicar. Este no va a tener la opción de elegir una ubicación origen y otra destino para ubicar, indiferentemente de su contenido, tipo de producto o fecha. En este caso al usuario se le solicitará solo la ubicación destino donde quiere ubicar material, y el sistema será el encargado de ofrecerle la lista de ubicaciones que contienen ese mismo producto.



Figura 4.29: Pantalla sugerir reposición al usuario con éxito.

En caso de elegir una ubicación de las que no han sido proporcionadas, nos mostrará un error y no nos dejará continuar con el movimiento.

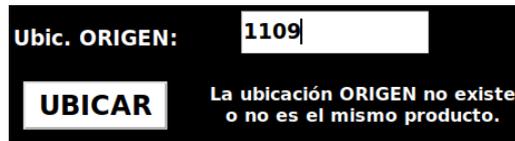


Figura 4.30: Pantalla sugerir reposición al usuario con error.

Si elegimos una opción válida, se completa el cambio de ubicación y nos muestra un mensaje como completo.



Figura 4.31: Pantalla sugerir reposición al usuario completo.

Esto sería todo lo que la interfaz de usuario puede ofrecernos o permitirnos hacer, y con esto finalizamos la descripción del sistema completa, habiendo visto las relaciones entre tablas de la base de datos, los datos que estas almacenan, los datos que se proporcionan a la interfaz o como incluye otros nuevos, y las acciones que los distintos roles pueden ejecutar.

Capítulo 5

Casos de Uso

A partir de los requisitos establecidos y tras un análisis en detalle de los mismos se han establecido los siguientes casos de uso que permiten abarcar toda la funcionalidad de la aplicación.

5.1 CU-01 - Iniciar Sesión

- **Objetivo:** Este caso de uso describe como un usuario se identifica en el sistema para comenzar a utilizar la interfaz.
- **Inicio:** La interfaz se encontrará en la pantalla inicial donde se esperará que se introduzca un nombre de usuario y una contraseña.
- **Flujo principal:**
 - **Actor:** El usuario introduce su usuario y contraseña y pulsa el botón “Iniciar sesión”.
 - **Aplicación:** Comprueba que existe el usuario y que coincide la contraseña en el sistema, y le muestra el menú correspondiente según el rol al que pertenezca.
- **Flujos excepcionales:** El usuario introduce un nombre de usuario o contraseña errónea y la interfaz te lo notifica para que pruebes de nuevo, sin límites de intentos.



Contraseña o usuario ERRÓNEO.

Figura 5.1: Contraseña o usuario erróneo.

- **Precondiciones:** Es necesario que se introduzca un nombre de usuario y contraseña que se encuentren registrados en el sistema previamente.
- **Finalización:** La aplicación se encontrará con una sesión iniciada y se nos mostrará el menú principal correspondiente al rol que tenga asignado el usuario.

En primer lugar vamos a tratar los casos de uso en el caso de haber iniciado sesión como administrador del sistema.

5.2 CU-02 - Limpiar BBDD Pedidos Pendientes/Historial

- **Objetivo:**

El sistema eliminará todo aquel pedido que se encuentre como pendientes o terminado, es decir, vaciará las tablas de pedidos pendientes e historial de la base de datos.

- **Inicio:**

Se trata de un botón que se debe pulsar en el menú principal, por lo que nos encontraremos en este hasta que se accione.

- **Flujo principal:**

- **Actor:**

El usuario deberá pulsar el botón con el nombre “Limpiar BBDD pedidos”.

- **Aplicación:**

El sistema eliminará los datos de las tablas pedidos pendientes e historial en caso de no tener problema con la conexión a la base de datos.

- **Flujos excepcionales:**

El único inconveniente que podría aparecer es que el sistema no se pudiese conectar a la base de datos pero no se ha contemplado ningún mensaje que informe de este error ya que no se nos dejaría iniciar sesión previamente.

- **Precondiciones:**

Encontrarnos en el menú principal del administrador ya que es donde se encuentra este botón.

- **Finalización:**

Se eliminarán tanto los pedidos finalizados almacenados en historial como los pendientes por hacer, estén o no comenzados por algún usuario.

5.3 CU-03 - Sugerir Reposición

- **Objetivo:**

Abrir la ventana donde se nos mostrarán las sugerencias de reposición de los productos del almacén para sus ubicaciones fijas.

- **Inicio:**

Se trata de un botón que se debe pulsar en el menú principal, por lo que nos encontraremos en este hasta que se accione.

- **Flujo principal:**

- **Actor:**

El usuario deberá pulsar el botón con el nombre “Sugerir reposición”.

- **Aplicación:**

La interfaz abrirá la nueva ventana donde aparecerán las sugerencias para la reposición de ubicaciones fijas. En caso de que el sistema detecte alguna ubicación fija vacía, nos lo mostrará con un mensaje tipo:



```
Necesitan reubicar: 1008
Ubicación fija: 1008 - Nombre: PAPEL - Id: 8
ubicacion_real - cantidad - fecha
1108 - 4 - 2020-12-29
1208 - 5 - 2021-01-03
```

Figura 5.2: Sugiere la reposición con éxito.

- **Flujos excepcionales:**

El único inconveniente que podría aparecer es que el sistema no se pudiese conectar a la base de datos pero no se ha contemplado ningún mensaje que informe de este error ya que no se nos dejaría iniciar sesión previamente.

- **Precondiciones:**

Encontrarnos en el menú principal ya que es donde se encuentra este botón.

- **Finalización:**

Se nos mostraría alguna de las dos opciones vistas en el punto “Flujo principal - Aplicación”, y se nos permitiría volver al menú principal con el botón cerrar.

5.4 CU-04 - Abrir Ventana Ubicar

- **Objetivo:**

Se trata de abrir la ventana de la interfaz donde se nos permita ubicar un pallet desde una ubicación origen hasta una destino.

- **Inicio:**

Se trata del botón “Ubicar” que se debe pulsar en el menú principal para poder abrir esta ventana.

- **Flujo principal:**

- **Actor:**

El usuario deberá pulsar el botón con el nombre “Ubicar”.

- **Aplicación:**

La interfaz abrirá una nueva ventana donde se solicitará al usuario los nuevos datos de entrada.

- **Flujos excepcionales:**

El único inconveniente que podría aparecer es que el sistema no se pudiese conectar a la base de datos pero no se ha contemplado ningún mensaje que informe de este error ya que no se nos dejaría iniciar sesión previamente.

- **Precondiciones:**

Encontrarnos en el menú principal ya que es donde se encuentra este botón.

- **Finalización:**

Se nos abrirá la nueva ventana de la interfaz, donde se solicitará al usuario que introduzca una ubicación origen del sistema y una destino para realizar el movimiento en el sistema que veremos a continuación.

5.5 CU-04-1 - Ubicar

- **Objetivo:**

Se trata de ubicar en el sistema la información de la ubicación que introducimos como origen en la ubicación destino.

- **Inicio:**

Nos encontramos en el menú ubicar por lo que dispondremos de dos entradas de datos, la ubicación origen y destino, y un botón para realizar el cambio.

- **Flujo principal:**

- **Actor:**

Introduce la ubicación origen y destino en su entrada correspondiente, y pulsa el botón para realizar el cambio.

- **Aplicación:**

Si el cambio no se encuentra con ningún conflicto de datos, la interfaz nos mostrará un mensaje de éxito.

- **Flujos excepcionales:**

Pueden darse situaciones donde no se podrá realizar el movimiento y se nos mostrará un mensaje de error. Algunas de ellas sería que alguna de las dos ubicaciones no exista en el sistema, que la ubicación origen no contenga producto, que la destino si lo contenga y no se puedan juntar...

- **Precondiciones:**

Encontrarnos dentro de la ventana ubicar a la que habremos accedido desde el menú principal y rellenar los datos que se nos solicita.

- **Finalización:**

El sistema moverá los datos de la ubicación origen a la destino, dejando vacía esta primera y lista para ser usada por otro producto.

5.6 CU-05 - Abrir Ventana Ver Reposición de Fija

- **Objetivo:**

Se trata de abrir la ventana de la interfaz donde se nos permita ver la reposición de una ubicación fija.

- **Inicio:**

Se trata del botón “Ver reposición de fija” que se debe pulsar en el menú principal para poder abrir esta ventana.

- **Flujo principal:**

- **Actor:**

El usuario deberá pulsar el botón con el nombre “Ver reposición de fija”.

- **Aplicación:**

La interfaz abrirá una nueva ventana donde se solicitará al usuario los nuevos datos de entrada.

- **Flujos excepcionales:**

El único inconveniente que podría aparecer es que el sistema no se pudiese conectar a la base de datos pero no se ha contemplado ningún mensaje que informe de este error ya que no se nos dejaría iniciar sesión previamente.

- **Precondiciones:**

Encontrarnos en el menú principal ya que es donde se encuentra este botón.

- **Finalización:**

Se nos abrirá la nueva ventana de la interfaz, donde se solicitará al usuario que introduzca una ubicación fija registrada en el sistema para mostrarle posibles ubicaciones que contengan este producto.

5.7 CU-05-1 - Ver Reposición de Fija

- **Objetivo:**

Se trata de, introduciendo la ubicación fija del producto que deseamos saber sus ubicaciones, recibir una lista con todas las ubicaciones donde se encuentra un producto determinado en el almacén.

- **Inicio:**

Nos encontramos en el menú “ver reposición de fija” por lo que dispondremos de una entrada de datos para la ubicación fija y un botón “ver” para recibir la lista de ubicaciones.

- **Flujo principal:**

- **Actor:**

Introduce la ubicación fija en su entrada correspondiente, y pulsa el botón para ver el listado de ubicaciones.

- **Aplicación:**

Si la ubicación fija está registrada en el sistema relacionada a un producto, se mostrará el listado de las ubicaciones que contienen dicho producto en el almacén. En caso de no quedar stock de este, se indicará.

- **Flujos excepcionales:**

Puede darse la situación donde el dato sea erróneo, como una entrada no válida, o que no corresponda a ningún producto como ubicación fija, y aquí se mostrará un mensaje de error por pantalla.

- **Precondiciones:**

Encontrarnos dentro de la ventana ver reposición de fija a la que habremos accedido desde el menú principal y rellenar los datos que se nos solicita.

- **Finalización:**

El sistema nos mostrará el listado de ubicaciones que contengan el mismo producto al que corresponde la ubicación fija introducida como dato.

5.8 CU-06 - Abrir Ventana Introducir Nuevo Producto

- **Objetivo:**

Se trata de abrir la ventana de la interfaz donde se nos permite introducir un nuevo producto a la base de datos de nuestro sistema.

- **Inicio:**

Se trata del botón “Introducir nuevo producto” que se debe pulsar en el menú principal para poder abrir esta ventana.

- **Flujo principal:**

- **Actor:**

El usuario deberá pulsar el botón con el nombre “Introducir nuevo producto”.

- **Aplicación:**

La interfaz abrirá una nueva ventana donde se solicitarán los datos del nuevo producto a añadir en la base de datos.

- **Flujos excepcionales:**

El único inconveniente que podría aparecer es que el sistema no se pudiese conectar a la base de datos pero no se ha contemplado ningún mensaje que informe de este error ya que no se nos dejaría iniciar sesión previamente.

- **Precondiciones:**

Encontrarnos en el menú principal ya que es donde se encuentra este botón.

- **Finalización:**

Se nos abrirá la nueva ventana de la interfaz, donde se solicitarán al usuario los datos del nuevo producto a introducir en la base de datos para que el almacén pueda almacenar cantidad controlada de este producto.

5.9 CU-06-1 - Introducir Nuevo Producto

- **Objetivo:**

Se trata de introducir un nuevo producto a la base de datos del sistema, de manera que el almacén pueda almacenar cantidad de este producto posteriormente conociendo ya sus datos previos, como la ubicación fija.

- **Inicio:**

Nos encontramos en el menú “introducir nuevo producto” por lo que dispondremos de una entrada de datos para el nombre del producto y para su ubicación fija deseada, eligiendo cero si se quiere que el sistema la seleccione automáticamente.

- **Flujo principal:**

- **Actor:**

Introduce el nombre del producto y la ubicación fija del producto que se desea dar de alta (cero si se quiere que la calcule el sistema la siguiente disponible), y se pulsa el botón “introducir”.

- **Aplicación:**

Si la ubicación fija está disponible sin otro producto ya asignado a esta, se da de alta el producto en la base de datos y queda relacionado con esa ubicación.

- **Flujos excepcionales:**

Puede darse la situación donde el dato sea erróneo, como una ubicación fija no válida, que ya esté utilizada por otro producto, que no se encuentre en el suelo, o fuera de los rangos del sistema. En tal caso se notificará con un mensaje por pantalla.

- **Precondiciones:**

Encontrarnos dentro de la ventana introducir nuevo producto a la que habremos accedido desde el menú principal y rellenar los datos que se nos solicita.

- **Finalización:**

El sistema nos mostrará un mensaje de haberse completado el proceso con éxito y ya podremos regresar al menú principal.

5.10 CU-07 - Abrir Ventana Dar Entrada

- **Objetivo:**

Se trata de abrir la ventana de la interfaz donde se nos permite dar entrada a una cantidad al almacén de un producto registrado previamente en la base de datos.

- **Inicio:**

Se trata del botón “Dar entrada” que se debe pulsar en el menú principal para poder abrir esta ventana.

- **Flujo principal:**

- **Actor:**

El usuario deberá pulsar el botón con el nombre “Dar entrada”.

- **Aplicación:**

La interfaz abrirá una nueva ventana donde se solicitarán los datos del nuevo producto a añadir en el almacén.

- **Flujos excepcionales:**

El único inconveniente que podría aparecer es que el sistema no se pudiese conectar a la base de datos pero no se ha contemplado ningún mensaje que informe de este error ya que no se nos dejaría iniciar sesión previamente.

- **Precondiciones:**

Encontrarnos en el menú principal ya que es donde se encuentra este botón.

- **Finalización:**

Se nos abrirá la nueva ventana de la interfaz, donde se solicitarán al usuario los datos del producto a dar entrada junto a su cantidad y la ubicación física en la que quedará colocada esa mercancía.

5.11 CU-07-1 - Dar Entrada

- **Objetivo:**

Se trata de dar de alta una cantidad al almacén de un producto ya registrado previamente en la base de datos y el cual tendrá asignado una ubicación fija previamente.

- **Inicio:**

Nos encontramos en el menú “dar entrada” por lo que dispondremos de una entrada de datos para la ubicación física, el identificador del producto (por el cual se conocerá la ubicación fija del producto), y la cantidad a dar entrada en el almacén.

- **Flujo principal:**

- **Actor:**

Introduce la ubicación donde se va a colocar la mercancía, el identificador del producto que se trata, y la cantidad de este que contiene.

- **Aplicación:**

Si la ubicación física se encuentra libre en el sistema, y el identificador del producto es válido, se da de entrada la cantidad asignada en los datos, sea dicha cantidad cualquier número entero válido.

- **Flujos excepcionales:**

Puede darse la situación donde el dato sea erróneo, como una ubicación real que no exista o que se encuentre ocupada por otra mercancía, o un identificador que no se encuentre registrado en el sistema, o una cantidad introducida que no sea un número entero. Sea cual sea el problema, se notificará al usuario por pantalla.

- **Precondiciones:**

Encontrarnos dentro de la ventana dar entrada a la que habremos accedido desde el menú principal y rellenar los datos que se nos solicita.

- **Finalización:**

El sistema nos mostrará un mensaje de haberse completado el proceso con éxito y ya podremos regresar al menú principal.

5.12 CU-08 - Actualizar Control de Stock

- **Objetivo:**

Con esta acción el sistema no abre ninguna ventana, simplemente actualiza las bases de datos. Se busca actualizar la tabla que registra el control del stock en cada momento.

- **Inicio:**

Se trata del botón “Actualizar control de stock” que se debe pulsar en el menú principal para ejecutarse, sin abrir ventanas secundarias.

- **Flujo principal:**

- **Actor:**

El usuario deberá pulsar el botón con el nombre “Actualizar control de stock”.

- **Aplicación:**

La interfaz actualizará la tabla de la base de datos encargada con el control del stock en tiempo real. Para cada producto que se encuentre en el almacén, esta tabla calculará la cantidad total física que disponemos de este en el almacén, la cantidad que se encuentra reservada para pedidos pendientes de realizar, y la diferencia de estos, que haría referencia a la cantidad disponible que poseemos para que otros pedidos nuevos puedan solicitar sin faltarles stock.

- **Flujos excepcionales:**

El único inconveniente que podría aparecer es que el sistema no se pudiese conectar a la base de datos pero no se ha contemplado ningún mensaje que informe de este error ya que no se nos dejaría iniciar sesión previamente.

- **Precondiciones:**

Encontrarnos en el menú principal ya que es donde se encuentra este botón.

- **Finalización:**

No habrá ningún mensaje ni de error ni de éxito ya que es un botón creado para que un administrador, al realizar pruebas manuales como eliminar o cambiar algún valor directamente en la base de datos, pueda refrescar esta tabla sin problemas. Esta acción se ejecuta de manera automática cada vez que se realiza un movimiento en el sistema, por lo que es un botón sin apenas uso exceptuando el caso anterior mencionado de simples pruebas o cambios puntuales.

5.13 CU-09 - Abrir Ventana Generador de Pedidos

- **Objetivo:**

Se trata de abrir la ventana de la interfaz donde se nos permite generar pedidos en el sistema y almacenarlos como pendientes de hacer.

- **Inicio:**

Se trata del botón “Generador de pedidos” que se debe pulsar en el menú principal para poder abrir esta ventana.

- **Flujo principal:**

- **Actor:**

El usuario deberá pulsar el botón con el nombre “Generador de pedidos”.

- **Aplicación:**

La interfaz abrirá una nueva ventana donde se solicitará el tipo de pedido que se quiere generar y un botón para realizar la acción.

- **Flujos excepcionales:**

El único inconveniente que podría aparecer es que el sistema no se pudiese conectar a la base de datos pero no se ha contemplado ningún mensaje que informe de este error ya que no se nos dejaría iniciar sesión previamente.

- **Precondiciones:**

Encontrarnos en el menú principal ya que es donde se encuentra este botón.

- **Finalización:**

Se nos abrirá la nueva ventana de la interfaz, donde se solicitarán al usuario el tipo de pedido que se quiere generar y un botón para crearlo, mostrándolo posteriormente por pantalla.

5.14 CU-09-1 - Generador de Pedidos

- **Objetivo:**

Se trata de generar un pedido y almacenarlo en la tabla de la base de datos encargada de recoger los pedidos pendientes a realizar.

- **Inicio:**

Nos encontramos en el menú “generador de pedidos” por lo que dispondremos de una entrada de datos para el tipo de pedido que queremos generar, y un botón para aprobar la acción.

- **Flujo principal:**

- **Actor:**

Introduce un valor entre 1 y 3 siendo uno un pedido pequeño, el dos un pedido mediano y el tres un pedido totalmente aleatorio, pudiendo ser de cualquier tamaño. Tras esta elección se pulsa el botón “Generar” y se muestra por pantalla el pedido creado.

- **Aplicación:**

Si el número que hace referencia al tipo de pedido es válido y queda mercancía en el almacén, se generará el pedido y se mostrará por pantalla.

- **Flujos excepcionales:**

Puede darse la situación donde el dato sea erróneo, como un valor fuera del rango solicitado, donde se mostrará el mensaje de error por pantalla. También puede darse que no quede material en el almacén, donde también se notificará por pantalla de este suceso.

- **Precondiciones:**

Encontrarnos dentro de la ventana del generador de pedidos a la que habremos accedido desde el menú principal y rellenar los datos que se nos solicita.

- **Finalización:**

El sistema nos mostrará un mensaje de haberse completado el proceso con éxito junto con el pedido creado, mostrando el identificador del producto y seguido de él la cantidad a coger.

5.15 CU-10 - Abrir Ventana Simular Pedido

- **Objetivo:**

Se trata de abrir la ventana de la interfaz donde se nos permite simular pedidos generados previamente.

- **Inicio:**

Se trata del botón “Simular pedido” que se debe pulsar en el menú principal para poder abrir esta ventana.

- **Flujo principal:**

- **Actor:**

El usuario deberá pulsar el botón con el nombre “Simular pedido”.

- **Aplicación:**

La interfaz abrirá una nueva ventana donde se mostrará el pedido a simular en caso de haber alguno pendiente, y una entrada de datos para elegir la ubicación en la que se encuentra.

- **Flujos excepcionales:**

El único inconveniente que podría aparecer es que el sistema no se pudiese conectar a la base de datos pero no se ha contemplado ningún mensaje que informe de este error ya que no se nos dejaría iniciar sesión previamente.

- **Precondiciones:**

Encontrarnos en el menú principal ya que es donde se encuentra este botón.

- **Finalización:**

Se nos abrirá la nueva ventana de la interfaz, donde el usuario irá indicando las ubicaciones que simula coger y se irá mostrando por pantalla lo que queda de pedido por realizar.

5.16 CU-10-1 - Simular Pedido

- **Objetivo:**

Se trata de simular un pedido previamente generado, de manera que vayamos indicando que ubicaciones de las que nos indica recoger vamos eligiendo, y nos muestra en todo momento el pedido pendiente restante.

- **Inicio:**

Nos encontramos en el menú “simular pedidos” por lo que dispondremos de una entrada de datos para la ubicación que vamos a ir recorriendo de las que quedan pendientes por simular en el pedido y un botón “simular” para aceptar esa ubicación.

- **Flujo principal:**

- **Actor:**

Introduce una ubicación válida de las que el sistema nos pide para ese pedido, y según vamos recorriendo ubicaciones, la interfaz nos muestra el resto de pedido pendiente hasta finalizarlo.

- **Aplicación:**

Si las ubicaciones introducidas son válidas, se van descontando las cantidades como pendiente hasta completar el pedido. Si falta cantidad en una ubicación, debemos retroceder, realizar el movimiento de reposición, y seguir cogiendo hasta completarlo.

- **Flujos excepcionales:**

Puede darse la situación donde el dato sea erróneo, como que la ubicación introducida no exista o no la solicite el pedido, o el valor de entrada no sea una ubicación. De cualquiera de estas formas, la interfaz nos mostrará un mensaje por pantalla indicándonos que no se ha podido realizar la acción.

- **Precondiciones:**

Encontrarnos dentro de la ventana del simulador de pedidos a la que habremos accedido desde el menú principal y rellenar los datos que se nos solicita.

- **Finalización:**

El sistema nos irá mostrando la parte del pedido que nos queda por recoger en todo momento hasta completarlo. Llegado ese momento, se nos indicará que el pedido se ha realizado por completo con un mensaje de éxito.

Si entramos a la interfaz como usuario en lugar de como administrador, nos encontraremos con opciones iguales exceptuando la de ubicar. Vamos a explicar la diferencia con la manera de ubicar una mercancía de un usuario.

5.17 CU-11 - Abrir Ventana Ubicar Como Usuario

- **Objetivo:**

Se trata de abrir la ventana de la interfaz donde se nos permite ubicar un pallet de una ubicación del sistema en otra que se encuentre vacía.

- **Inicio:**

Se trata del botón “Ubicar” que se debe pulsar en el menú principal para poder abrir esta ventana.

- **Flujo principal:**

- **Actor:**

El usuario deberá pulsar el botón con el nombre “Ubicar”.

- **Aplicación:**

La interfaz abrirá una nueva ventana donde se solicitará la ubicación destino donde se quiere ubicar la mercancía y un botón “Seguir”.

- **Flujos excepcionales:**

El único inconveniente que podría aparecer es que el sistema no se pudiese conectar a la base de datos pero no se ha contemplado ningún mensaje que informe de este error ya que no se nos dejaría iniciar sesión previamente.

- **Precondiciones:**

Encontrarnos en el menú principal ya que es donde se encuentra este botón.

- **Finalización:**

Se nos abrirá la nueva ventana de la interfaz, donde el usuario seguirá los pasos que la pantalla le vaya indicando para completar la ubicación de mercancía.

5.18 CU-11-1 - Ubicar Como Usuario

- **Objetivo:**

Se trata de ubicar la mercancía de una ubicación origen en otra destino, indicadas como entrada de datos.

- **Inicio:**

La diferencia con la opción del administrador es que los usuarios se encontrarán sólo una entrada de datos, la de ubicación destino. Nos encontramos en el menú “ubicar” por lo que dispondremos de esta entrada de datos , junto a un botón seguir que nos indicará los pasos siguientes.

- **Flujo principal:**

- **Actor:**

Introduce una ubicación destino válida de las que el sistema tiene registradas, pulsa el botón “seguir”, y se le mostrará una lista de ubicaciones que contienen ese producto y de las cuales tiene que elegir. En este caso elige la ubicación origen de la cual se va a mover la mercancía y se vuelve a pulsar “seguir”.

- **Aplicación:**

Si las ubicaciones introducidas son válidas, tanto en primer lugar la de destino como después la de origen corresponde al mismo producto, el sistema completa el movimiento y quedaría la mercancía ubicada en su nuevo lugar.

- **Flujos excepcionales:**

Puede darse la situación donde el dato sea erróneo, como que la ubicación destino tenga mercancía, la ubicación origen no tenga nada, o alguna de ellas no exista en el sistema. Para estos casos se mostrará por pantalla un mensaje de error indicando la causa a la que corresponde.

- **Precondiciones:**

Encontrarnos dentro de la ventana de ubicar a la que habremos accedido desde el menú principal y rellenar los datos que se nos solicita.

- **Finalización:**

El sistema nos indicará que el movimiento se ha realizado con éxito y podremos volver al menú a seguir con el pedido o continuar ubicando otras mercancías.

Capítulo 6

Conclusiones y líneas futuras

6.1 Conclusiones

Al comienzo del proyecto se establecieron algunos objetivos y se va a evaluar ahora si han sido alcanzados y de que manera:

- Los primeros objetivos fueron de la parte técnica. Se buscaba obtener técnicas de programación con peticiones SQL, y se ha superado con creces, ya que todos los movimientos realizados en el sistema se han resuelto mediante estas. Fuese la entrada de nuevos productos, como cambiando ubicaciones, hasta llegar a la parte de descontar un producto de una ubicación al realizar un pedido, todo ha tenido que realizarse con este tipo de peticiones por lo que esta parte del objetivo podemos darlo por superado. La implementación de estas peticiones ha sido llevada a cabo bajo el lenguaje Python, y juntando esto con la librería con la que hemos creado la interfaz, hemos conseguido la unión que el sistema necesitaba. Se ha podido asignar a cada botón o a cada entrada de datos una función vinculada directamente a una petición a la base de datos, y así agilizar el funcionamiento de esta.
- Otro objetivo era otorgar una lógica al almacén donde consiguieras tiempos de respuesta rápidos, pensando siempre en los usuarios del sistema y en la parte de moverse por el almacén, y en cómo optimizar su trabajo. La repartición del almacén, el tamaño de este, las alturas otorgadas y las distancias de los pasillos es un tema que ya se explicó previamente, pero si cabe destacar en este apartado alguna de las decisiones tomadas más personales como la asignación a cada producto de una ubicación fija. Con esta decisión se buscó, en primer lugar, familiarizar a los usuarios del sistema a encontrar un mismo producto en una misma ubicación siempre, agilizando los pedidos y facilitándolos la organización a la hora de moverse por el almacén, y en segundo lugar, al sistema le facilita las cosas a la hora de filtrar, buscar ubicaciones donde reponer material, y agilizar búsquedas de información.
- Por último teníamos como objetivo agregar mejoras que pudieran ir surgiendo durante el desarrollo y que, sobre todo, nos fueran útiles. Algunas que se han implementado a posteriori han sido los botones de limpiar base de datos de historial y pedidos pendientes, para poder realizar pruebas de pedidos grandes y eliminarlos rápidamente para que no causen conflictos en

el almacén. Otro de ellos podría ser el botón de actualizar el control del stock, ya que, aunque se realiza de manera automática tras cada ejecución de la interfaz, se puede desear usarlo al ejecutar alguna petición a la base de datos de manera directa sin el uso de la aplicación.

6.2 Líneas Futuras

Como pasos futuros que se podrían realizar en la aplicación podemos encontrar los siguientes:

- A la hora de iniciar sesión, la base de datos almacena las contraseñas en texto plano, como una cadena de texto normal, para que a la hora de realizar capturas, pruebas y de cara a la memoria fuese todo más visible y sencillo. Como mejora, se debería encriptar ese valor en la base de datos y que la petición a la base de datos tuviese un tipo específico de parámetros para tratar que la conexión fuese lo más privada y segura posible.
- Cuando elegimos la opción "sugerir reposición", nos muestra solo las ubicaciones fijas que no tienen nada de mercancía en ellas, y de esta manera, podría ocultarnos información valiosa como que en una de ellas queden solo una o dos unidades de producto, el cual necesitaría reposición pronto. Se podría implementar un número límite a partir del cual se comenzase a notificar como próximos movimientos.
- A la hora de ubicar un pallet de un lugar a otro, se solicita la ubicación de origen y destino para realizar este movimiento. A pesar de que se realizan todas las comprobaciones necesarias y que no se hace ningún movimiento indebido, podría implementarse una opción en la que al elegir un pallet solo te deje ubicarlo en ubicación fija, para evitar movimientos sospechosos entre ubicaciones de altura que no tienen producto en común.
- Como punto a mejorar o siguiente trabajo, se podría añadir el proceso en el que cada pedido tendrá un identificador único y a la hora de simular pedidos se pudiese elegir este. Esto sería una buena práctica en el momento que se tuviesen varios usuarios trabajando de manera simultánea y cada uno tuviese pedidos inicializados, o quisiera elegir otros por problemas de stock en suelo, u otros problemas ajenos al sistema. Así cada pedido tendría asignado también el usuario de quien lo comenzó, pudiendo ser solo este el encargado de continuarlo, pero sin la obligación de tener que seguir el último iniciado.

Capítulo 7

Bibliografía

- <https://github.com/mysql/mysql-server>
- <https://www.geeksforgeeks.org/difference-between-mysql-and-mariadb/>
- <https://www.techtarget.com/searchdatamanagement/definition/MariaDB>
- <https://www.geeksforgeeks.org/difference-between-mariadb-and-postgresql/?ref=rp>
- <https://www.ionos.com/digitalguide/hosting/technical-matters/mariadb-vs-mysql/>
- <https://dbeaver.com/docs/wiki/Application-Window-Overview/>
- <https://docs.python.org/es/3/library/tkinter.html>
- <https://realpython.com/python-gui-tkinter/controlling-layout-with-geometry-managers>
- <https://dbeaver.com/docs/wiki/SQL-Terminal/>
- <https://retos-operaciones-logistica.eae.es/gestion-de-almacenes-todo-lo-que-hay-que-saber/>

Universidad de Alcalá
Escuela Politécnica Superior



ESCUELA POLITECNICA
SUPERIOR



Universidad
de Alcalá