

TIDYFOOD

2º Ingeniería Informática 2021-2022

Trabajo Del Segundo Cuatrimestre

Gorka Etxebarria, Andoni Martí, Unai Sasiain y Sergio Villain

*10 de junio de 2022*

RESUMEN

En esta memoria se habla sobre la solución que la empresa TidyFood da a los problemas de la gestión de productos en los bancos de alimentos con el objetivo de reducir la cantidad de alimentos que son desechados principalmente en supermercados. Primero se hablará sobre el diseño del producto, después el funcionamiento de este, para continuar con la infraestructura de la empresa y finalizar explicando las conclusiones sacadas por la empresa TidyFood.

LABURPENA

Memoria honetan, TidyFood enpresak elikagai-bankuetan produktuak kudeatzeko arazoei ematen dien irtenbideari buruz hitz egiten da. Proiektu honen helburua, batez ere supermerkatuetan, botatzen diren elikagaien kopurua murriztea da. Lehenik eta behin, produktuaren diseinuari buruz hitz egingo da, ondoren haren funtzionamenduari buruz, enpresaren azpiegiturarekin jarraitzeko, eta, azkenik, TidyFood enpresak ateratako ondorioak azalduko dira.

ABSTRACT

This report talks about the solution that TidyFood gives to the problems of product management in food banks with the aim of reducing the amount of food that is disposed in supermarkets. First, this report will talk about the design of the product, then the performance of it, then it will be presented the infrastructure of the company and in the end the conclusions taken by the company TidyFood.

ÍNDICE

[1 INTRODUCCIÓN 1](#_Toc105796028)

[2 PLANTEAMIENTO DEL ÁMBITO 2](#_Toc105796029)

[2.1 ÁREA DE OPERACIONES TRADICIONAL 2](#_Toc105796030)

[2.1.1 RECIBO 2](#_Toc105796031)

[2.1.2 ALMACENAMIENTO 2](#_Toc105796032)

[2.1.3 ENTREGA 3](#_Toc105796033)

[3 PLANTEAMIENTO DEL PROBLEMA Y NUEVO FUNCIONAMIENTO 4](#_Toc105796034)

[3.1 RECIBO 4](#_Toc105796035)

[3.2 ALMACENAMIENTO 5](#_Toc105796036)

[3.3 ENTREGA 5](#_Toc105796037)

[4 DISEÑO DEL PRODUCTO 6](#_Toc105796038)

[5 PLANIFICACIÓN 8](#_Toc105796039)

[6 DESARROLLO DE LA APLICACIÓN 9](#_Toc105796040)

[7 FUNCIONAMIENTO DEL PRODUCTO 12](#_Toc105796041)

[8 RED DEL SISTEMA 16](#_Toc105796042)

[9 CONEXIÓN SERIAL 19](#_Toc105796043)

[10 REPERCUSIÓN DEL PRODUCTO 21](#_Toc105796044)

[11 CONCLUSIONES 25](#_Toc105796045)

[12 CONCLUSIONES PERSONALES 26](#_Toc105796046)

[13 LÍNEAS FUTURAS 28](#_Toc105796047)

[13.1 PLANTILLAS DE LOTES 28](#_Toc105796048)

[13.2 INTERFACES PARA LOS BENEFICIARIOS 28](#_Toc105796049)

[13.3 REPERCUSIÓN DEL PRODUCTO EN UNA SITUACIÓN REAL 28](#_Toc105796050)

[14 BIBLIOGRAFÍA 29](#_Toc105796051)

[15 ANEXO 1 - IDEAS INICIALES 30](#_Toc105796052)

[15.1 Introducción 30](#_Toc105796053)

[15.2 Lluvia de ideas 30](#_Toc105796054)

[15.2.1 Gestión de banco de alimentos (TidyFood) 30](#_Toc105796055)

[15.2.2 Gestión de vuelos 30](#_Toc105796056)

[15.2.3 Máquina expendedora 30](#_Toc105796057)

[15.2.4 Gestión de museos 30](#_Toc105796058)

[15.2.5 Gestión de residuos médicos 30](#_Toc105796059)

[15.3 Matriz de ideas 31](#_Toc105796060)

[15.4 Idea elegida 31](#_Toc105796061)

[15.4.1 Gestión de banco de alimentos (TidyFood) 31](#_Toc105796062)

[16 ANEXO 2 – ANEXOS ADS 32](#_Toc105796063)

[16.1 ANEXO 2.1 - ERS 32](#_Toc105796064)

[16.1.1 Introducción 32](#_Toc105796065)

[16.1.2 Descripción general 33](#_Toc105796066)

[16.1.3 Requisitos específicos 35](#_Toc105796067)

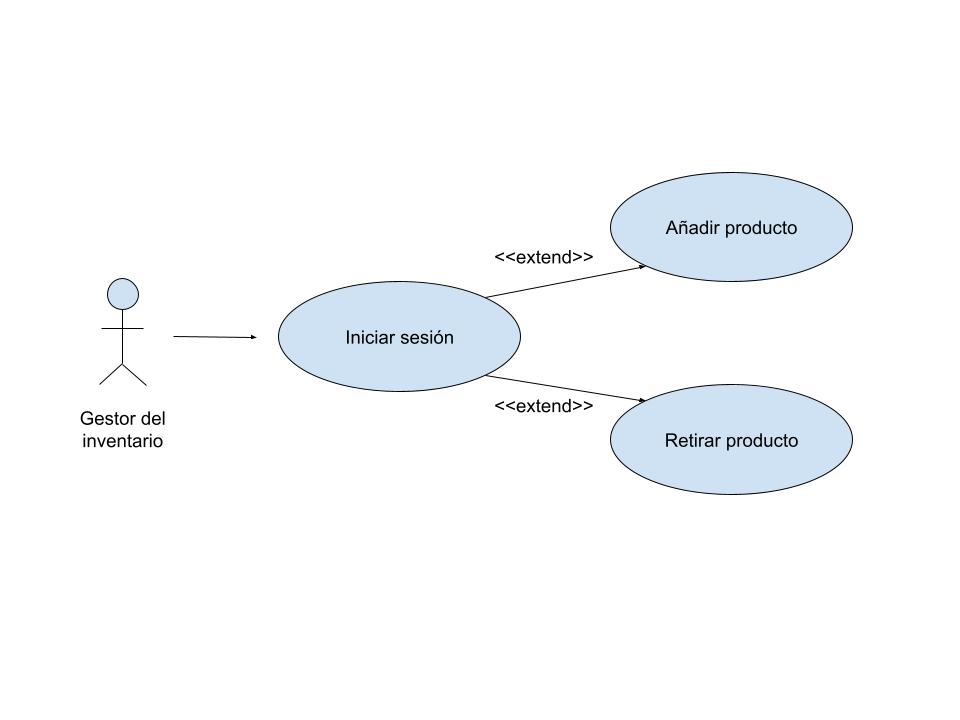
[16.2 ANEXO 2.2 - Casos de uso y escenarios 40](#_Toc105796068)

[16.2.1 Diagrama de casos de uso total 40](#_Toc105796069)

[16.2.2 Diagrama de casos de uso general 40](#_Toc105796070)

[16.2.3 Diagrama de casos de uso de recibos 41](#_Toc105796071)

[16.2.4 Diagrama de casos de uso de inventario 43](#_Toc105796072)

[ 44](#_Toc105796073)

[16.2.5 Diagrama de casos de uso de entregas 46](#_Toc105796074)

[16.2.6 Diagrama de casos de uso de gestión de usuarios 49](#_Toc105796075)

[16.3 ANEXO 2.3 - Diagrama de estados 52](#_Toc105796076)

[16.4 ANEXO 2.4 - Diagramas de actividad 54](#_Toc105796077)

[16.4.1 Inicio de sesión 54](#_Toc105796078)

[16.4.2 Registro de productos 55](#_Toc105796079)

[16.4.3 Creación de nuevos usuarios 56](#_Toc105796080)

[16.5 ANEXO 2.5 – Diagramas de clases 57](#_Toc105796081)

[16.5.1 Registro de productos 57](#_Toc105796082)

[16.5.2 Inicio de sesión 58](#_Toc105796083)

[16.5.3 Creación de usuarios 59](#_Toc105796084)

[17 ANEXO 3 - BASE DE DATOS 60](#_Toc105796085)

[17.1 Consultas MySQL 64](#_Toc105796086)

[17.1.1 Creación de tablas 64](#_Toc105796087)

[17.1.2 Motores de almacenamiento 65](#_Toc105796088)

[17.1.3 Usuarios 66](#_Toc105796089)

[17.2 Seguridad 66](#_Toc105796090)

[17.2.1 Replicación 66](#_Toc105796091)

[17.2.2 Copias de seguridad 67](#_Toc105796092)

[17.3 Conexión de la base de datos con la aplicación 67](#_Toc105796093)

[18 ANEXO 4 - REDES 70](#_Toc105796094)

[18.1 PAT 70](#_Toc105796095)

[18.2 OSPF MULTIAREA 70](#_Toc105796096)

[18.3 OSPF con autenticación 71](#_Toc105796097)

[18.4 PPP con autenticación 71](#_Toc105796098)

[18.5 SNMP server 72](#_Toc105796099)

[18.6 CDP 72](#_Toc105796100)

[18.7 NTP 72](#_Toc105796101)

[18.8 TFTP 72](#_Toc105796102)

[18.9 ACL extendidas 73](#_Toc105796103)

[19 ANEXO 5 - MICROPROCESADOR 74](#_Toc105796104)

[19.1 Función 74](#_Toc105796105)

[19.2 Tecnologías empleadas 74](#_Toc105796106)

[19.2.1 KeilUvision5 74](#_Toc105796107)

[19.2.2 Lenguaje C 75](#_Toc105796108)

[19.2.3 Protocolo UART 75](#_Toc105796109)

[19.3 Sistema de archivos 77](#_Toc105796110)

[19.4 Funcionalidades 77](#_Toc105796111)

[19.4.1 Inicialización de la comunicación 77](#_Toc105796112)

[19.4.2 Función de lectura de datos 78](#_Toc105796113)

[19.4.3 Función de visualización 78](#_Toc105796114)

[19.5 Conclusión 79](#_Toc105796115)

[20 ANEXO 6 – APLICACIÓN JAVA 80](#_Toc105796116)

[20.1 FRAMEWORKS UTILIZADOS 80](#_Toc105796117)

[20.1.1 SWING 80](#_Toc105796118)

[20.1.2 JSSC 80](#_Toc105796119)

[20.1.3 JAVA CONNECTOR MYSQL 80](#_Toc105796120)

[20.1.4 JFREECHART 80](#_Toc105796121)

[20.2 DISTRIBUCION DE FICHEROS 81](#_Toc105796122)

[20.3 FUNCIONAMIENTO DEL PRODUCTO 82](#_Toc105796123)

[20.4 FUNCIONALIDADES QUE DESTACAR 82](#_Toc105796124)

[20.4.1 FUNCIÓN MEDIANTE ESCALA 82](#_Toc105796125)

[20.4.2 GESTIÓN DE DATOS MEDIANTE LA BASE DE DATOS 83](#_Toc105796126)

[20.4.3 REGISTRO DE PRODUCTOS, PROVEEDORES Y USUARIOS 84](#_Toc105796127)

[20.4.4 INICIO DE SESION 85](#_Toc105796128)

ÍNDICE DE IMÁGENES

[Ilustración 1 Interfaz registro de donaciones 4](https://mgep-my.sharepoint.com/personal/unai_sasiain_alumni_mondragon_edu/Documents/1.-Memoria%20(1).docx#_Toc105795965)

[Ilustración 2 Botonera 6](https://mgep-my.sharepoint.com/personal/unai_sasiain_alumni_mondragon_edu/Documents/1.-Memoria%20(1).docx#_Toc105795966)

[Ilustración 3 Planificación 8](https://mgep-my.sharepoint.com/personal/unai_sasiain_alumni_mondragon_edu/Documents/1.-Memoria%20(1).docx#_Toc105795967)

[Ilustración 4 Boceto pantalla principal 9](https://mgep-my.sharepoint.com/personal/unai_sasiain_alumni_mondragon_edu/Documents/1.-Memoria%20(1).docx#_Toc105795968)

[Ilustración 5 Boceto pantalla creación de lotes 10](https://mgep-my.sharepoint.com/personal/unai_sasiain_alumni_mondragon_edu/Documents/1.-Memoria%20(1).docx#_Toc105795969)

[Ilustración 6 Boceto pantalla inventario 10](https://mgep-my.sharepoint.com/personal/unai_sasiain_alumni_mondragon_edu/Documents/1.-Memoria%20(1).docx#_Toc105795970)

[Ilustración 7 Pantalla Inicio 12](https://mgep-my.sharepoint.com/personal/unai_sasiain_alumni_mondragon_edu/Documents/1.-Memoria%20(1).docx#_Toc105795971)

[Ilustración 8 Pantalla principal 12](https://mgep-my.sharepoint.com/personal/unai_sasiain_alumni_mondragon_edu/Documents/1.-Memoria%20(1).docx#_Toc105795972)

[Ilustración 9 Pantalla Historial 13](https://mgep-my.sharepoint.com/personal/unai_sasiain_alumni_mondragon_edu/Documents/1.-Memoria%20(1).docx#_Toc105795973)

[Ilustración 10 Pantalla Proveedores 13](https://mgep-my.sharepoint.com/personal/unai_sasiain_alumni_mondragon_edu/Documents/1.-Memoria%20(1).docx#_Toc105795974)

[Ilustración 11 Pantalla Inventario 14](https://mgep-my.sharepoint.com/personal/unai_sasiain_alumni_mondragon_edu/Documents/1.-Memoria%20(1).docx#_Toc105795975)

[Ilustración 12 Pantalla Beneficiarios 14](https://mgep-my.sharepoint.com/personal/unai_sasiain_alumni_mondragon_edu/Documents/1.-Memoria%20(1).docx#_Toc105795976)

[Ilustración 13 Creación de Unidad Almacenamiento 15](https://mgep-my.sharepoint.com/personal/unai_sasiain_alumni_mondragon_edu/Documents/1.-Memoria%20(1).docx#_Toc105795977)

[Ilustración 14 Pantalla Usuarios 15](https://mgep-my.sharepoint.com/personal/unai_sasiain_alumni_mondragon_edu/Documents/1.-Memoria%20(1).docx#_Toc105795978)

[Ilustración 15 Red del sistema 16](https://mgep-my.sharepoint.com/personal/unai_sasiain_alumni_mondragon_edu/Documents/1.-Memoria%20(1).docx#_Toc105795979)

[Ilustración 16 Conexión Serial 19](https://mgep-my.sharepoint.com/personal/unai_sasiain_alumni_mondragon_edu/Documents/1.-Memoria%20(1).docx#_Toc105795980)

[Ilustración 17 Nube de puntos 21](#_Toc105795981)

[Ilustración 18 Recta de la regresión 21](#_Toc105795982)

[Ilustración 19 Formula T de student 22](https://mgep-my.sharepoint.com/personal/unai_sasiain_alumni_mondragon_edu/Documents/1.-Memoria%20(1).docx#_Toc105795983)

[Ilustración 20 Grafico 22](https://mgep-my.sharepoint.com/personal/unai_sasiain_alumni_mondragon_edu/Documents/1.-Memoria%20(1).docx#_Toc105795984)

[Ilustración 21 Formula T1 23](https://mgep-my.sharepoint.com/personal/unai_sasiain_alumni_mondragon_edu/Documents/1.-Memoria%20(1).docx#_Toc105795985)

[Ilustración 22 Limites B1 y B0 23](https://mgep-my.sharepoint.com/personal/unai_sasiain_alumni_mondragon_edu/Documents/1.-Memoria%20(1).docx#_Toc105795986)

[Ilustración 24 Predicciones 23](https://mgep-my.sharepoint.com/personal/unai_sasiain_alumni_mondragon_edu/Documents/1.-Memoria%20(1).docx#_Toc105795987)

[Ilustración 25 Grafica general 24](https://mgep-my.sharepoint.com/personal/unai_sasiain_alumni_mondragon_edu/Documents/1.-Memoria%20(1).docx#_Toc105795988)

[Ilustración 26 Caso de uso total 40](https://mgep-my.sharepoint.com/personal/unai_sasiain_alumni_mondragon_edu/Documents/1.-Memoria%20(1).docx#_Toc105795989)

[Ilustración 27 Caso de uso general 40](https://mgep-my.sharepoint.com/personal/unai_sasiain_alumni_mondragon_edu/Documents/1.-Memoria%20(1).docx#_Toc105795990)

[Ilustración 28 Caso de uso recibidos 41](https://mgep-my.sharepoint.com/personal/unai_sasiain_alumni_mondragon_edu/Documents/1.-Memoria%20(1).docx#_Toc105795991)

[Ilustración 29 Caso de uso inventario 44](#_Toc105795992)

[Ilustración 30 Caso de uso entregas 46](https://mgep-my.sharepoint.com/personal/unai_sasiain_alumni_mondragon_edu/Documents/1.-Memoria%20(1).docx#_Toc105795993)

[Ilustración 31 Caso de uso usuarios 49](https://mgep-my.sharepoint.com/personal/unai_sasiain_alumni_mondragon_edu/Documents/1.-Memoria%20(1).docx#_Toc105795994)

[Ilustración 32 DiagramaAct Registro de producto 55](https://mgep-my.sharepoint.com/personal/unai_sasiain_alumni_mondragon_edu/Documents/1.-Memoria%20(1).docx#_Toc105795995)

[Ilustración 33 DiagramaAct Crear usuarios 56](https://mgep-my.sharepoint.com/personal/unai_sasiain_alumni_mondragon_edu/Documents/1.-Memoria%20(1).docx#_Toc105795996)

[Ilustración 34 DiagramaClases Registro de productos 57](https://mgep-my.sharepoint.com/personal/unai_sasiain_alumni_mondragon_edu/Documents/1.-Memoria%20(1).docx#_Toc105795997)

[Ilustración 35 DiagramaClases Login 58](https://mgep-my.sharepoint.com/personal/unai_sasiain_alumni_mondragon_edu/Documents/1.-Memoria%20(1).docx#_Toc105795998)

[Ilustración 36 DiagramaClases Creación de usuarios 59](https://mgep-my.sharepoint.com/personal/unai_sasiain_alumni_mondragon_edu/Documents/1.-Memoria%20(1).docx#_Toc105795999)

[Ilustración 37 BD Modelo 60](https://mgep-my.sharepoint.com/personal/unai_sasiain_alumni_mondragon_edu/Documents/1.-Memoria%20(1).docx#_Toc105796000)

[Ilustración 38 BD Tablas 61](https://mgep-my.sharepoint.com/personal/unai_sasiain_alumni_mondragon_edu/Documents/1.-Memoria%20(1).docx#_Toc105796001)

[Ilustración 39 BD Modelo 1 62](https://mgep-my.sharepoint.com/personal/unai_sasiain_alumni_mondragon_edu/Documents/1.-Memoria%20(1).docx#_Toc105796002)

[Ilustración 40 BD Tablas 1 63](https://mgep-my.sharepoint.com/personal/unai_sasiain_alumni_mondragon_edu/Documents/1.-Memoria%20(1).docx#_Toc105796003)

[Ilustración 41 BD Modelo entero 64](https://mgep-my.sharepoint.com/personal/unai_sasiain_alumni_mondragon_edu/Documents/1.-Memoria%20(1).docx#_Toc105796004)

[Ilustración 42 Creación de tablas 1 64](https://mgep-my.sharepoint.com/personal/unai_sasiain_alumni_mondragon_edu/Documents/1.-Memoria%20(1).docx#_Toc105796005)

[Ilustración 43 Creación de tablas 2 65](https://mgep-my.sharepoint.com/personal/unai_sasiain_alumni_mondragon_edu/Documents/1.-Memoria%20(1).docx#_Toc105796006)

[Ilustración 44 Creación de tablas 3 65](https://mgep-my.sharepoint.com/personal/unai_sasiain_alumni_mondragon_edu/Documents/1.-Memoria%20(1).docx#_Toc105796007)

[Ilustración 45 Motores de almacenamiento 65](https://mgep-my.sharepoint.com/personal/unai_sasiain_alumni_mondragon_edu/Documents/1.-Memoria%20(1).docx#_Toc105796008)

[Ilustración 46 Usuarios BD 66](https://mgep-my.sharepoint.com/personal/unai_sasiain_alumni_mondragon_edu/Documents/1.-Memoria%20(1).docx#_Toc105796009)

[Ilustración 47 Seguridad BD 67](https://mgep-my.sharepoint.com/personal/unai_sasiain_alumni_mondragon_edu/Documents/1.-Memoria%20(1).docx#_Toc105796010)

[Ilustración 48 Conexión BD - JAVA 1 68](https://mgep-my.sharepoint.com/personal/unai_sasiain_alumni_mondragon_edu/Documents/1.-Memoria%20(1).docx#_Toc105796011)

[Ilustración 49 Conexión BD - JAVA 2 68](https://mgep-my.sharepoint.com/personal/unai_sasiain_alumni_mondragon_edu/Documents/1.-Memoria%20(1).docx#_Toc105796012)

[Ilustración 50 Conexión BD - JAVA 3 69](https://mgep-my.sharepoint.com/personal/unai_sasiain_alumni_mondragon_edu/Documents/1.-Memoria%20(1).docx#_Toc105796013)

[Ilustración 51 RB 75](https://mgep-my.sharepoint.com/personal/unai_sasiain_alumni_mondragon_edu/Documents/1.-Memoria%20(1).docx#_Toc105796014)

[Ilustración 52 RS 75](https://mgep-my.sharepoint.com/personal/unai_sasiain_alumni_mondragon_edu/Documents/1.-Memoria%20(1).docx#_Toc105796015)

[Ilustración 53 Protocolo UART 76](https://mgep-my.sharepoint.com/personal/unai_sasiain_alumni_mondragon_edu/Documents/1.-Memoria%20(1).docx#_Toc105796016)

[Ilustración 54 Protocolo UART 2 76](#_Toc105796017)

[Ilustración 55 TX/RX baud 77](https://mgep-my.sharepoint.com/personal/unai_sasiain_alumni_mondragon_edu/Documents/1.-Memoria%20(1).docx#_Toc105796018)

[Ilustración 56 Función de lectura de datos 1 78](https://mgep-my.sharepoint.com/personal/unai_sasiain_alumni_mondragon_edu/Documents/1.-Memoria%20(1).docx#_Toc105796019)

[Ilustración 57 Función de lectura de datos 2 78](https://mgep-my.sharepoint.com/personal/unai_sasiain_alumni_mondragon_edu/Documents/1.-Memoria%20(1).docx#_Toc105796020)

[Ilustración 58 Función visualización 1 79](https://mgep-my.sharepoint.com/personal/unai_sasiain_alumni_mondragon_edu/Documents/1.-Memoria%20(1).docx#_Toc105796021)

[Ilustración 59 Función visualización 2 79](https://mgep-my.sharepoint.com/personal/unai_sasiain_alumni_mondragon_edu/Documents/1.-Memoria%20(1).docx#_Toc105796022)

[Ilustración 60 Distribución de ficheros 81](https://mgep-my.sharepoint.com/personal/unai_sasiain_alumni_mondragon_edu/Documents/1.-Memoria%20(1).docx#_Toc105796023)

[Ilustración 61 Sustitución de valores query 83](https://mgep-my.sharepoint.com/personal/unai_sasiain_alumni_mondragon_edu/Documents/1.-Memoria%20(1).docx#_Toc105796024)

[Ilustración 62 Conexión base de datos 83](https://mgep-my.sharepoint.com/personal/unai_sasiain_alumni_mondragon_edu/Documents/1.-Memoria%20(1).docx#_Toc105796025)

[Ilustración 63 Formulario Registro 84](https://mgep-my.sharepoint.com/personal/unai_sasiain_alumni_mondragon_edu/Documents/1.-Memoria%20(1).docx#_Toc105796026)

[Ilustración 64 Verificacion Inicio Sesion 85](https://mgep-my.sharepoint.com/personal/unai_sasiain_alumni_mondragon_edu/Documents/1.-Memoria%20(1).docx#_Toc105796027)

# INTRODUCCIÓN

En esta memoria se habla sobre la solución que la empresa TidyFood da a los problemas de la gestión de productos en los bancos de alimentos con el objetivo de reducir la cantidad de alimentos que son desechados principalmente en supermercados. Primero se hablará sobre el diseño del producto, después el funcionamiento de este, para continuar con la infraestructura de la empresa y finalizar explicando las conclusiones sacadas por la empresa TidyFood.

En España el porcentaje de la cantidad de comida desperdiciada es muy alto. Es por ello por lo que se decidió desarrollar esta aplicación para dar soporte a los bancos de alimentos a la hora de solucionar este problema.

A través de esta aplicación se busca informatizar y automatizar la infraestructura organizativa de un banco de alimentos proporcionando una base de datos mediante la cual se maneje, no solo el inventario del almacén de alimentos, sino también el tráfico de alimentos y dinero entre proveedores, banco y beneficiarios. Logrando de esta forma poder aplicar al mismo tiempo, una serie de algoritmos de automatización, mediante los cuales aplicar sistemas de inventariado de manera que los alimentos de menor vida útil, que por lo general coinciden con los alimentos almacenados durante un mayor tiempo, se envíen antes que el resto a fin de aprovechar lo máximo posible todos los recursos.

Además, en un comienzo se plantearon diferentes ideas a desarrollar, que fueron estudiadas y descartadas por la descompensación de pros y contras o simplemente por razones éticas hasta acabar decidiendo la idea del sistema del restaurante. Estas ideas y las razones por las que se descartaron se pueden observar en el [ANEXO 1 - IDEAS INICIALES](#_bookmark44).

# PLANTEAMIENTO DEL ÁMBITO

Para comenzar, es imprescindible entender el funcionamiento común de un banco de alimentos. Este está principalmente dividido en tres funcionalidades: gestión de donaciones, gestión del inventario y por último gestión de entregas.

Por un lado, desde la aplicación se podrán registrar las donaciones de los proveedores. En este apartado se incluyen todas las funciones necesarias para registrar las donaciones de los proveedores y guardar estas en cajas en el inventario. En esta área el usuario tendrá que indicar quien es el proveedor, que productos ha donado y en qué fecha caducan para luego almacenar estos y que los datos se guarden automáticamente en la base de datos.

Por otro lado, con respecto a la gestión del inventario, el sistema dispondrá de interfaces para ver los productos almacenados en el inventario y podrá deshacerse de estos en caso de que vayan a caducar antes de ser entregados a los beneficiarios. Los datos del inventario se actualizarán a tiempo real al recibir o entregar los productos y algunos de los datos se mostrarán los datos de manera gráfica. Los datos del inventario serán necesarios a la hora de cumplir con las demás funcionalidades

Por último, la última funcionalidad que cumple la aplicación es la de gestionar las entregas de los lotes. Los lotes serán repartidos a los beneficiarios en lotes y los propios beneficiarios podrán elegir que lotes quieren entre los creados por el empleado del banco de alimentos. Los beneficiarios únicos podrán coger tan solo un lote por familia y por día.

Una vez aclaradas las tres funcionalidades principales en las que todo banco de alimentos se basa, habría que concretar cuál es el proceso por el que pasan los productos desde la fase de recogida hasta la entrega de los lotes.

## ÁREA DE OPERACIONES TRADICIONAL

A continuación, se describe el área de operaciones que un banco de alimentos común posee en general. Esta área se dividirá en tres secciones indispensables para el proceso, las cuales serán recibo, almacenamiento y entrega.

### RECIBO

La donación del proveedor es recogida por el banco de alimentos o entregada directamente al banco. Para tener la trazabilidad de los productos recibidos los trabajadores disponen de un cuaderno donde apuntan la información sobre los productos que reciben.

### ALMACENAMIENTO

Una vez los productos están en el banco de alimentos y ya están registrados empieza la fase de almacenamiento.

Cada banco de alimentos su manera de almacenar productos, se pueden almacenar en distintas unidades de almacenamiento y también puede variar dependiendo del tamaño del almacén, la cantidad de productos que reciben, etc.

En conclusión, lo que sí que tienen todos los bancos de alimentos en común es que almacenan los productos recibidos en sus determinadas unidades de almacenamiento y les ponen una etiqueta para poder identificar los productos más tarde.

### ENTREGA

Por último, la mayoría de los bancos de alimentos utilizan lotes de alimentos para la entregar los alimentos, generalmente lotes predeterminados por los bancos de alimentos.

Con los productos del inventario se trata de hacer la mayoría de los lotes de alimentos posible. Por cada producto que sacan de su unidad de almacenamiento y meten en un lote tienen que actualizar a mano el stock de la unidad de almacenamiento y en general la del inventario del banco de alimentos.

Al entregar los productos si algún lote no es entregado habrá que volver a meter los productos en el inventario o bien deshacerse definitivamente de los productos si van a caducar.

# PLANTEAMIENTO DEL PROBLEMA Y NUEVO FUNCIONAMIENTO

Al método de trabajo anteriormente comentado el grupo ha identificado unos problemas de eficiencia. Este proceso es una gran carga de trabajo y algunas veces incluso lioso.

La solución que presenta la empresa TidyFood para responder a estos problemas es una aplicación la cual consiste en automatizar y digitalizar todos los datos durante el proceso por el que pasa un producto desde que entra a un banco de alimentos hasta que sale. Esta aplicación alivia cargas de trabajo y asegura que no haya fallos a la hora de registrar o modificar los datos. La forma de trabajar no cambia mucho para los trabajadores por lo que la adaptación a esta nueva forma de trabajo no será demasiado costosa.

Al igual que en el área de operaciones tradicional, se dividirá esta área en tres secciones indispensables para el proceso, las cuales tendrán su pantalla propia.

## RECIBO

El procedimiento es muy similar al anterior, pero con la aplicación desarrollada por TidyFood se resolverán algunos de los problemas que pueden ocurrir al no digitalizar los datos registrados.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamenteAl igual que el proceso tradicional, con la implantación de TidyFood habrá un trabajador que se encargue de recoger los productos recibidos y registrar estos. Para ello el usuario tendrá que iniciar sesión en la aplicación. El usuario tendrá que estar ya registrado para que pueda iniciar sesión con sus credenciales. Una vez iniciada la sesión, desde la pantalla principal el usuario tendrá que rellenar algunos necesarios para registrar los productos de cada donación.

Ilustración 1 Interfaz registro de donaciones

Al acabar de rellenar los campos necesarios el usuario tendrá que dar al botón de registrar y en ese momento los productos quedan ya registrados.

## ALMACENAMIENTO

Una vez los productos son registrados se tienen que hacer cajas con los productos registrados que aún no se encuentran en cajas. Para ello, en el menú de la página principal hay un botón “Crear Cajas” que lleva al usuario directamente a la interfaz para crear cajas.

Una vez en la interfaz de crear cajas, el usuario tendrá que elegir los productos que quiere meter en la caja y dar al botón “Crear” para que la caja quede registrada. La función principal de la creación de cajas es para, en un futuro a la hora de crear lotes tener la trazabilidad de cada uno de los productos registrados en el inventario. Después de crear cajas con los productos registrados, se asume que los productos ya se encuentran en el inventario y que están guardadas en diferentes cajas.

En el instante momento en el que el usuario pulsa el botón “Crear” se creará una caja que será registrada en la base de datos junto a los productos incluidos en ella.

## ENTREGA

En cuanto a las entregas, en nuestra aplicación se utilizarán lotes para entregar los productos a los beneficiarios. Se podrán crear hasta seis tipos de lotes para que a la hora de la entrega haya variedad de lotes. A la hora de crear los lotes se deberá tener en cuenta principalmente que haya variedad, así los beneficiarios que tengan intolerancias tendrán posibilidades de escoger uno de los lotes sin problemas.

Para crear los lotes el usuario desde la pantalla principal accederá a la pantalla de creación de lotes mediante el botón “Crear lotes”. En esa interfaz lo primero que tendrá que hacer es pasar los productos del inventario a los lotes indicando que cantidad de ese producto quiere pasar al lote. Cuando tenga todos los productos en el lote, el usuario tendrá que dar al botón crear y elegir cuantos lotes de ese tipo quiere crear. Una vez todos los tipos de lotes son creados, se abrirá una pantalla con la información de cada uno de los lotes para que el usuario pueda verificar que los tipos de lotes creados y las cantidades de productos que tienen estos son correctos.

A diferencia del procedimiento estándar, con el procedimiento aplicado por TidyFood se consigue una mayor fluidez de trabajo, un registro de todos los procesos del área de operaciones y una mayor efectividad y tolerancia a errores.

# DISEÑO DEL PRODUCTO

Antes de nada, hay que tener en cuenta la forma de implementar la aplicación. Esta va a estar instalada en unos dispositivos en varios lugares clave del banco de alimentos, principalmente en los almacenes.

Interfaz de usuario gráfica

Descripción generada automáticamenteTidyFood dispone de varias interfaces para cumplir con cada una de las funcionalidades que nos ofrece la aplicación. Así, se ha diseñado la pantalla principal de forma que todas las funcionalidades son accesibles desde ella. Fuera de la pantalla principal, en la mayoría de las pantallas, también se mantendrá el menú y el menú de botones por lo cual casi todas las pantallas serán accesibles de las demás pantallas. Aun así, habrá funcionalidades que solo serán accesibles desde la pantalla principal.

Ilustración 2 Botonera

Para usar la aplicación de TidyFood se necesitará tener acceso a la base de datos guardada en los servidores de la sede central de la empresa, ya que es la única fuente de los datos que se muestran en pantalla. Una vez el banco de alimentos se conecte al servidor donde se encuentra la base de datos, el programa se encargará de actualizar la base de datos y junto a la base de datos las interfaces del programa.

Al encender la aplicación lo primero que habrá que hacer será iniciar sesión con el usuario que tendrá cada trabajador. Las funcionalidades que se puedan cumplir en la aplicación variarán dependiendo de cada usuario. Por lo que habrá dos tipos de usuarios para entrar a la aplicación y empezar a utilizarla: usuarios normales sin permisos de administrador y usuarios con permisos de administrador.

Los usuarios con permisos de administrador, además de poder hacer todo lo que puedan hacer los demás usuarios, tendrán la posibilidad de crear nuevos usuarios. Esos datos introducidos por el usuario administrador pasarán directamente a una tabla de la base de datos que contendrá todos los usuarios con sus contraseñas.

Una vez iniciada la sesión cada trabajador podrá cumplir con sus funcionalidades a través de la aplicación. No se restringirá ninguna de las funcionalidades (a parte de las funcionalidades del administrador) a los trabajadores a través de la aplicación, pero sí que habrá que organizar quien va a cumplir con cada labor.

En los almacenes donde se guarden las cajas de comida, para ayudar a identificar las cajas, mediante una placa, se ha implementado un lector de código de barras que permitirá leer los códigos de barras que tendrán las cajas, mostrando así en la pantalla de la misma placa la información sobre la caja. La placa mostrará: el id de la caja, los nombres de los productos que contiene la caja y la fecha de caducidad del producto con la fecha de caducidad más reciente.

Esto facilitará el acceso al contenido de las cajas y para los trabajadores será mucho más fácil identificar qué contiene cada una de las cajas.

# PLANIFICACIÓN

Imagen que contiene Gráfico

Descripción generada automáticamenteEn la siguiente tabla ([Ilustración 5-1 Planificación](#_bookmark12)) se puede ver la planificación que se ha utilizado para conseguir llevar a cabo el producto. El grupo ha optado por utilizar un diagrama de Gantt en el cual se han definido fases genéricas y partes más específicas para desarrollar estas fases. Los tiempos han sido orientativos para el grupo y al final ha habido algunas tareas que se han alargado y otras que se han terminado antes.

Ilustración 3 Planificación

# DESARROLLO DE LA APLICACIÓN

Para un mejor desarrollo del sistema se ha decidido utilizar un proceso de desarrollo unificado. Este es un proceso iterativo e incremental dirigido por casos de uso. De esta manera, se capturan los requisitos del usuario a través de diferentes tipos de modelos que ayudarán a identificar y especificar la estructura de la aplicación gráfica, así como a planificar los incrementos y las iteraciones.

Una vez bien definidas las necesidades del producto en la fase de análisis y correctamente documentadas en el ERS (Especificación de Requisitos de Software), que se presenta en el anexo x, se comenzó con el diseño de la aplicación.

Interfaz de usuario gráfica

Descripción generada automáticamenteSin embargo, además de esto sería necesario definir la forma del sistema. Por ello, como se observa en las siguientes imágenes, se realizó un primer boceto de la forma que tendrían las diferentes ventanas a desarrollar. La mayoría de las ventanas tienen la parte de la botonera y diferentes apartados divididos.

Ilustración 4 Boceto pantalla principal

Word

Descripción generada automáticamente con confianza bajaInterfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Ilustración 5 Boceto pantalla creación de lotes

Ilustración 6 Boceto pantalla inventario

Una vez hecho esto, se podía comenzar con la fase de implementación. Cada desarrollador podía comenzar con su incremento para, más adelante, acabar comprobando el correcto funcionamiento del sistema en su conjunto. A partir de estos bocetos empezamos a desarrollar la aplicación y fuimos añadiendo nuevas funcionalidades a lo largo del proyecto.

Por otro lado, para el correcto funcionamiento de la aplicación se han tenido que diseñar otros sistemas. Para empezar, era imprescindible disponer de un sistema de almacenamiento de datos que se diseñó en una fase temprana del desarrollo para poder crear la aplicación de la manera más fiel posible, es decir, obteniendo los datos directamente desde este sistema de almacenamiento. Además, se utilizó un microcontrolador que serviría de enlace entre el lector de tarjetas y la aplicación desarrollada.

# FUNCIONAMIENTO DEL PRODUCTO

En el punto actual se explicará el funcionamiento de la aplicación y su asociación con la placa.

Interfaz de usuario gráfica

Descripción generada automáticamenteLo primero de todo, para conseguir acceso a la pantalla principal, cada trabajador se deberá de identificar con su usuario y contraseña en una interfaz de inicio de sesión. Los datos introducidos se verificarán en la base de datos y al verificar estos, el usuario podrá acceder a la pantalla principal.

Ilustración 7 Pantalla Inicio

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamenteLa pantalla principal de la aplicación es el centro de control del producto, pantalla desde la cual se puede acceder a cualquier funcionalidad disponible. La cual dispone de un formulario para registrar productos recibidos y la capacidad ver los proveedores junto con una pequeña descripción del proveedor seleccionado. En la parte superior, se encuentra la barra de tareas formada con botones y menús, que será útil para acceder a diferentes pantallas del software.

Ilustración 8 Pantalla principal

En lo que al formulario se refiere, este te da la posibilidad de registrar cualquier producto proporcionándole de los datos necesarios para el registró del mismo. A cada registro que se hace, se le asigna una fecha de donación, que es la fecha real del día que se registra el producto en la aplicación. A su derecha, nos encontramos un panel donde nos muestra información de los proveedores junto con un botón de agrandar lista que te muestra todos los proveedores que tiene el banco de alimentos.

Gráfico

Descripción generada automáticamenteRespecto a la barra de botos y el menú, lo primero que nos encontramos si empezamos por la izquierda es el botón y la opción de proveedores. Este icono nos llevara directamente a la pantalla de proveedores, pantalla en la que se muestran todos los órganos donantes que ayudan a reducir el desperdicio de alimentos.

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamenteEn la pantalla mencionada en el párrafo anterior, nos encontramos cuatro iconos o botones diferentes y un buscador, empezando de izquierda a derecha; lo primero que nos encontramos es la opción de volver a la pantalla principal. Junto a este está el buscador de proveedores para facilitar la búsqueda de información sobre un donante. En la parte de la derecha, el primer botón que se encuentra es el de ordenar los proveedores por orden alfabético. A continuación, está el icono que nos mostrara información sobre los donantes y finalmente tenemos el icono de cerrar la aplicación. Si el usuario clica en alguno de los proveedores estando este en verde, se montará la información de dicho proveedor. Finalmente, tenemos el plus, opción que permitirá al usuario registrar un nuevo proveedor.

Ilustración 9 Pantalla Historial

Ilustración 10 Pantalla Proveedores

Volviendo a la pantalla principal, la segunda opción que nos encontramos es la opción del historial, pantalla donde se almacena el registro de todos los productos que han estado en el almacén.

La primera opción que nos permite esta pantalla es filtrar los productos según los grupos alimentarios, para facilitar la visualización de información sobe X producto. la aplicación da la opción de busca cualquiera de los productos almacenados de manera fácil y rápida mediante un buscador de productos. Observando la parte del centro derecha de la barra, se encuentra la opción de visualizar gráficamente la entrada y salida de X alimento en un tiempo máximo de 4 semanas. Finalmente, para volver a la pantalla principal o cerrar la aplicación, están los dos iconos de la derecha.

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza bajaLa tercera opción que nos da la barra de opciones de la pantalla central es la de visualizar es inventario. Los productos se registrar y a continuación, se guardan en diferentes cajas según el producto y su grupo. Los alimentos registrados en cajas se guardan directamente en el inventario.

Ilustración 11 Pantalla Inventario

La pantalla de inventario es muy similar a la de historial, solo que esta al ser un simple inventario no entrega ninguna visualización de datos gráficos al usuario.

Interfaz de usuario gráfica

Descripción generada automáticamenteDe vuelta en la pantalla principal, la aplicación da la opción de crear beneficiarios clicando el icono o eligiendo la opción de ir a la pantalla de beneficiarios, pantalla en la que nos permite, como se ha dicho anteriormente, tanto crear, editar y ver intolerancias de este.

Ilustración 12 Pantalla Beneficiarios

Sobre los botones de esta interfaz, el plus es la opción que permite al usuario crear un beneficiario nuevo, mientras que el lápiz te permite editar la información sobre cualquier beneficiario seleccionado. Como es de costumbre, el icono de la “I” es el que nos proporciona la información.

Imagen que contiene Interfaz de usuario gráfica

Descripción generada automáticamentePara la creación de unidades de almacenamientos, tenemos las opciones de crear caja y crear lote. Estas 2 pantallas son muy similares, las 2 te permiten crear una unidad de almacenamiento, que en el caso de las cajas irán al inventario y en de ser lotes, serán entregados.

Ilustración 13 Creación de Unidad Almacenamiento

La pantalla está formada por una lista para visualizar el inventario y otra para ver los productos introducidos a la unidad. Para la introducción o eliminación de un producto, la aplicación dispone de 2 botones principales, con la fecha para indicar la función de cada uno de ellos. Una vez creados los lotes o cajas, el usuario deberá de presionar el botón crear y, en el caso de crear un lote, entregar, para la entrega y creación de estos

Para finalizar, tenemos la pantalla de creación y visualización de usuario, pantalla que solamente el administrador del sistema le puede sacar partido, ya que el será el único con posibilidad de crear uno pulsando el icono del plus, modificar clicando en el lápiz y visualizar su información, eligiendo la opción de la información.

Interfaz de usuario gráfica, Aplicación, Tabla

Descripción generada automáticamente

Ilustración 14 Pantalla Usuarios

# RED DEL SISTEMA

Diagrama, Esquemático

Descripción generada automáticamenteSe ha diseñado una red de la empresa para proporcionar servicios de red a los trabajadores de manera rápida, segura y redundante.

Ilustración 15 Red del sistema

Como se aprecia en la imagen superior, la empresa cuenta con un servidor alojado en otro edificio en el que se encontraran todas las copias de seguridad de la empresa, para seguir los protocolos de seguridad en caso de desastre. El servidor maestro, alojado en la oficina de la empresa, es el encargado de proporcionar los servicios DHCP, DNS y NTP a la empresa. Por otro lado, el servidor llamado “Ficheros” es el encargado de gestionar las bases de datos con la información relacionada con los restaurantes y al que estos acceden.

En la red del sistema se ha intentado maximizar el espacio utilizado de las direcciones IP. Para eso se ha implementado VLSM, lo cual ha dividido la red principal en varias subredes diferentes. De esta forma se optimiza más el uso de las direcciones IP para cada subred. Y permite evitar el agotamiento de direcciones IP, ya que se ha utilizado una IPv4.

Para salir de la red de la oficina a las redes de los restaurantes se ha utilizado una NAT estática. Esto permitirá traducir las direcciones IP privadas en direcciones IP públicas y, al contrario, las públicas en privadas. Así los paquetes podrán viajar por la red de Internet con una sola IP pública, en vez de utilizar muchas direcciones IP diferentes. Por otra parte, esta configuración limita la posibilidad de conectarse con los dispositivos de la red. Por lo que se ha tenido que implementar una ruta estática, la cual permitirá a los restaurantes conectarse al servidor con la base de datos instalada para poder consultarla.

Se ha configurado el protocolo CDP con el objetivo de ser más sencilla la inspección de la red. Este protocolo hará que sea mucho más sencillo el obtener información de enrutadores y switches que están conectados localmente. Dado la actual tendencia de las redes por una dinámica cambiante este protocolo se vuelve bastante útil. Por lo cual, al descubrir los dispositivos vecinos más fácilmente, el trabajo de esquematizar la red se hace más eficientemente.

Con el objetivo de tener sincronizados todos los dispositivos de la red, se ha instalado un servidor NTP. Lo que hace este servidor es marcar la hora para todos los dispositivos que se conecten a él. Por lo cual, todos los dispositivos tendrán la misma hora y estarán sincronizados.

Por otro lado, para poder hacer copias de seguridad de las configuraciones de los dispositivos de la red. También se ha instalado un servidor TFTP, el cual hará de almacenamiento para diferentes archivos. De ser necesario por falta de almacenamiento, este servidor además de guardar las copias de seguridad también será capaz de guardar otros tipos archivos u otras copias.

Con el objetivo de facilitar la administración de la red se ha instalado un servidor SNMP. Esto permitirá a los administradores de la red supervisar la red, facilitará localizar y resolver los posibles problemas que ocurran y planear el crecimiento de la red en un futuro.

Por otro lado, se ha utilizado OSPF multitarea para conectar las diferentes áreas de la red. Sin esta configuración los dispositivos dentro de diferentes redes no serían capaces de comunicarse entre ellos. Por ejemplo, dos redes conectadas entre dos enrutadores, para poder tener comunicación necesitan saber qué red está al otro lado del enrutador. Por lo que OSPF es una forma de hacer esta conexión automáticamente.

Para asegurarse de mantener la privacidad y la integridad de los datos que se estén transmitiendo por la red de la empresa, se ha optado por configurar varios protocolos de seguridad. Estos protocolos se especificarán más detalladamente a continuación:

Se ha configurado OSPF con autenticación MD5. Se habilita esta autenticación para intercambiar la información de actualización de ruteo de una forma segura. Al configurar OSPF, por defecto este protocolo no utiliza ningún tipo de cifrado para los mensajes que se envían. Por lo que al activar la autenticación MD5 hará que se utilicen contraseñas y estas contraseñas se cifran. Así evitando ser interceptado por algún dispositivo externo o nocivo, aumentando la seguridad en la red.

Además de eso, se ha configurado PPP con autenticación en varias interfaces de los enrutadores de la red. Esto hará que la comunicación entre dos redes diferentes sea más segura. Para ello se configuró un usuario y contraseña para la comunicación. Después la contraseña se cifra para no poder ser recogida por ningún intruso. Al final, se consigue que nadie pueda utilizar la conexión punto a punto entre los dos enrutadores sin la contraseña.

Por seguridad de la red también se han implementado varias ACL. Las ACL bloquearán tráfico no deseado dentro o fuera de la red. Por ejemplo, en la red de la empresa se ha instalado un ACL. Una de ellas está instalada en el enrutador principal de la oficina y evita que cualquier conexión externa a esa red se conecte con el servidor de ficheros. Para que nadie que esté fuera de la oficina sea capaz de conectarse a ese servidor. Así se puede asegurar que sólo los dispositivos dentro de la oficina pueden conectarse, y en caso de haber un ataque no se podría entrar a ese servidor más que desde la oficina.

# CONEXIÓN SERIAL

En el contexto del proyecto TidyFood la conexión serial se da en dos puntos del sistema, siendo ambos puntos en los que se requiere transferir información entre dos dispositivos diferentes. Por ello antes de comenzar con la explicación es necesario describir la topología del sistema de forma que se comprendan tanto la situación como el propósito de cada uno de los puntos en los que se da la mencionada conexión serial.

Imagen de la pantalla de un celular con la imagen de un videojuego

Descripción generada automáticamente con confianza bajaEs sistema o en nuestro caso el entorno de simulación, puesto que TidyFood todavía se encuentra en fase de desarrollo, está conformado por 4 dispositivos diferentes interconectados entre sí. Estos son, un servidor (PC portátil) en el que corren tanto el cliente de TidyFood como la base de datos principal, un servidor de respaldo (Msi mpg trident 3) en el que junto al primer servidor se aplica la replicación para dar mayor seguridad al sistema, una placa de desarrollo (Olimex STM32-P107), utilizada para mostrar información y configurar periféricos y por último un módulo externo (Barcode Click) empleado junto a placa de desarrollo en la lectura de códigos QR.

Ilustración 16 Conexión Serial

Como se observa en el diagrama de la parte superior, el servidor principal se conecta con el servidor secundario o de respaldo mediante la red local en la que ambos se encuentran, y lo hacen de forma bidireccional. En cambio, el servidor utilizando el cliente TidyFood, se comunica con la placa de desarrollo mediante un cable serial, siendo que ambos extremos están configurados para utilizar el protocolo de comunicación UART. En este caso la comunicación también es bidireccional, debido a que ambos dispositivos necesitan intercambiar información para realizar sus respectivas funciones.

Finalmente, el módulo de lectura de códigos impresos se comunica de la misma manera con la placa, si bien es cierto que en vez de un cable serial utiliza un conjunto de cables que realizan la misma función utilizando el mismo protocolo, aunque en este caso la comunicación es unidireccional, ya que, en este caso no se requiere un intercambio de información, sino una lectura de información que simplemente se transmite a los dispositivos que requieren utilizarla.

En resumen, la comunicación serial se da entre la placa y el servidor maestro además de entre la placa de desarrollo y el módulo de lectura de códigos impresos. Estos enlaces se utilizan en la transmisión de datos de utilización inmediata como son los códigos leídos o parámetros obtenidos en base a estos, los cuales no se guardan, solo se emplean para realizar la acción requerida en dicho momento y al recibir nueva información esta se pierde, lo cual no es un problema siendo que el usuario es quien elige cuando se lee y por ende se recibe, nueva información.

# REPERCUSIÓN DEL PRODUCTO

La idea principal del producto es mejorar y reducir el desperdicio de comida tanto en supermercado, restaurante y bares, con el fin de reducir lo máximo posible el hambre en el mundo.

Se ha utilizado la regresión lineal para inferir en la comida que se desperdicia según el nivel de riqueza de cada país. La idea sería conseguir los datos de cada gobierno del país participe en esta investigación, pero dado que solamente alrededor de 17 países en el mundo proporcionan datos fiables sobre su desperdicio de comida, se ha tomado la decisión y la libertad de generar los datos de riqueza, tomando como referencia el nivel de desarrollo y riqueza que se le da a cada país y el rango de riqueza que obtienen según ese nivel. Respecto a los datos de desperdicio de comida, estos se han generado basándose en los datos obtenidos de los 17 países fiables. Para que estos datos se visualicen más claros se ha creado unas graficas donde se muestra la recta de la regresión y la nube de puntos.

Gráfico, Gráfico de dispersión

Descripción generada automáticamente

Ilustración 17 Nube de puntos

Gráfico, Gráfico de líneas

Descripción generada automáticamente

Ilustración 18 Recta de la regresión

Con la nube de puntos y los datos en crudo, para saber qué tipo de relación que tienen estos datos se debe de calcular el coeficiente de correlación, es decir, calcular la relación estadística entre 2 variables. Este coeficiente asciende desde -1 hasta 1 y representa la dirección de la recta de regresión, si el valor es 0 > 0 se trata de una relación positiva o ascendente y si es < 0 se trata de una relación inversa o descendente. Los valores más próximos a los límites de 1 y -1 indican una fuerte relación entre las dos variables. Se debe aplicar una hipótesis nula de que el coeficiente ρ (rho) es 0 utilizando la fórmula que se puede observar abajo, teniendo en cuenta de que r es un estimador de ρ. En este caso, el coeficiente de correlación se ha calculado con el método *Pearson* y el valor de esta correlación para la distribución es de 0,57 lo cual significa que hay una relación entre las dos variables, pero no es tan fuerte. Para descartar la hipótesis nula el resultado de T debe de ser mayor que tinv. El p-value de la hipótesis nula es de 0.

Ilustración 19 Formula T de student

Con la T calculada, se ha observado que al ser el valor de esta es 4,96 y el valor de tinv es 2.0086 se puede confirmar que la hipótesis no es nula.

Gráfico, Gráfico de dispersión

Descripción generada automáticamenteDespués de confirmar que la hipótesis que estamos intentando calcular no es nula, se empieza con el cálculo y estimación de 𝛽1 y de 𝛽0 que serán las constantes de la recta. Se utiliza el método de mínimos cuadrados el cual nos da una recta donde la suma de las diferencias entre los datos y los puntos de la recta da 0. En otras palabras, la suma de la distancia de los datos en crudo hasta la recta de regresión por encima y por debajo dan lo mismo. En esta observación se ha conseguido reflejar la correlación entre variables mediante un gráfico.

Ilustración 20 Grafico

Para asegurarse de que los datos tienen una correlación con un 95% de certeza se realiza una prueba de hipótesis donde se ha utilizado la fórmula que se puede observar abajo y se ha comparado con el valor de la tinv con un alfa de 0,05. En este caso la hipótesis nula es 𝛽1 = 0. Los resultados indican que esta hipótesis nula se puede descartar. El valor de p-value para la hipótesis nula de que 𝛽1 = 0 es de 0.



Ilustración 21 Formula T1

Después de calcular la formula se ha obtenido como resultado lo siguiente: 4,926> 2.0086.

Después de comprobar y verificar que los datos tienen una relación causal, hay que tener en cuenta que la recta de regresión representa a la muestra y no a la población entera. Los estimadores 𝛽1 y 𝛽0 siguen una distribución normal la cual representa a la población y se puede obtener una estimación de estos parámetros para el 95% de la población mediante el cálculo de intervalos de confianza. En esta observación el grupo ha sacado que el intervalo de confianza para 𝛽0 y 𝛽1 del 95%.

Ilustración 22 Limites B1 y B0

Texto

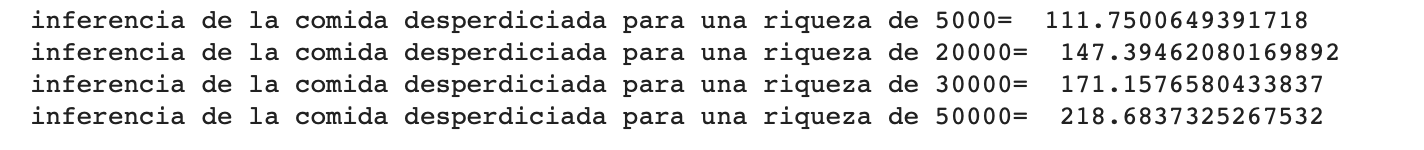
Descripción generada automáticamenteTeniendo en cuenta estos datos se puede estimar por dónde caerá la recta real de la regresión y cuál sería el intervalo donde saldrían las observaciones futuras, las dos con un intervalo de confianza del 95%. En el gráfico de abajo se pueden observar el intervalo de estimación para la respuesta media y el intervalo de predicción para una observación futura. Aquí abajo se puede observar teniendo cuanta comida desperdiciaría un país con una riqueza de 5000, 20000, 30000, 50000 de media al mes por persona.

Ilustración 24 Predicciones

Gráfico, Gráfico de dispersión

Descripción generada automáticamente

Ilustración 25 Grafica general

# CONCLUSIONES

Después de desarrollar este sistema el grupo ha llegado a unas conclusiones. La implementación de TidyFood en los bancos de alimentos supondrá una gran mejoría, en comparación al método tradicional de los bancos de alimentos, en cuanto a la trazabilidad de productos y análisis de datos

Además, todos los movimientos que se hagan en la aplicación quedarán registrados en los servidores de los bancos, así si en un futuro se quiere hacer un estudio sobre cualquier tema que afecte al banco de alimentos la información estará siempre disponible. Por poner un ejemplo, todas las donaciones y entregas quedarán registradas por lo cual, en un futuro, podríamos acceder a esa información y ver de qué proveedores son los productos que más se entregan a los beneficiarios.

Junto a todo esto, tener todos los productos que se encuentran en el inventario disponibles a través de un botón y hacer una búsqueda para encontrar un producto en concreto facilita mucho la forma de trabajar del banco de alimentos. Siguiendo la forma tradicional, se tendría que buscar el producto deseado entre cientos de papeles lo cual puede resultar muy tedioso.

En cuanto al coste de la instalación, creemos que con instalar un ordenador por almacén en el banco de alimentos sería suficiente. El coste solo supondría comprar esos ordenadores, teniendo en cuenta que la aplicación no exige tener características muy locas en el ordenador los ordenadores no se excederán en cuanto al precio.

Para finalizar, pensamos que la adaptación al nuevo sistema de trabajo sería más que sencilla para los trabajadores, ya que TidyFood es una aplicación muy visual y las funcionalidades que ofrece la aplicación son fácilmente accesibles para los usuarios. .

# CONCLUSIONES PERSONALES

Gracias a este proyecto hemos conseguido mejorar y reforzar los conocimientos de las diferentes áreas trabajadas durante el semestre y adquirir nuevos conocimientos en diferentes campos de manera autodidacta.

Por un lado, documentar todo antes de empezar a desarrollar el proyecto nos ha parecido muy útil ya que se definen todos los requisitos y evita dudas al desarrollarlo. Los casos de uso son un buen sistema de visualizar las funcionalidades a las que accede cada actor.

La parte de programar la aplicación nos ha ayudado a ampliar nuestro conocimiento sobre el lenguaje de programación de Java. Hemos descubierto que podemos conectar la aplicación a una base de datos para hacer consultas, actualizar los datos e introducir nuevos datos en esta base de datos.

Además, nos hemos dado cuenta del potencial del microcontrolador. Este microcontrolador nos permite cumplir con cualquier funcionalidad con los periféricos que tiene ya instalados e incluso crear librerías para la configuración de estos, facilitando trabajos futuros muy fácilmente, ya que el lenguaje de programación C es muy adecuado para este tipo de aplicaciones. También tiene infinidad de opciones para periféricos ya que se le puede conectar cualquier periférico siempre y cuando se programe el driver para este.

La comunicación serial también es muy completa y con un par de simples registros y la previa configuración de otros registros se pueden mandar y recibir mensajes seriales. Lo malo del STM32 es que crear estas librerías para la configuración de los periféricos y la programación de los drivers es muy tedioso ya que hay que utilizar direcciones de memoria las cuales hay que consultar en el manual de referencia de la placa, que es muy extenso.

Para finalizar, la parte del estudio de la repercusión de este producto ha sido un campo donde se podría ahondar muchísimo más ya que los datos han sido creados artificialmente y las conclusiones no son muy reales. Aunque ha sido una oportunidad para preparar un estudio sobre el impacto de este producto, el crear los datos con una función ya definida y metiendo un ruido constante tiene poca incertidumbre ya que el resultado es bastante predecible. La regresión es un campo muy interesante para el estudio de la evolución de la facturación y la repercusión de cualquier producto, ya sea una regresión lineal o una regresión exponencial. Utilizando los logaritmos y sus reglas podemos conseguir que una regresión exponencial se convierta en una regresión lineal.

En lo referente microcontroladores y la placa de desarrollo, se ha determinado que es crucial el revisar la documentación del fabricante antes de intentar realizar ningún ajuste o configuración basado en otros proyectos, dispositivos o referencias de otro tipo. Esto se debe a que cada fabricante puede implementar cambios por motivos técnicos o de rendimiento que pueden variar con respecto a otros dispositivos de iguales características o fabricante. Es especialmente relevante al momento de ahorrar tiempo, ya que, aunque se pierde parte de este revisando la documentación se amortiza sobradamente con el tiempo ahorrado en el método de prueba y error.

A este respecto, se ha considerado también que la elaboración de librerías es de gran ayuda durante el proceso de desarrollo, pues permite centrarse más fácilmente en los apartados problemáticos, dejando de lado en otros ficheros las funciones que no dan problemas. Otra gran ventaja es la facilidad de uso que otorga el usar dichas librerías al momento de llamar a ciertas funciones que se utilizan con gran frecuencia y variedad de periféricos.

Para finalizar con el campo de los microcontroladores, la mayor epifanía tiene que ver con el versionado del programa desarrollado en oposición con la constante edición del mismo código. La razón es que, el conservar diferentes, aunque erróneas versiones del programa, ayuda enormemente en la recuperación y revisión de funciones y fragmentos de código que con anterioridad funcionaban y que actualmente no lo hacen o lo hacen por. Además, ahorra tiempo a la hora de realizar pruebas, ya que existe constancia de la mayoría de pruebas ya realizadas en otras versiones.

# LÍNEAS FUTURAS

En los siguientes apartados se va a explicar posibles mejoras futuras.

## PLANTILLAS DE LOTES

Sería conveniente crear plantillas de lotes que contengan variedad de productos y asignar nombres a estos. A la hora de crear lotes, en caso de tener los productos necesarios, la creación de lotes sería mucho más sencilla. Incluso se podrían crear plantillas según que lotes son los que más se entregan a los beneficiarios.

## INTERFACES PARA LOS BENEFICIARIOS

También se ha planteado, hacer unas interfaces que cumplan con funcionalidades de los beneficiarios. Por ejemplo, hacer que los beneficiarios puedan registrarse en la aplicación y a la hora de recoger los lotes reservar su lote preferido.

Así llevar la trazabilidad de cuantos lotes se lleva cada beneficiario sería mucho más sencillo ya que se tendría todo registrado en la base de datos.

## REPERCUSIÓN DEL PRODUCTO EN UNA SITUACIÓN REAL

Estudiar la repercusión del producto en una situación real sería un caso realmente beneficioso para la empresa, ya que por el momento solo ha sido planteado teóricamente, y no se ha llevado a la práctica. El planteamiento inicial sería implementarlo en unos 5 bancos de alimentos y estudiar independientemente la repercusión. Después, comparar la información recibida y afirmar que la mejoría no ha sido casual sino causal.

# BIBLIOGRAFÍA

<https://ccnadesdecero.es/configuracion-de-ospf-multiarea/> [https://www.timigate.com/2018/05/cisco-wan-ppp-implementation-on-packet-](https://www.timigate.com/2018/05/cisco-wan-ppp-implementation-on-packet-tracert.html) [tracert.html](https://www.timigate.com/2018/05/cisco-wan-ppp-implementation-on-packet-tracert.html)

<https://ccnadesdecero.es/snmp-funcionamiento-configuracion/>

[https://www.itesa.edu.mx/netacad/networks/course/module3/3.3.2.1/3.3.2.1.html#](https://www.itesa.edu.mx/netacad/networks/course/module3/3.3.2.1/3.3.2.1.html#%3A~%3Atext%3DAutenticaci%C3%B3n%20del%20PPP%2Cred%20transmitan%20por%20el%20enlace)

[:~:text=Autenticaci%C3%B3n%20del%20PPP,red%20transmitan%20por%20el%20](https://www.itesa.edu.mx/netacad/networks/course/module3/3.3.2.1/3.3.2.1.html#%3A~%3Atext%3DAutenticaci%C3%B3n%20del%20PPP%2Cred%20transmitan%20por%20el%20enlace) [enlace](https://www.itesa.edu.mx/netacad/networks/course/module3/3.3.2.1/3.3.2.1.html#%3A~%3Atext%3DAutenticaci%C3%B3n%20del%20PPP%2Cred%20transmitan%20por%20el%20enlace)

<https://ccnadesdecero.es/cisco-discovery-protocol-cdp/>

# ANEXO 1 - IDEAS INICIALES

## Introducción

El objetivo principal de este documento es mostrar las ideas que hemos tenido en la lluvia de ideas que hemos hecho para elegir el tema de este cuarto PBL.

Primero se mencionan todos los temas que hemos pensado y se explicarán brevemente. Después se valorarán las ideas mediante una tabla de atributos que pueden tener los proyectos. Por último, como hay algunas ideas que hemos planteado más a fondo que otras, se explicarán estas con más detalle que en el primer apartado.

## Lluvia de ideas

### Gestión de banco de alimentos (TidyFood)

Una aplicación para facilitar la entrada y salida de los productos en los bancos de alimentos.

### Gestión de vuelos

Aplicación software para la gestión de reservas de vuelos y gestión del equipaje de los clientes.

### Máquina expendedora

Aplicación para máquinas expendedoras que nos proporciona información extra por ejemplo de cuánta ofrenda/ demanda tiene cada producto.

### Gestión de museos

Aplicación software para la gestión de museos, gestionando tanto las visitas del personal o de los clientes, como el inventario y el estado de cada obra.

### Gestión de residuos médicos

Aplicación para la gestión de residuos médicos, para hospitales incluso farmacias.

## Matriz de ideas

En la siguiente tabla se muestran algunos atributos que pueden o no tener los proyectos y si nuestras ideas cumplen con los atributos.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Idea** | **Reusabilidad** | **Complejidad** | **Utilidad** | **Originalidad** |
| Gestión de banco de alimentos | ✓ | ✘ | ✓ | ✓ |
| Gestión de vuelos | ✘ | ✓ | ✓ | ✘ |
| Máquina expendedora | ✘ | ✘ | ✓ | ✘ |
| Gestión de museos | ✘ | ✓ | ✓ | ✓ |
| Gestión de residuos médicos | ✘ | ✓ | ✓ | ✘ |

## Idea elegida

A continuación, se explicará más a fondo la idea elegida.

### Gestión de banco de alimentos (TidyFood)

Teniendo en cuenta que vamos a seguir con esta idea, haremos una aplicación que registre las entradas y las salidas de todos los productos del banco de alimentos manteniendo así actualizado el inventario.

Crearemos una base de datos para guardar todos los datos tanto de los productos como de los proveedores, empleados, etc.

En cuanto al microprocesador, lo usaremos para registrar las cajas de productos que nos manden los proveedores usando un escáner de código de barras.

Para la salida de productos, la aplicación dispondrá de un apartado para crear lotes y que estos sean destinados a un usuario ya registrado en la aplicación.

# ANEXO 2 – ANEXOS ADS

En este anexo se van a mostrar todos los documentos y diagramas que se han creado para elaborar el análisis y el diseño del software.

## ANEXO 2.1 - ERS

### Introducción

En este documento se especifican los requisitos software de una aplicación de gestión dirigida a los bancos de alimentos. Esta especificación se ha estructurado inspirándose en las directrices dadas por el estándar “IEEE Recommended Practice for Software Requirement Specifications ANSI/IEEE 830 1998”.

#### Propósito

El propósito de este documento es realizar el análisis de requisitos de la aplicación diseñada para facilitar la gestión de los productos en los bancos de alimentos, llamada TidyFood, explicando sus diferentes funcionalidades.

#### Ámbito del sistema

TidyFood es una aplicación que tiene varias funcionalidades relacionadas con la gestión del banco de alimentos. Para empezar el usuario tendrá que iniciar sesión con su usuario y contraseña. Habrá varios tipos de usuario con distintos permisos.

Una vez iniciada la sesión, el usuario tendrá que registrar las donaciones de los proveedores. Para ello, el usuario introducirá el nombre del proveedor, nombre del producto y la fecha de caducidad del producto.

Además de registrar donaciones, también se podrá gestionar el tráfico de los productos del banco de alimentos, el usuario podrá organizar los productos en cajas a la hora de almacenarlos y tendrá que crear distintos tipos de lotes para entregar los productos a los beneficiarios.

Los trabajadores del banco de alimentos podrán también acceder a la información del inventario y gestionar los productos que se encuentran en él. Los productos del inventario se mostrarán por cajas y se podrán ordenar según quiera el usuario, por fecha de caducidad más cercana, por ejemplo. En esa interfaz el usuario podrá acceder al contenido de las cajas y decidir qué hacer con los productos dentro de las cajas.

Toda la información se guardará sobre las donaciones, productos del inventario, etc. Se guardarán en la base de datos y se necesitará conexión con la base de datos para acceder a ella.

#### Definiciones, siglas y abreviaciones

En esta subsección se definirán los términos, acrónimos y abreviaturas utilizadas en el documento:

* **Donación**: Nos referimos a donación a todo tipo de préstamo por parte de los donantes ya sean préstamos de comida tanto económicos.
* **Caja**: Nos referimos a caja a la unidad de almacenamiento utilizada en el banco de alimentos.
* **Lote**: Nos referimos a lote a una unidad que contiene varios productos listos para la entrega a los beneficiarios.
* **Beneficiario**: Un beneficiario es cada una de las personas, comedores sociales, etc. que reciben lotes de alimentos.

#### Referencias

En esta subsección se mostrará una lista completa de todos los documentos referenciados en la ERS:

- ANSI/IEEE Std. 830-1984 Guía del IEEE para la Especificación de Requerimientos Software.

#### Resumen global

Este documento va a constar de dos apartados: la descripción del producto y los requisitos que este requiere.

Respecto a la descripción del producto, se mostrará las funciones que la aplicación tiene, se definirán las características de los usuarios y las restricciones para los desarrolladores. En cuanto a los requisitos, se especificarán todas las acciones que requerirán de software.

### Descripción general

En esta sección se describen todos aquellos factores que afectan al producto y a sus requisitos. No se describen los requisitos, sino su contexto. Esto permitirá definir con detalle los requisitos en la sección tres, haciendo que sean más fáciles de entender.

#### Perspectiva del producto

TidyFood consta de un único módulo principal independiente: esta será la aplicación desarrollada. En él, toda la información puede ser mostrada, modificada y añadida mediante esta. TidyFood ayuda a gestionar los productos que pasan por los bancos de alimentos, dando la posibilidad de crear cajas para almacenar los productos y crear lotes para entregar estos a los beneficiarios.

La aplicación guarda toda la información registrada en una base de datos que se encuentra en la sede central de TidyFood para tener acceso a todos los datos actualizados en todo momento.

TidyFood consta con las siguientes funcionalidades:

* Registro de donaciones
* Administración del inventario
* Creación de cajas
* Creación de lotes

#### Funciones del producto

En esta subsección se describirán las funciones que cumple el producto.

Para empezar, desde la aplicación se podrán crear cajas para almacenar los productos en el inventario. Saber en qué caja se encuentra cada producto podría ser clave a la hora de localizar los productos para evaluar la calidad de estos y saber de qué donantes vienen, por ejemplo.

Otra de las funciones que tiene nuestra aplicación es calcular y mostrar gráficamente la cantidad de productos que entran en el banco de alimentos comparando estas cantidades con las cantidades que salen del banco de alimentos.

Además de las anteriores funciones, el usuario podrá acceder mediante un botón al inventario y aparte de ver lo que hay en el inventario deshacerse de los productos que están a punto de caducar.

En esa interfaz, se podrán filtrar los productos según quiera el usuario ya sea por tipo de producto, grupo de productos, fecha de caducidad más previa, etc.

A la hora de entregar las cajas, nuestra aplicación permite sacar los productos de las cajas en las que se encuentran almacenadas y meter estos productos en lotes para que haya variedad de lotes. Así los beneficiarios tendrán opción a elegir a la hora de recoger la comida. Se ha implementado esto para que aquellos beneficiarios que sean intolerantes a ciertos tipos de alimentos puedan escoger.

#### Características de los usuarios

Los usuarios que utilizarán este sistema serán los trabajadores del propio banco de alimentos. Entre los usuarios se distinguen dos tipos.

Habrá un usuario para cada uno de los trabajadores. Estos usuarios podrán utilizar prácticamente todas las funcionalidades.

El siguiente tipo de usuario, además de tener los privilegios del usuario del nivel anterior también podrá gestionar los demás usuarios. Es decir, podrá añadir usuarios a la base de datos dándoles así permiso para entrar en la aplicación y también podrá eliminar usuarios de la base de datos causando el efecto contrario.

#### Restricciones

Cómo sigue la política de la empresa el software se desarrollará en el lenguaje de programación Java. Un lenguaje de programación orientado a objetos que permite una gran flexibilidad, mientras que el motor de la base de datos será mySQL.

Es necesario permitir al usuario llegar a la información que necesita de forma rápida y sencilla, por lo que a la hora de buscar datos se hará a base de menús desplegables. Los dos tipos de usuarios tendrán la misma pantalla principal pero los usuarios administradores tendrán acceso a más botones que los demás usuarios.

La seguridad de los datos de la aplicación es un punto importante a la hora de desarrollar la aplicación. Puesto que los datos que se van a mostrar en la aplicación son totalmente confidenciales. Para asegurar la confidencialidad se seguirán las normas de seguridad indicadas en el estándar de seguridad de la información IEC 60950-1.

#### Suposiciones y dependencias

Para utilizar TidyFood sólo será necesario tener la aplicación instalada en un ordenador. Además, para los trabajadores serán necesarias unas credenciales oficiales distribuidas a los trabajadores dependiendo del puesto de trabajo que tengan. Esto también dará la opción de dar diferentes permisos dependiendo del puesto de trabajo. Estas credenciales constaran de un usuario y una contraseña para cada trabajador.

La aplicación se desarrollará enfocada al sistema operativo Windows. Por lo que los requisitos definidos para el sistema asumen que el banco de alimentos utilizará el sistema operativo Windows y no cambiará dichos sistemas.

### Requisitos específicos

En este apartado se definirán los requisitos específicos para poder desarrollar todas las funcionalidades. Los requisitos aquí recogidos son imprescindibles y se han especificado de manera que sea simple comprobar si el sistema los ofrece adecuadamente.

#### Requisitos comunes de las interfaces

En este apartado se definen los requisitos comunes que comparten las interfaces de la aplicación.

##### Interfaces de usuario

Las interfaces de usuario están relacionadas con las pantallas, son las ventanas que el usuario debe manipular para realizar una operación específica.

Las interfaces de usuario incluirán los siguientes componentes:

* Botones
* Menús desplegables
* Mensajes informativos
* Mensajes de error
* Cuadros de diálogo
* Formularios para añadir, modificar o eliminar datos.
* Listas

##### Interfaces de hardware

* **Pantalla**: se mostrará toda la información relacionada con la aplicación a través de la pantalla.
* **Ratón**: mediante el ratón se interactúa con la aplicación, y el software deberá responder a los clicks realizados con los botones del ratón.
* **Teclado**: el software deberá responder en los formularios a las teclas pulsadas.

#### Requisitos funcionales

Los requisitos funcionales de la aplicación se dividen en varias secciones:

##### Validar usuarios

|  |  |
| --- | --- |
| Introducción | El sistema debe permitir el ingreso del nombre y contraseña del usuario para realizar las diferentes funciones que tendrá cada uno. |
| Proceso | Al acceder a la aplicación, el sistema te pedirá introducir tu nombre de usuario y contraseña. Una vez metidos le dará al botón de entrar, y en caso de introducirlos correctamente el usuario pasará al menú y tendrá acceso a las funciones correspondientes. |

##### Registrar usuarios

|  |  |
| --- | --- |
| Introducción | El sistema permite registrar usuarios a los usuarios administradores. |
| Proceso | Los usuarios administradores tendrán la posibilidad de acceder a una interfaz donde introduciendo las credenciales del usuario y el tipo de usuario pueden registrar un nuevo usuario. |

##### Gestión del inventario

|  |  |
| --- | --- |
| Introducción | El sistema permite gestionar los alimentos que se encuentran en el inventario del banco de alimentos. |
| Proceso | El usuario puede acceder al inventario para ver el stock del banco de alimentos y mediante un botón retirar los productos que estén cerca de caducar. |

##### Registro de donaciones

|  |  |
| --- | --- |
| Introducción | El sistema permite registrar las donaciones de los proveedores. |
| Proceso | Para registrar una donación habrá que indicar qué proveedor ha sido el donante, que alimentos ha donado y cuál es su fecha de caducidad. |

##### Gestión de entregas

|  |  |
| --- | --- |
| Introducción | El sistema permite crear lotes para luego entregar estos a los beneficiarios. |
| Proceso | Los usuarios podrán crear diferentes tipos de lotes pasando los productos |

#### Requisitos no funcionales

Se consideran requisitos no funcionales las limitaciones en servicios y funciones ofrecidas por el sistema. Estos requisitos no cumplen una relación directa con la funcionalidad del sistema y son impuestos por el cliente. Se analizarán a continuación en tres apartados: requisitos del producto, requisitos de organización y requisitos externos.

##### Requisitos del producto

En esta subsección se desarrollarán los requisitos que especifican el comportamiento que debe tener el sistema.

* **Usabilidad**: La aplicación se puede utilizar tanto en dispositivos móviles como en dispositivos Windows de sobremesa para los trabajadores que tienen un puesto fijo. La operabilidad está basada en menús así consiguiendo una facilidad en el uso de ella. El tiempo de respuesta a cualquier operación ha de ser menor de 10 segundos.
* **Fiabilidad**: La aplicación debe estar disponible en todo momento por lo que deberá ser a prueba de fallas (caída de un servidor, caída del servicio…).
* **Eficiencia**: La aplicación ha de ser eficiente para poder tener una buena respuesta en tiempo real. La estructura de datos para el almacenamiento de los datos será sencilla para que el sistema no utilice más recursos de los debidos.
* **Portabilidad**: La aplicación deberá funcionar en el sistema operativo Windows. Se ejecutará tanto en pantallas táctiles como en monitores de dispositivos con este sistema operativo.

##### Requisitos organizacionales

Los requisitos organizacionales exigidos por el cliente son la utilización de los servidores en Windows y la implementación de MySQL para la creación y gestión de la base de datos.

##### Requisitos tecnológicos

Para el correcto funcionamiento de la aplicación en Windows se exige una configuración mínima.

Respecto al programa que se entrega, se entregará un ejecutable capaz de conectarse con el servidor para poder recibir los datos de la base de datos.

## ANEXO 2.2 - Casos de uso y escenarios

### Diagrama Descripción generada automáticamenteDiagrama de casos de uso total

Ilustración 26 Caso de uso total

El diagrama de casos de uso total recoge todos los diagramas de casos de uso y todos los actores que intervienen en la aplicación. Todos los actores tienen acceso a los casos de uso general aparte del diagrama de casos de uso asociado a este. Todos los casos de uso tienen como precondición que el actor esté validado previamente.

### Diagrama Descripción generada automáticamenteDiagrama de casos de uso general

Ilustración 27 Caso de uso general

|  |  |
| --- | --- |
| Nombre | CU1 Ver inventario |
| Descripción | El usuario puede ver los productos que se encuentran almacenados en el inventario también dispondrá de información sobre estos. |
| Actores | Cualquier trabajador |
| Precondiciones | Que el actor esté registrado en la aplicación |
| Flujo normal | * Abrir la aplicación * Introducir nombre y contraseña * Seleccionar el botón “Inventario” |
| Flujo alternativo | Si los datos introducidos no son correctos salta un mensaje y se permite volver a introducirlos correctamente |
| Postcondiciones | El usuario accederá al inventario |

### Diagrama Descripción generada automáticamenteDiagrama de casos de uso de recibos

Ilustración 28 Caso de uso recibidos

|  |  |
| --- | --- |
| Nombre | CU2.1 Iniciar sesión |
| Descripción | El usuario debe introducir sus credenciales para entrar a la aplicación. |
| Actores | Gestor de recibos |
| Precondiciones | Que el actor esté registrado en la aplicación |
| Flujo normal | * Abrir la aplicación * Introducir credenciales * Dar botón de aceptar * Extiende caso de uso **CU2.2 Registrar donaciones** * Extiende caso de uso **CU2.3 Crear cajas** |
| Flujo alternativo | Si los datos introducidos no son correctos salta un mensaje y se permite volver a introducirlos correctamente |
| Postcondiciones | El usuario accede a la aplicación. |

|  |  |
| --- | --- |
| Nombre | CU2.2 Registrar donaciones |
| Descripción | El usuario puede registrar las donaciones una vez estas lleguen al banco de alimentos. |
| Actores | Gestor de recibos |
| Precondiciones | Un proveedor dona comida al banco de alimentos y la donación se encuentra en el banco |
| Flujo normal | * Rellenar los datos de la donación * Dar al botón de registrar |
| Flujo alternativo | Si falta algún dato por rellenar salta un mensaje de error |
| Postcondiciones | La donación queda registrada |

|  |  |
| --- | --- |
| Nombre | CU2.3 Crear cajas |
| Descripción | El usuario puede crear cajas para tener más trazabilidad de los productos ya registrados. |
| Actores | Gestor de recibos |
| Precondiciones | Un proveedor dona comida al banco de alimentos |
| Flujo normal | * Rellenar los datos de la donación * Dar al botón de registrar |
| Flujo alternativo | Si los datos introducidos no son correctos salta un mensaje y se permite volver a introducirlos correctamente |
| Postcondiciones | La caja es creada con los productos dentro |

### Diagrama de casos de uso de inventario

### 

Ilustración 29 Caso de uso inventario

|  |  |
| --- | --- |
| Nombre | CU3.1 Iniciar sesión |
| Descripción | El usuario debe introducir sus credenciales para entrar a la aplicación. |
| Actores | Gestor de inventario |
| Precondiciones | Que el actor esté registrado en la aplicación |
| Flujo normal | * Abrir la aplicación * Introducir credenciales * Dar botón de aceptar * Extiende caso de uso **CU3.2 Añadir productos** * Extiende caso de uso **CU3.3 Retirar productos** |
| Flujo alternativo | Si los datos introducidos no son correctos salta un mensaje y se permite volver a introducirlos correctamente |
| Postcondiciones | El usuario accede a la aplicación. |

|  |  |
| --- | --- |
| Nombre | CU3.2 Añadir productos |
| Descripción | El usuario puede añadir productos directamente al inventario. Se utilizará en caso de que la cantidad puesta a la hora del registro sea incorrecta. |
| Actores | Gestor de inventario |
| Precondiciones | Ha habido algún error al registrar la cantidad de productos que había en la donación y se quiere corregir. |
| Flujo normal | * Entrar al inventario a través del botón “Inventario” * Dar al botón “Añadir producto” * Rellenar los datos necesarios * Dar al botón “Aceptar” |
| Flujo alternativo | Si alguno de los campos no es rellenado salta un mensaje de error. |
| Postcondiciones | Se añade la cantidad indicada del producto indicado. |

|  |  |
| --- | --- |
| Nombre | CU3.3 Retirar productos |
| Descripción | El usuario puede retirar la cantidad de productos que quiera del inventario |
| Actores | Gestor del inventario |
| Precondiciones | Algún producto a caducado o se encuentra en mal estado y se quiere retirar el producto del inventario |
| Flujo normal | * Entrar al inventario a través del botón “Inventario” * Dar al botón “Retirar producto” * Rellenar los datos necesarios * Dar al botón “Aceptar” |
| Flujo alternativo | Si alguno de los campos no es rellenado salta un mensaje de error. |
| Postcondiciones | El producto es retirado del inventario. |

### Diagrama de casos de uso de entregas

Ilustración 30 Caso de uso entregas

|  |  |
| --- | --- |
| Nombre | CU4.1 Iniciar sesión |
| Descripción | El usuario debe introducir sus credenciales para entrar a la aplicación. |
| Actores | Gestor de entregas |
| Precondiciones | Que el actor esté registrado en la aplicación |
| Flujo normal | * Abrir la aplicación * Introducir credenciales * Dar botón de aceptar   Extiende caso de uso **CU4.2 Crear lotes**  Extiende caso de uso **CU4.3 Validar lotes**  Extiende caso de uso **CU4.4 Administrar lotes** |
| Flujo alternativo | Si los datos introducidos no son correctos salta un mensaje y se permite volver a introducirlos correctamente |
| Postcondiciones | El usuario accede a la aplicación |

|  |  |
| --- | --- |
| Nombre | CU4.2 Crear lotes |
| Descripción | El usuario puede crear lotes utilizando los productos del inventario |
| Actores | Gestor de entregas |
| Precondiciones | Hay productos registrados en el inventario y hay que consumirlos antes de que caduquen. |
| Flujo normal | * Seleccionar el botón “Crear Lotes” * Elegir los productos que se quieren sacar del inventario para meter estos en lotes * Elegir la cantidad de cada tipo de lote * Dar al botón “Entregar” |
| Flujo alternativo | No hay suficientes productos para crear la cantidad deseada de lotes. |
| Postcondiciones | Los lotes son creados y crean pendientes de validar. |

|  |  |
| --- | --- |
| Nombre | CU4.3 Validar lotes |
| Descripción | El usuario puede validar los lotes ya creados. |
| Actores | Gestor de entregas |
| Precondiciones | Se han creado lotes previamente. |
| Flujo normal | * Verificar que cada lote tiene los productos deseados * Dar al botón de “Aceptar” |
| Flujo alternativo | Alguno de los lotes no tiene los productos deseados por lo cual el usuario modifica el contenido de los lotes. |
| Postcondiciones | El lote queda verificado. |

### Diagrama de casos de uso de gestión de usuarios

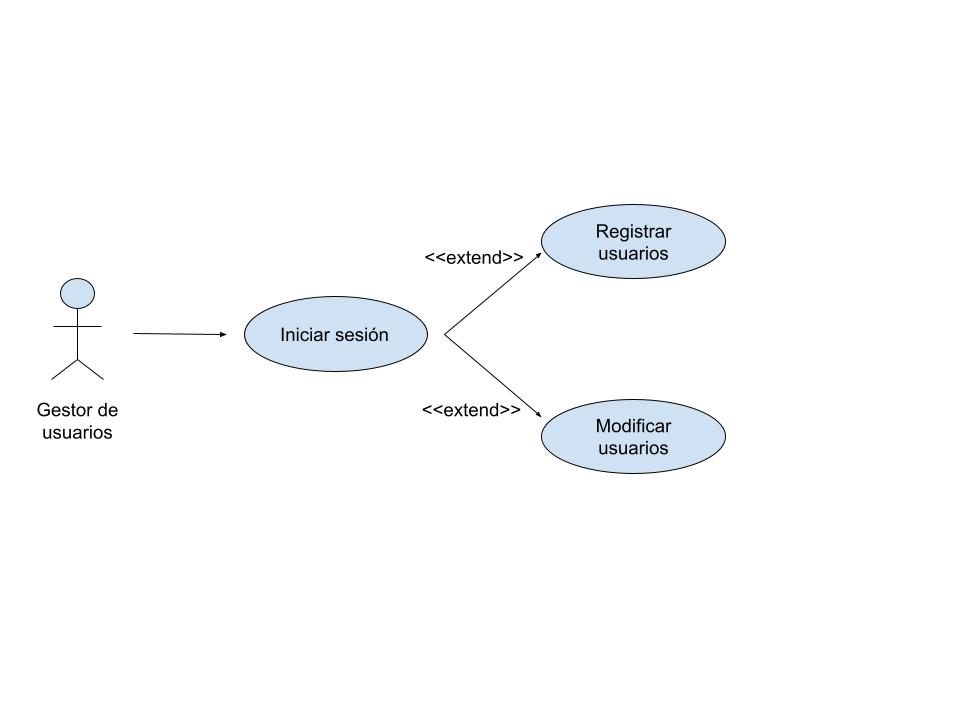


Ilustración 31 Caso de uso usuarios

|  |  |
| --- | --- |
| Nombre | CU5.1 Iniciar sesión como administrador |
| Descripción | El usuario debe introducir sus credenciales para entrar a la aplicación. |
| Actores | Gestor de usuarios |
| Precondiciones | El usuario debe estar registrado en la aplicación. |
| Flujo normal | * Abrir la aplicación * Introducir credenciales * Dar al botón “Aceptar” * Extiende caso de uso **CU5.2 Registrar usuarios** * Extiende caso de uso **CU5.3 Modificar usuarios** |
| Flujo alternativo | Si las credenciales son incorrectas salta un mensaje de error |
| Postcondiciones | El usuario accede a la aplicación |

|  |  |
| --- | --- |
| Nombre | CU5.2 Registrar usuarios |
| Descripción | El usuario administrador puede registrar nuevos usuarios. |
| Actores | Gestor de usuarios |
| Precondiciones | Se tiene que registrar un nuevo usuario al sistema. |
| Flujo normal | * Seleccionar botón “Usuarios” * Introducir credenciales * Seleccionar botón “Aceptar” |
| Flujo alternativo | Ninguno |
| Postcondiciones | El usuario queda registrado en el sistema |

|  |  |
| --- | --- |
| Nombre | CU5.3 Modificar usuarios |
| Descripción | El usuario administrador puede modificar las credenciales de los demás usuarios. |
| Actores | Gestor de usuarios |
| Precondiciones | El usuario que se quiere modificar debe estar registrado en el sistema |
| Flujo normal | * Seleccionar botón de “Usuarios” * Seleccionar usuario a modificar * Seleccionar botón de “Modificar” * Modificar credenciales * Seleccionar botón de “Aceptar” |
| Flujo alternativo | Ninguno |
| Postcondiciones | Las credenciales del usuario quedan modificadas. |

## ANEXO 2.3 - Diagrama de estados

Es este apartado se mostrará y explicarán los diferentes estados por el que pasa un producto desde el momento que llega al banco de alimentos hasta que el producto es entregado o desechado en caso de caducarse antes de la entrega. Para ello se ha elaborado un diagrama de estados.

El primer paso al recibir una donación es clasificar los productos. En este caso se clasifican los productos por tipos de producto y fecha de caducidad.

Una vez que se clasifican los productos hay que registrar los productos usando la aplicación para que los productos donados queden registrados en la base de datos. Habrá que rellenar los campos necesarios para el registro del producto en la pantalla inicial de la aplicación.

Después de registrar los productos hay que meter los productos registrados en cajas para almacenarlos. Existe una pequeña posibilidad de que el producto caduque antes de ser almacenado en una caja, en ese caso el producto sería desechado y retirado de la base de datos. El proceso de registrar y almacenar los productos en cajas es casi continuo por lo cual no debería de ocurrir.

Las cajas tienen una fecha de caducidad equivalente a la fecha de caducidad más cercana entre todos los productos que se encuentran en la caja. Si la caja caduca se revisa el estado de cada producto. En caso de que el producto esté caducado el producto sería desechado y retirado de la base de datos. Por otro lado, si el producto aún no está caducado volverá a estar en estado registrado.

En caso de que el producto siga el flujo normal, después de ser almacenado en una caja, a la hora de crear lotes se utilizará ese producto y después de crear el lote será entregado a algún beneficiario.

Gráfico, Gráfico de cajas y bigotes

Descripción generada automáticamente

## ANEXO 2.4 - Diagramas de actividad

Para cada funcionalidad implementada se ha creado un diagrama de actividad que describe que pasos hay que seguir para cumplir con la funcionalidad.

Cada uno de los miembros del equipo ha implementado una nueva funcionalidad a la aplicación y estas funcionalidades serán descritas en el siguiente apartado.

### Inicio de sesión

Se ha implementado una funcionalidad para que antes de entrar a la pantalla principal el usuario tenga que iniciar sesión en la aplicación.

Diagrama

Descripción generada automáticamenteEn el momento de ejecutar la aplicación sale una ventana que bloquea la pantalla principal hasta que el usuario inicia la sesión. El proceso es muy simple, el usuario tiene que introducir sus credenciales y pulsar el botón de aceptar. En caso de que las credenciales sean correctas el usuario accederá a la pantalla principal. En caso contrario, saltará una alerta advirtiendo de que las credenciales son incorrectas.

### Registro de productos

Se ha creado esta funcionalidad con el fin de llevar un registro minucioso sobre lo alimentos que entra y los proveedores que nos los dan.

Como se puede observar en el diagrama, al recoger un producto, la aplicación te da 2 opciones para el registro de este, el registro a mano mediante un formulario en la interfaz del sistema y mediante el escáner QR. Al ser es registro a mano una opción infalible a errores, se terminará el proceso al pulsar el botón de “Registrar”. Si embargo, en la opción QR, la más rápida a priori, puede tener errores debido al mal estado del código o al estado del escáner. Por lo tanto, si da error deberá meter los datos a mano, en cambio si no da error, de manera rápida y sencilla el proceso de registro se habrá terminado.

Diagrama

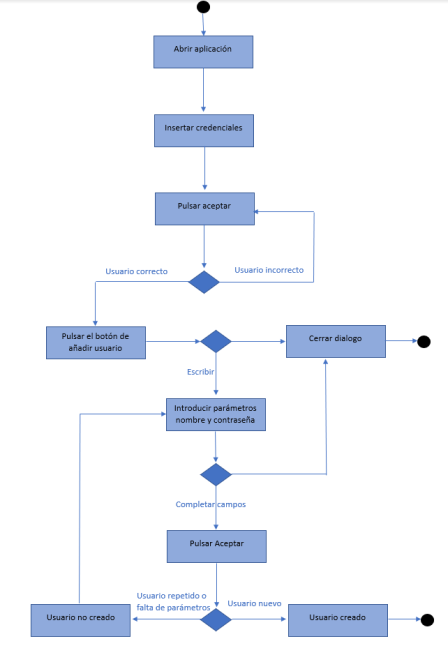
Descripción generada automáticamente

Ilustración 32 DiagramaAct Registro de producto

### Creación de nuevos usuarios

Para la crear un nuevo usuario primero hay que iniciar sesión en la aplicación con un usuario administrador. Después de iniciar sesión, como anteriormente se ha explicado, el usuario deberá pulsar el botón añadir usuario e insertar las credenciales del nuevo usuario haciendo así que el usuario quede registrado en la base de datos. En caso de que el usuario ya exista saltará un aviso y se tendrán que introducir de nuevo las credenciales.

Ilustración 33 DiagramaAct Crear usuarios



## ANEXO 2.5 – Diagramas de clases

En el siguiente apartado se presentarán los diagramas de clase que hemos diseñado para tener otra visión de las funcionalidades teniendo en cuenta las clases que utilizaremos en la aplicación.

### Registro de productos

Para el registro de productos lo primero **se utilizará será la clase producto, el cual contiene las propiedades básicas de los ítems. Los ítems son la unidad en la que se contabilizan los productos. Para obtener los datos, el usuario deberá introducir los datos en los diferentes JTextBox-es de la pantalla principal y mediante el controlador de la pantalla principal, se ejecutarán los comando insert para almacenar los datos en la BD mediante la conexión establecida gracias a “miconnection”.**

Diagrama

Descripción generada automáticamente

Ilustración 34 DiagramaClases Registro de productos

### Inicio de sesión

Texto, Carta

Descripción generada automáticamentePara cumplir con la funcionalidad de iniciar sesión primero el programa utiliza la clase VistaInicioSesión donde se crea la ventana de inicio de sesión. Para verificar que las credenciales son correctas se llama a la clase ModeloInicioSesión que utilizando funciones de la clase Select consigue la lista de usuarios y va comparando uno a uno a ver si hay algún usuario con el nombre de usuario y la contraseña que ha puesto.

Ilustración 35 DiagramaClases Login

### Creación de usuarios

Tabla

Descripción generada automáticamentePara la creación de usuarios se utilizan solamente la clase usuario y la clase vista principal. Una vez escritas las credenciales del usuario, el usuario se guarda en la base de datos.

Ilustración 36 DiagramaClases Creación de usuarios

# ANEXO 3 - BASE DE DATOS

Diagrama

Descripción generada automáticamentePrimero de todo, y lo que básicamente abarca todo el diagrama, es la parte del modelo de nuestra aplicación. Además de eso, incluye una columna sin estar unido a nada, pero por eso no es menos importante en el funcionamiento del programa, ya que incluye el campo donde se guardan los usuarios de la aplicación y sus contraseñas.

Ilustración 37 BD Modelo

Por un lado, como ya se ha mencionado, estaría la parte del funcionamiento de la aplicación, el cual está compuesta de la información necesaria de los proveedores, productos, tipo de alimentos, intolerancias, cajas, lotes y beneficiarios Si se observa la imagen que está unas líneas más abajo,,que es la de una parte del modelo relacional, se puede ver que a la izquierda del todo, tenemos una tabla llamada proveedores, donde se guarda un número de identificador único, además del nombre del proveedor y la dirección de este.

A su derecha, se puede observar la tabla Producto. Esta guarda como campos primarios, tanto la fecha de caducidad, como el identificador del producto. Gracias a esto, es posible diferenciar entre unos yogures que caducan mañana y otros que caducan la semana que viene. El mismo día, en cambio, sí que se pueden meter diferentes artículos. Además de eso, la tabla guarda el nombre del del producto introducido, la cantidad de unidades introducidas, la fecha de donación, un campo donde se puede verificar si la fecha de caducidad es anterior a la fecha actual, y finalmente, el proveedor del producto, haciendo esta referencia a un proveedor que ya tiene que estar introducido anteriormente.

Abajo a la izquierda, hay una tabla llamada Grupo alimentos, en donde se dispone de un identificador para el tipo de grupo, y un nombre para el grupo.

Para unir tanto el producto, como el grupo, hay una tabla, la cual se le ha puesto como nombre pertenece. Esta tabla, se encarga de unir tanto los productos como los grupos, ya que, un producto puede pertenecer tanto a un grupo, como a varios grupos de alimentos. Y los grupos, están creados tanto por un solo producto como varios productos.

Ilustración 38 BD Tablas

Diagrama

Descripción generada automáticamenteLa siguiente parte por explicar, es la que Une tanto Lotes, Caja e Intolerancias con el producto

Empezando por abajo a la derecha, se dispone de una lista con diferentes intolerancias que se pueden tener a productos, guardando tanto un identificador, como el nombre de estas.

Para unir tanto el producto como las intolerancias, hay una tabla intermedia en el que se trata de unir tanto uno o varios productos, con una intolerancia en cuestión, y decir si un producto puede contener una o varias intolerancias.

Por el medio, hay una tabla llamada caja. Aquí cada caja, guarda con un identificador propio, el ID para cada caja que se haga. Además de eso, la caja tiene una fecha de caducidad propia, que consta al final del producto que antes se caduca. Poniendo el ejemplo de antes, si se dispone de unos yogures con fecha de caducidad de mañana, y otros con la fecha de la semana que viene, al tener los yogures de mañana, una fecha de caducidad más cercana será la que se quede implementada. Además de eso, la tabla incluye información necesaria con la cantidad de productos y de cajas que se disponen.

Para unir la caja con el producto se ha incluido la tabla ProductoACaja, el cual se encarga de unir qué productos van de cada caja, y que cantidad de productos se incluyen en esta.

Finalmente, arriba está el campo lote, el cual, como todos, cada lote se diferencia con un identificador diferente, y tiene incluidos tanto el nombre que se pone al lote, la fecha de entrega de este, y el identificador de un beneficiario, que será mencionado más adelante.

Diagrama

Descripción generada automáticamentePara juntar Lotes y productos, hay una tabla llamada Contiene, el cual está compuesto por las claves primarias de las dos tablas que se buscan unir, y tenemos otro campo llamado tabla donde se guarda la cantidad de productos hay en un lote.

Ilustración 39 BD Modelo 1

Finalmente, la base de datos está compuesta por dos últimas tablas más. La primera de ellas es la ya mencionada Beneficiarios, la cual se encarga de guardar el nombre y el ya habitual identificador. Este se encarga de hacer que los lotes tengan a alguien al que llegar.

Para acabar, la tabla llamada Relación, se encarga de describir qué beneficiarios tienen intolerancias, y en caso de tenerlas a qué tienen intolerancias, para eso, se han unido los identificadores de tanto los beneficiarios como los de las intolerancias.

Ilustración 40 BD Tablas 1

Diagrama

Descripción generada automáticamente

A continuación, después de haberla desglosado y explicado parte por parte, se incluye el Modelo Relacional completo de la base de datos. No se ha hablado acerca de la tabla Usuario, porque su funcionamiento ya ha sido explicado cuando se hablaba del Modelo Entidad Relación.

Diagrama

Descripción generada automáticamente

Ilustración 41 BD Modelo entero

## Consultas MySQL

Una vez se obtiene un Modelo Entidad Relación bien estructurado, se pasa a crear las tablas en cuestión haciendo uso del DDL (Data Definition Language) y tras crearlas, se insertan los datos y valores necesarios para poner en funcionamiento la aplicación gracias al uso de manipulación de datos llamado DML (Data manipulation Language).

### Creación de tablas

Imagen que contiene Tabla

Descripción generada automáticamenteA la hora de la creación de tablas, se han aplicado diferentes tipos de variables, que son INTEGER para los números, VARCHAR(X) siendo X una cifra variable, para las cadenas de caracteres, y finalmente DATE, para introducir fechas. También ha habido algún campo del cual se ha precisado que sea NOT NULL porque es necesaria la información. A continuación, se mostrarán un par de tablas para ver cómo hemos introducido estos datos.

Ilustración 42 Creación de tablas 1

Además de los tipos de variables mencionados anteriormente, la tabla producto incluye una columna extra que no es creada junto a la base de datos, sino después, el cual consiste en un valor booleano (es decir, si o no) el cual va a ser el resultado de si una fecha de caducidad, supera al día actual, lo que hace que el valor se ponga en 1, afirmando que el producto está caducado.

Ilustración 43 Creación de tablas 2

Finalmente, la base de datos también incluye ciertas restricciones, también conocidas como Constraints, las cuales son utilizadas para definir las claves primarias y foráneas de las tablas. A continuación, se puede ver como son incluidas algunas en nuestra base de datos:

Texto

Descripción generada automáticamente con confianza media

Ilustración 44 Creación de tablas 3

### Motores de almacenamiento

La base de datos utiliza todo el tiempo el motor de almacenamiento InnoDB, ya que, aunque sea más lenta, esta permite hacer referencias a otra tabla mediante el uso de las claves foráneas. InnoDB es la que viene por defecto a la hora de utilizar el sistema de gestión de bases de datos relacional MySQL.

Texto, Word

Descripción generada automáticamente con confianza mediaEso no quiere decir, que no se haya utilizado ningún otro motor de almacenamiento, ya que, como la tabla de Usuarios, es una tabla única, sin estar unida a ninguna otra, para hacerla más rápida, se ha decidido asignarle el motor de almacenamiento MYISAM. Para asignarle, ha bastado con poner al final de la creación de la tabla el motor a utilizar como se puede apreciar en la foto:

Ilustración 45 Motores de almacenamiento

### Usuarios

Aunque la aplicación utilice diferentes usuarios como se ha hablado hasta ahora, con la tabla “Usuarios”, a la hora de manipular la base de datos, disponemos de dos usuarios, que son el del Administrador, y el del usuario común.

El administrador tiene control total en el servidor principal, mientras que el usuario, solo tiene acceso a la base de datos, y no puede realizar más allá de las funciones de SELECT, UPDATE, INSERT, DELETE.

Tal y como se puede apreciar, ambos usuarios tienen permisos de acceder a la base de datos desde cualquier IP. Esto se puede apreciar ya que después de @, la red es %.

Además de eso, aunque ahora ponga que la contraseña es ‘a’, en un futuro no muy lejano, a la hora de sacar el producto al mercado, estas contraseñas serán cambiadas, ya que ahora, para una más rápida programación, se decidió dejarlas cortas, porque nadie más a parte del equipo de desarrollo sabe ni que IP, ni donde está ubicado el servidor.

Ilustración 46 Usuarios BD

Texto, Carta

Descripción generada automáticamente

## Seguridad

En el apartado de seguridad, se tomarán en cuenta dos puntos. Primero, se hablará acerca de la conexión desde dos servidores distintos, y acto seguido, se hablará de la realización de copias de seguridad.

### Replicación

Nunca se sabe cuándo puede ocurrir un desastre natural como una inundación a causa de lluvias torrenciales, un incendio, un corte de luz o simplemente que la conexión de internet del servidor falle. Al final, si pasa algo de las cosas mencionadas, significa que es imposible acceder a la base de datos, por lo que para prevenir eso, nuestra infraestructura cuenta tanto con un servidor primario, como con otro servidor secundario.

Para eso, contamos con un ordenador Windows 10, cuya IP siempre es fija, y es 172.17.100.70. El servidor, normalmente está siempre en marcha, y gracias al binlog, y a la IP, el servidor secundario, lo recoge y lo monta para su uso propio. Para el servidor secundario, al principio se decidió utilizar otro ordenador Windows 10, pero por temas de tareas automatizadas, se decidió descartar esta opción, y se decidió apostar por utilizar un equipo con una distribución de Linux como sistema operativo, y el elegido fue Ubuntu.

### Copias de seguridad

Respecto a las copias de seguridad, se decidió crear una tarea automatizada, para que todas las noches, a las 00:30 de la noche, se creen copias de seguridad.

Esta fue la razón por la que en el apartado anterior se menciona que se decidió hacer un cambio de sistema operativo, ya que crontab, es una herramienta eficiente y muy fácil, para automatizar tareas, y a su vez hacer copias de seguridad automatizadas.

Para realizar las copias de seguridad, se utiliza la herramienta de mysqldump, el cual se ejecuta 4 veces. Primero realiza una copia de seguridad de la base de datos entera, tras eso, realiza una copia de la base de datos que incluye el comando de creación de la BD.

Tras eso, se encarga de realizar una copia de la estructura de la BD sin incluir los datos, y finalmente, realiza una copia de los datos de la BD, incluyendo solo los datos.

Texto

Descripción generada automáticamenteLos 4 comandos mostrados, se guardan en un fichero con la extensión .sh, y crontab, se encarga a la hora ya mencionada, ejecutar el script, y cada día la base de datos, está actualizada con las cosas actualizadas desde el día anterior.

Ilustración 47 Seguridad BD

## Conexión de la base de datos con la aplicación

La base de datos se conecta con la aplicación mediante el uso de procedimientos almacenados.

Primero de todo, en MySQL, se elige la base de datos a utilizar. Tras eso, se cambia el delimitador, para que el procedimiento se ejecute completamente, ya que, si no, el programa solo leería hasta el punto y coma (;) y ni iniciaría ni acabaría el procedimiento bien.

Después, se crea el procedimiento, y si se desea, se definen parámetros a utilizar en la base de datos y se da a empezar al Gracias al begin.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamenteAllí, va a realizar todo lo que le pongas, como si son, insertar, actualizar o borrar tablas o datos, crear variables internas etc., y tras el END, se terminará el procedimiento. Finalmente, se vuelve a poner el delimitador, que por defecto es el punto y coma.

Ilustración 48 Conexión BD - JAVA 1

Para llamar al proceso, se utiliza el comando call, y esta puede ser definida tanto en el entorno de desarrollo de la base de datos, como se puede observar en la foto anterior, como en el entorno de desarrollo del programa.

Nuestro programa, ha sido realizado en java, y para eso, ha hecho falta integrar la librería del MySQL-J Connector, e implementarlo en el programa.

Imagen que contiene Texto

Descripción generada automáticamentePara iniciar la conexión, se mete dentro de un método, el nombre de la base de datos, si es local o remoto (en la foto se muestran ambas diferencias), y se mete el nombre de usuario y contraseña a utilizar.

Ilustración 49 Conexión BD - JAVA 2

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamenteSi todo va bien, la conexión se realizará y podrás hacer diferentes llamadas a diferentes procedimientos, como se puede observar a continuación, esta llamada estaría realizada desde JAVA a diferencia de la de MySQL, que es la que se ha mostrado anteriormente.

Ilustración 50 Conexión BD - JAVA 3

En la foto, tal y como se puede apreciar, también realiza un comando call. Una vez llamado, los resultados se guardan en una clase llamada ResultSet, y para mostrarlos, se recorren una y otra vez la fila RS, hasta que esté vacía, y dentro de ella, se indican numéricamente qué columnas se quieren mostrar.

# ANEXO 4 - REDES

En los siguientes apartados se van a explicar diferentes protocolos de red.

## PAT

Cuando los paquetes de los hosts internos salen del enrutador configurado con un PAT a Internet, se traducen a una dirección IPv4 del grupo PAT. Por lo que habrá que verificar si la IP origen del paquete ha cambiado por una pública. Para saber si esta conversión ha tenido lugar habrá que escribir el comando **show ip nat statistics**.

En este comando se detalla la información sobre la cantidad y el tipo de traducciones activas, los parámetros de configuración NAT, la cantidad de direcciones en el pool y la cantidad que se asignó. Pero a lo que hay que mostrar atención será a los “Hits” de las interfaces. El número que aparezca será el indicador del total de direcciones IP que han sido traducidas.

Si al hacer ping fuera de la red no se suma este contador, eso quiere decir que PAT no está funcionando correctamente. Una causa puede ser que los pools de direcciones IP están mal configuradas. Para comprobar los pools de PAT habrá que usar el comando**show ip nat translations**. Si alguna de los pools tiene direcciones IP incorrectas habría que borrar los pools y volver a configurarlas con los siguientes comandos:

* **access-list 1 permit *“IP red” “Wildcard de IP”*** → Un access-list para todas las direcciones IP que se quieran traducir
* **ip nat pool *“NombrePool” “IP de inicio” “IP de final”* netmask *“Máscara de red”*** → Para definir el pool de IP públicas a las que se traducirá.
* **ip nat inside source list 1 interface s0/0/0 overload** → Este comando define el PAT, todas las direcciones IP de la lista 1 se traducirán cuando a la IP del puerto s0/0/0 o de la interfaz que se especifique.
* **ip nat *inside/outside*** → Comando utilizado al configurar interfaces, sirve para determinar si son las direcciones IP a traducir (inside) o a las que se traducen (outside).

## OSPF MULTIAREA

Para comprobar si OSPF multitarea está bien configurado se pueden seguir varios pasos distintos.

1. Hacer ping de todos los enrutadores a todos los enrutadores a los que no estén directamente conectados, utilizando el cable serial y el ordenador.
2. Si los pings no funcionan se usaría **show ip route** en los dos enrutadores para ver si falta integrar alguna red → detectar las letras **O IA** (O: red encontrada por OSPF, IA: red encontrada a partir de otras áreas)
   1. Si ese es el caso se ha de escribir
3. **router ospf 56**
4. **network *“IP red” “Wildcard de IP”* area *“id del area”***

Una vez se haya metido las redes vecinas restantes se vuelve a hacer ping a un enrutador de una red no vecina, para comprobar que el OSPF esté bien implementado. Hay que tener en cuenta que todas las áreas tienen que estar conectadas con el área 0.

## OSPF con autenticación

Al comprobar que OSPF con autenticación funcione se debe tener en cuenta que ambos extremos de la conexión deben tener el mismo tipo de autenticación. Para comprobar que tipo de autenticación tiene configurada una interfaz habrá que introducir el siguiente comando **show ip ospf interface “interfaz”**. Este comando mostrará bastante información, pero la necesaria para este caso será las siguientes líneas:

* + **Simple password authentication enabled**
  + **Message digest authentication enabled Youngest key id is 1**

En el primer caso la interfaz está configurada con autenticación de contraseña en texto sin cifrar. En el segundo caso en cambio la interfaz está configurada con autenticación MD5 la cual cifra las contraseñas.

Otra forma de verificar la autenticación de OSPF es ingresar el comando **debug ip ospf packet**. Este comando hará que aparezcan todos los eventos de OSPF que están ocurriendo. Por lo cual, si la autenticación está activada, se verá que la comunicación está usando una “key” de algún tipo.

En caso de que la configuración no sea correcta, habrá que volver a configurarlo con los siguientes comandos (esta es la configuración para autentificación MD5, ya que es la más segura):

* **ip ospf message-digest-key 1 md5 *“contraseña”*** → Este comando se usa en cada interfaz, sirve para configurar el tipo de autenticación que se utilizará en esa interfaz.
* **area 0 authentication message-digest** → Se usa en la configuración de OSPF, sirve para que todas las interfaces de esa área (en este caso la 0) utilicen esa configuración de autenticación.

## PPP con autenticación

Para verificar la configuración del protocolo PPP se utilizará el comando **show interfaces “interfaz”**, este comando visualizará la configuración de una interfaz. En esta configuración se podrá observar si la encapsulación de PPP está activada o no. Este comando habrá que introducirlo para cada una de las interfaces que se quieran comprobar.

Otra forma de verificar la autenticación de PPP es ingresar el comando **debug ppp authentication**. Este comando hará que aparezcan todos los paquetes de PPP que se están enviando. Habrá que apagar uno de los extremos de la conexión punto a punto. Al apagar un extremo se enviará un paquete PPP, y si al observar esta comunicación aparece “key”, entonces la autenticación PPP está activa.

Para volver a configurar la interfaz con encapsulación de PPP se utilizarán los siguientes comandos:

* **encapsulation ppp** → Para establecer PPP como el método de encapsulación de la interfaz
* **ppp authentication chap** → Habilita CHAP en una interfaz serial
* **username “nombre” pass “contraseña”** → Especificar nombre y contraseña para PPP

## SNMP server

Para verificar el protocolo SNMP hará falta la aplicación “snmpwalk”. Una vez en ella, hay que hacer un request al servidor para confirmar su estado. Para ello habrá que introducir la IP del VLAN del switch al que está conectado el servidor SNMP. Introducir la comunidad (en el caso de la red de la empresa será “cisco”) en “Read Community”, y se ha de buscar el OID de lo que se pretende monitorizar. Después se empieza el monitoreo, si no aparece ningún paquete es que el protocolo SNMP no funciona.

## CDP

Para los dispositivos Cisco el protocolo CDP está activado de forma predeterminada. Por lo que se podría recopilar mucha información sobre la red con este protocolo. Para verificar la configuración de este protocolo se puede utilizar varios comandos.

El primero sería **show cdp**, este comando permitirá visualizar el estado e información sobre CDP. Por lo que con este comando podría verse si CDP está configurado. Otro comando sería **show cdp neighbors**, este permite ver la información de los dispositivos conectados directamente al él. Por si alguno de los dispositivos conectados no aparece en esta lista puede que sea que la interfaz no tiene CDP activado aun teniendo CDP activado en el dispositivo. Para esta última comprobación el comando a utilizar sería **show cdp interface**. Este comando mostrará la configuración CDP por cada interfaz del dispositivo.

Si es necesario volver a configurar el protocolo CDP serán necesarios los siguientes comandos:

* **cdp run** → Para habilitar CDP globalmente para todas las interfaces compatibles en el dispositivo
* **cdp enable** → Para habilitar CDP en una interfaz específica

## NTP

Para verificar que el protocolo NTP está funcionando correctamente y está activado podrá ingresarse el siguiente comando: **show clock detail**. El cual mostrará si la fuente de la hora es NTP o no. En caso de no estar activado se usará el comando **ntp server “IP server”**, con la IP del server que gestionará el protocolo NTP.

## TFTP

Para verificar que el protocolo TFTP está bien configurado, habrá que hacer una copia y comprobar que la copia se ha realizado correctamente. Los comandos para diferentes copias serán:

* Copiar conf al tftp: **copy running-config tftp:**
* Copiar conf del tftp: **copy tftp: running-config**
* Copiar IOS al tftp: **copy flash tftp:**
* Copiar IOS del tftp: **copy tftp: flash**

## ACL extendidas

Hay dos formas generalmente de verificar las ACL extendidas. Para la primera opción se usará el comando **show ip interface “interfaz”**. Con este comando se visualizará toda la información sobre una interfaz, y ahí se puede comprobar si alguna ACL está configurada para esa interfaz.

Otra forma de comprobarlo más claramente es utilizando el comando **show access-list**. Pero para comprobar las ACL con este comando habrá que generar tráfico por la red. Cuando las ACL bloquean o permiten los paquetes quedará registrado en este comando.

En caso de que las ACL no funcionen correctamente puede deberse al orden de configuración de las ACL o una mala implementación de ellas. Para la configuración se utilizarán los siguientes comandos:

* **access-list “ACL-id” permit/deny TCP/UDP/IP “IP denegada” “IP destino” eq “protocolo”** → Para configurar un ACL (hay que tener en cuenta que primero se tiene que ingresar el ACL más general)
* **ip access-group “ACL-id” in/out** → Se utiliza en la configuración de una interfaz, para configurar si los paquetes a filtrar saldrán o entrarán por la interfaz

# ANEXO 5 - MICROPROCESADOR

En el siguiente anexo se explicará de la forma más breve y concisa posible las funcionalidades que implementa la placa de desarrollo Olimex STM32-P107 junto al módulo lector de códigos Barcode Click y el servidor maestro. De esta forma se pretende dar a entender cuál es el funcionamiento conjunto del sistema desde la perspectiva de la placa de desarrollo.

## Función

La principal función de la placa de desarrollo es la lectura y visualización de información. En cuanto a la información leída, esta se extrae de códigos QR personalizados con la información requerida. La extracción o lectura se lleva a cabo mediante el mencionado módulo de lectura de códigos, el cual tras realizar una lectura correcta envía la información a la placa mediante comunicación serial.

Tras este paso la placa actúa como intermediario y transfiere la información leída por el escáner o lector al servidor maestro donde se encuentra la base de datos. Es entonces cuando, en base a la información transferida se realizan consultas, con el propósito de obtener información relevante para el usuario.

En caso de que el código sea correcto el resultado de la consulta realizada por el cliente de TidyFood a la base de datos se transferirá de nuevo a la placa de desarrollo, donde ahora si se utilizara dicha información para realizar la función de visualización. Dicha función se basa en visualizar en la pantalla de la placa el contenido de la caja escaneada tanto en tipo como en cantidad y contenido.

## Tecnologías empleadas

A continuación, se justificará la razón de uso de cada una de las tecnologías empleadas en el desarrollo de las mencionadas funcionalidades haciendo balance de sus ventajas y desventajas.

### KeilUvision5

En el caso del entorno de desarrollo se ha optado por emplear KeilUvision5 sin consulta o comparación previa. Esta decisión está fundada en dos motivos principales el primero de ellos es la experiencia, teniendo experiencia previa acerca del funcionamiento de este entorno y conocimiento de sus características y configuraciones, el tiempo invertido en la búsqueda de un software superior en prestaciones o facilidad de uso junto con el tiempo requerido para comprenderlo, hacen que en el caso de que dicho software exista no sea rentable su utilización.

El siguiente motivo está estrechamente relacionado con el anterior, puesto que, bajo la premisa de reciclar funciones y fragmentos de código previamente programados en este entorno, se concluyó que con el fin de evitar posibles problemas de implementación que tendrían que corregirse posteriormente era mejor continuar en el mismo entorno con la certeza de que las funciones y librerías funcionarían correctamente.

### Lenguaje C

En el caso del lenguaje, si bien es cierto que el entorno de programación condiciona al usuario a emplearlo, se determino era una ventaja, dado que todos los miembros del equipo tenían experiencia previa con el lenguaje lo que facilitaba la labor de generar nuevo código y la recepción de feedback de parte de otros miembros del equipo. Lo cual no habría sido posible de haber utilizado un lenguaje que, aunque posiblemente mejor de cara a la velocidad de programación, no fuera comprendido por todo el equipo. Además, se considera que es un lenguaje intrínsecamente sencillo lo que ahorra quebraderos de cabeza al realizar acciones que en otro caso podrían ser más complejas.

### Protocolo UART

UART (universal asynchronous receiver / transmitter) o USART cuando es síncrono es el protocolo de comunicación que se ha empleado para realizar las comunicaciones entre los diferentes dispositivos que componen el sistema a excepción del servidor de respaldo.

El motivo por el que se selecciono este protocolo o conjunto de normas, para comunicar los dispositivos radica en su simpleza. Al fin y al cabo, UART es sumamente simple y solo utiliza dos hilos entre el transmisor (pin de transmisión de datos) y el receptor (pin de recepción de datos) para transmitir y recibir en ambas direcciones. Estas líneas de comunicación pueden estar configuradas modo simplex (una sola dirección de envió de datos), semidúplex (ambos envían, pero uno cada vez) o dúplex completo (ambos envían al mismo tiempo).

Además, el que sea asíncrono también es una ventaja, ya que de lo contrario seria necesario configurar un reloj en cada extremo de la comunicación y ambos deberían de estar sincronizados para que funcionaran correctamente. En cambio, siendo asíncronos, basta con configurar el mismo baudrate o tasa de baudios (unidades de señal por segundo) para que puedan comunicarse correctamente. Un baudio puede contener varios bits por lo que no hay que confundir el baudrate con el bitrate ya que, aunque pueden coincidir no siempre lo harán.

Ilustración 51 RB

Ilustración 52 RS

Diagrama

Descripción generada automáticamente

Ilustración 53 Protocolo UART

Otra de las ventajas es que los datos transmitidos con el protocolo UART se envían en forma de trama lo que hace fácil entender la cantidad de datos y paquetes que se envían dependiendo del mensaje y de la configuración que se haga de la trama ya que ciertos parámetros se pueden configurar dependiendo de los requerimientos. En la imagen inferior se puede observar cual es la composición de una trama enviada mediante el protocolo UART la cual se describe a continuación.

Gráfico en cascada

Descripción generada automáticamente con confianza baja

Ilustración 54 Protocolo UART 2

La trama de datos comienza con un start bit ya que al ser asíncrono hay que notificar al extremo contrario de que comienza una nueva trama con el primer bit de esta, tras el start bit vienen los data bits que conforman el mensaje, en total hay 8 data bits, pero se puede configurar para que sean 9. Esto quiere decir que una sola trama no puede transferir datos individuales que requieran más de 8 bits para ser interpretados, a menos que se active el bit adicional. Antes de finalizar esta el parity bit que se emplea en la detección de errores y puede ser configurado como bit de datos adicional, y finalmente están los stop bits los cuales son 2 y se pueden configurar para usar uno o ambos, estos bits son los que indican el final de la trama.

## Sistema de archivos

El mostrado arriba es el sistema de archivos, del proyecto TidyFood correspondiente a la parte de microcontroladores. Esta organizado de manera que se pueda tener una idea general del funcionamiento simplemente observando las diferentes librerías empleadas.

La organización está dividida en ficheros de encabezado (.h) ficheros de funciones (.c) programa principal (main.c) y librería para el uso de la pantalla (libMU) perteneciente a Eñaut Muxika.

## Funcionalidades

A continuación, se describen en mayor profundidad las funcionalidades implementadas en la placa de desarrollo utilizando las tecnologías anteriormente mencionadas y que en conjunto conforman el lector de contenido de cajas del sistema TidyFood. Estas funcionalidades se centran en la forma en que se realizan la lectura de datos y la visualización de estos mediante la comunicación serial.

### Inicialización de la comunicación

Aunque propiamente no es una funcionalidad, ya que el usuario no es capaz de actuar sobre dicha función, es importante conocer la forma en que se configura la comunicación para comprender la forma en que posteriormente actuara cada uno de los dos puertos configurados.

Como se puede observar en la imagen de la parte superior, lo que se hace en primer lugar es activar los que serán los pines de transmisión y recepción de datos en la placa, configurándolos para ello en el modo de función alternativa, lo que permite más tarde seleccionar una de entre las varias funciones que tiene ese pin, en nuestro caso la función USART1 es la que nos interesa.

Forma

Descripción generada automáticamenteA continuación, se define el baudrate de forma que coincida con el definido por defecto en el lector de códigos. Este se calcula con la fórmula de la parte inferior teniendo en cuenta la frecuencia del bus en el que se encuentra, el oversampling (nulo en este caso) y el baudrate requerido. El resultado obtenido se divide en la parte entera y la parte decimal y ambas se introducen en sus respectivas posiciones configurando el baudrate para este USART.

Ilustración 55 TX/RX baud

Para ir finalizando se activan ambos pines de recepción y transmisión en el correspondiente USART de manera que sea capaz de realizar ambas acciones a través de cada uno de los pines. También se configura la cantidad de stop bits para que sea uno solo y finalmente activamos el propio USART1. Adicionalmente se configura también la correspondiente interrupción de USART1 de forma que al escribir datos en el registro de datos se detecte automáticamente, aunque dicha función se encuentra en otro bloque.

La configuración de USART3 que es el empleado en la comunicación placa-servidor se configura de igual manera variando el baudrate y los pines configurados.

### Función de lectura de datos

Imagen que contiene Diagrama

Descripción generada automáticamenteLa función de lectura de datos es la responsable de enviar al servidor los datos leídos por el lector de códigos, primero a la placa mediante USART1 y posteriormente de la placa al servidor mediante USART3. Esta acción es necesaria para poder hacer la consulta de información necesaria en el servidor.

Ilustración 56 Función de lectura de datos 1

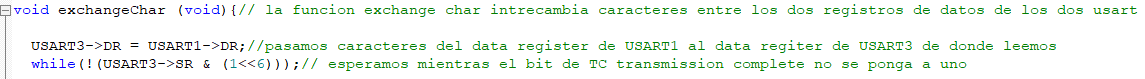
Como se observa en la imagen superior, el desencadenante de la función de lectura es la interrupción de USART1 que se activa al recibir datos desde el lector de códigos. Esta función se activará tantas veces como caracteres contenga el código leído o escaneado y cada vez que eso pase llamará a la función exchangeChar la cual intercambia información entre ambos USART 1 y 3, además de cambiar de estado un led para proporcionar confirmación visual de que se esta realizando la transferencia de datos.

Ilustración 57 Función de lectura de datos 2

En la función exchangeChar se intercambia la información tal y como indica su nombre y se espera a que la confirmación de transferencia completa se active para poder leer un nuevo carácter si es que lo hay. Tras intercambiar la información, al igual que las interrupciones, el cliente es capaz de detectar que hay información por leer en USART3 y recibirá la información leída por el escáner de códigos.

### Función de visualización

La función de visualización es la encargada de mostrar en pantalla los datos obtenidos a partir del código escaneado y posteriormente enviado al servidor para hacer la consulta de información correspondiente. Tal y como se ve en imagen inferior y al igual que en el caso anterior el desencadenante es la interrupción de USART3 en este caso, la cual se activa al recibir información desde el servidor. En este caso cada vez que se reciba un carácter que no sea el carácter de finalización, este se guardará en una cadena de caracteres reconstruyendo el mensaje poco a poco, esto se logra igual que en la función de exchangeChar solo que la información en este caso se Texto

Descripción generada automáticamentetransfiere a data en vez de a otro USART.

Ilustración 58 Función visualización 1

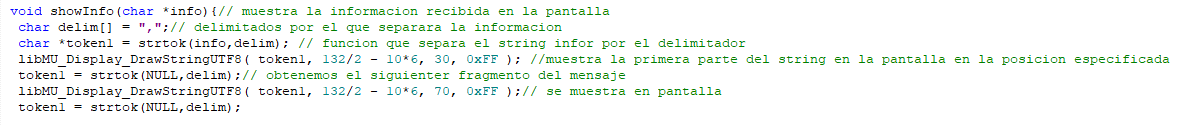
Una vez recibido el carácter de finalización y estando seguros de que no va a haber mas interrupciones, se llama a 2 funciones la primera muestra en el servidor el mensaje recibido por la placa para confirmar que es correcto. El segundo en cambio llama a la función de la parte inferior llamada showInfo, esta función es la encargada de mostrar en la pantalla de la placa el mensaje recibido desde el servidor, habiéndolo fraccionado previamente para mostrar cada parámetro en su correspondiente campo.

Ilustración 59 Función visualización 2

## Conclusión

Para concluir se podría decir que este apartado se ha desarrollado de forma que sea lo más simple y visual posible con el código mínimo imprescindible y buena organización, todo ello con la finalidad de desarrollar las 2 funciones principales de las que se compone de la mejor manera posible. Existe una 3 funcionalidad que actualmente no se ha implementado por completo pero que se espera tener terminada próximamente y que consiste en la modificación y sobreescritura de la información recibida. Sin embargo, esta es la última versión plenamente funcional y la que actualmente conforma el sistema de gestión de alimentos TidyFood.

# ANEXO 6 – APLICACIÓN JAVA

En el siguiente se explicará de manera técnica la aplicación desarrollada con el lenguaje de programación Java junto con los frameworks que se explicaran a continuación.

## FRAMEWORKS UTILIZADOS

En este subpunto se explicarán las librerías involucradas en el desarrollo del software de TidyFood.

### SWING

Swing es la librería utilizada para la creación de las interfaces gráficas y todo tipo de componentes para interacción del usuario con estas. La librería mencionada, es la base sobre la que se ha desarrollado la aplicación.

### JSSC

Esta librería se utiliza para la comunicación por el puerto serial. La librería facilita al usuario la conexión por línea serial utilizando la clase que ofrece SerialPort. Esta clase ofrece una conexión simple con parámetros estáticos para la paridad, el baudrate o los bits de información, por ejemplo, que hacen más legible la configuración del puerto. Facilita mucho mandar y recibir bytes e incluso, como en esta aplicación, permite leer una serie de 6 bytes sin ninguna dificultad.

### JAVA CONNECTOR MYSQL

La librería JAVA Connector que MYSQL proporciona a los desarrolladores, es una librería que permite al desarrollador conectarse a la base de datos y ejecutar cualquier tipo de Query en él.

### JFREECHART

JFreeChart es una librería de Java que nos permite hacer gráficos complejos de manera sencilla. Se ha usado esta librería para la realización de gráficos que muestran la cantidad de productos recibidos y entregados.

## DISTRIBUCION DE FICHEROS

Imagen que contiene Tabla

Descripción generada automáticamenteEn este apartado se muestra como está distribuido el diagrama de paquetes en este desarrollo.

Ilustración 60 Distribución de ficheros

Se ha creado paquete para las clases generales como pueden ser ‘Calendario’, ‘MiAccion’ o ‘Escala’, es decir, clases que en general se quieren usar en muchos sitios y que en su mayoría tienen funciones estáticas.

Por otro lado, se tiene el paquete de ‘BaseDeDatos’, que posee toda la infraestructura para guardar, eliminar y coger la información de la base de datos.

El paquete de ‘lineaSerial’ se utiliza para tanto enviar como recibir mensajes de la basys.

Los paquetes de modelo contienen las clases que como su nombre indica, actúan de modelo, es decir, en la que se guarda toda la información, dentro de estos paquetes existen tanto los ‘ListModel’ que se usan en las vistas como el modelo normal (que no extiende ningún ListModel). Se han creado clases para guardar la información, aunque sean muy parecidas a los modelos de ListModel, ya que se ha considerado que, aunque no siempre suceda, los modelos que extienden de ListModel tienen un funcionamiento muy específico, y dependiendo del uso que le quieras dar en la vista distan entre ellos (como ejemplo, hemos creado una clase ‘Items’ que posee toda la información de los ítems, y por otro lado se han creado los ListModel ‘ListaItem’ e ‘ItemsSimple’.) Esto sucede porque en una vista mostramos los ítems y se puede buscar por nombre además de aplicar filtros, y en otra enseñamos toda la lista de Items al completo. Por tanto, creamos un modelo genérico ‘Items’ que principalmente sirve para guardar y escribir correctamente en ficheros, así como para ordenar la lista. Esto se repite en todas las clases que poseen listas, como pueden ser Personas, Proveedores, Productos o Lotes.

Las clases que no extienden de ListModel y que por tanto se usan para recibir y guardar datos en ficheros, pertenecen todas a un único modelo denominado ‘ModeloGeneral’. Mas adelante se explicará el porqué de esta decisión.

Los paquetes de ventanas poseen la vista de cada ventana, así como su controlador y los renderers que usan.

Finalmente, el paquete de ‘vistaPrincipal’ contiene el JFrame general, así como el controlador principal. Mas adelante incidiremos en el funcionamiento de este.

## FUNCIONAMIENTO DEL PRODUCTO

Debido a que el programa iba a tener más de una ventana, se pensó en plantear la aplicación de modo en el que no se crearan más JFrame, sino que se actualizase el panel del JFrame principal. Estos paneles tendrían su propio controlador, sin embargo, era necesario saber cuál era la próxima ventana que se iba a ejecutar, por tanto, se decidió que todos los paneles implementasen un ‘PropertyChangeSupport’ en el controlador y que el listener fuera el controlador principal, de manera en la que cuando se clicase un botón que cambiara a otra ventana, este mandase un ‘contentsChanged’ a el controlador principal con un mensaje sobre que ventana abrir a continuación.

En el JFrame principal se crea el ‘ModeloGeneral’, que posee todos los modelos usados en las ventanas. Este modelo es el que se pasa a los paneles creados, y dentro de estos se coge los modelos que se usan mediante getters.

Se decidió implementar de esta manera ya que gráficamente es mucho más agradable, es un sistema organizado en el que la escalabilidad es muy alta y nos permite transitar entre ventanas en paralelo y no solo linealmente.

## FUNCIONALIDADES QUE DESTACAR

En este apartado se van a destacar las principales funcionalidades que le aportan un valor a la aplicación.

### FUNCIÓN MEDIANTE ESCALA

Para ser más exactos, la aplicación está diseñada para verse en pantalla completa, y da igual en que tamaño de pantalla se vea, la aplicación se verá igual. Esto funciona así debido a que las posiciones de los separadores, los tamaños de los botones, las imágenes… todas tienen un tamaño en función del tamaño de pantalla, ya que no poseen un valor exacto para el tamaño, sino que dividen el tamaño de la pantalla (por ejemplo, la imagen puede tener un tamaño en x que sea el tamaño total de la pantalla en horizontal entre 20, siendo igual en cualquier pantalla).

Esta escalabilidad se realiza mediante la clase ‘Escala’, que proporciona funciones estáticas para escalar correctamente imágenes o botones, así como el tamaño de la pantalla.

### GESTIÓN DE DATOS MEDIANTE LA BASE DE DATOS

Para la gestión de datos eficiente, se han desarrollado interfaces para las 4 funciones básicas utilizadas en MySQL, para actualizar los datos, para seleccionarlos, para eliminarlo y para almacenarlos. Para hacer esto posible se ha utilizado la librería de MySQL creada para la comunidad de desarrolladores.

Para la ejecución de Querys lo primero que se debe hacer es establecer conexión con la base de datos, es decir, en este caso con jdbc:mysql://localhost:3306/Tidyfood. Una vez conseguida la conexión con la BD, y configurado el Statement que se quiere ejecutar, se podrá modificar la base de datos y obtener información de manera sencilla.

En el caso de las modificaciones, es decir en los casos de actualizar, eliminar o insertar datos, con preparar el statement será suficiente. Pero en los casos de obtener datos, se debe guardar los datos en una variable de tipo ResultSet e ir leyendo estos de uno en uno.

Texto

Descripción generada automáticamenteEn esta imagen se puede ver la preparación de un statement de modificación, donde se debe configurar este, sustituyendo los valores obtenidos en los parámetros por los signos de interrogación de la query.

Ilustración 61 Sustitución de valores query

Imagen que contiene Escala de tiempo

Descripción generada automáticamenteEn la siguiente imagen, se puede visualizar la estructura de una función de recopilación de información de una base de datos.

Ilustración 62 Conexión base de datos

### REGISTRO DE PRODUCTOS, PROVEEDORES Y USUARIOS

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamenteComo se ha comentado en el punto 7 del documento, la aplicación te da la opción de registrar diferentes entidades del programa como pueden ser productos, proveedores y usuarios. Para llevar esto acabo, la obtención de datos se realiza mediante formularios. Una vez recogidos los datos de los JTextBoxes, estos se envían a una función que verifica que no existen en la base de datos. Finalmente, con lo explicado anteriormente sobre la conexión con la base de datos, se hace la modificación que haga falta y se almacenan los datos en la base de datos.

Ilustración 63 Formulario Registro

### INICIO DE SESION

Para garantizar la seguridad de la comida y del software, se ha implementado un sistema de seguridad, un sistema de inicio de sesión. El funcionamiento de este es muy simple. En la pantalla de inicio de sesión mencionada anteriormente en el punto 7, el **usuario introduce sus credenciales. En ese instante, se envía una petición a la base de datos y se verifica el resultado de la llamada. Si la llamada viene vacía, significara que no hay ningún usuario con ese nombre y esa contraseña en la base de datos. Si los datos no coinciden, el sistema lanzara un mensaje de error, de la otra manera, si los datos coinciden, el usuario podrá acceder al sistema. Además de eso, el usuario administrador tendrá permisos únicos, entre ellos la creación de un usuario nuevo y la modificación de estos. Para el registro de proveedores se necesitará permisos de administrador. La sesión permanecerá iniciada gracias a la clase “UsuarioActual”, el cual es una clase estática que guarda los datos de inicio de sesión y lo mantiene mientras que la aplicación siga activa.**

Texto

Descripción generada automáticamente

Ilustración 64 Verificacion Inicio Sesion