UNIVERSIDAD DE BURGOS ESCUELA POLITÉCNICA SUPERIOR



Grado en Ingeniería en Informática

TFG del Grado en Ingenier´ıa Inform´atica

**LucErik**

**Documentaci´on T´ecnica**



Presentado por Carolina Colina Zamorano

en Universidad de Burgos — 20 de enero de 2021 Tutor: D. José Manuel Galán Ordax y D. José Ignacio Santos Martín

**Índice General**

**Índice de Figuras**

**Índice de Tablas**

**Plan de Proyecto Software**

1. Introducción

En este primer apartado se procede a explicar cómo se ha planificado el desarrollo del proyecto explicado más adelante, en la planificación temporal, así como el estudio de viabilidad donde se calcularán los costes y beneficios que se podrán obtener con la aplicación.

1. Planificación temporal

En la primera reunión se hizo un repaso de los primeros pasos a seguir para es desarrollo del proyecto.

Antes de comenzar el proyecto, se decidió utilizar una metodología SCRUM para su gestión de tal manera que en cada sprint se establecen los prototipos funcionales que se van a implementar.

Se crea un repositorio del proyecto en GitHub y se instala la extensión ZANU que nos permite crear las Issues y ubicarlas en el panel o Board. De esta forma podemos gestionar cada tarea de manera muy visual y cómoda.

Al comienzo de cada Sprint se hace una reunión para establecer cuáles serán las funcionalidades a implementar de tal manera que siempre sea un prototipo funcional, aunque no cumpla con todos los requisitos.

**SPRINT 1 – 16/11/2020 – 29/11/2020**

En esta primera semana se ha creado y configurado el repositorio del proyecto en GitHub, se ha elegido el editor de texto para la memoria, el lenguaje de programación del proyecto, búsqueda de trabajos relacionados y la instalación de la extensión de Zotero así como el aprendizaje del uso de la herramienta.

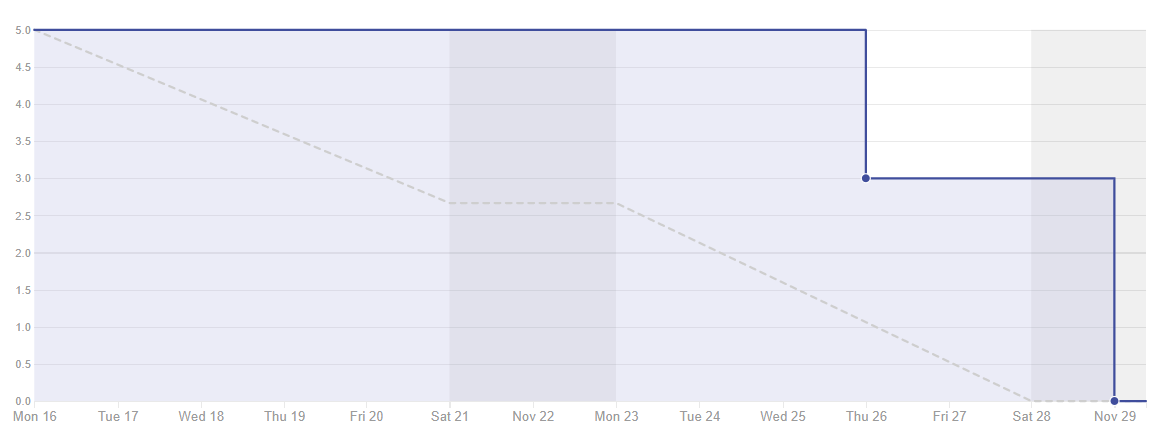


Figura A1: Burndown Report - Sprint 1



Figura A2: Issues - Sprint 1

|  |  |  |
| --- | --- | --- |
| **TAREA** | **NOMBRE DE LA TAREA** | **TAG** |
| #1 | Elegir editor de texto para la memoria | Decisión |
| Investigación |
| #2 | Buscar trabajos relacionados | Investigación |
| #3 | Crear y configurar repositorio | Configuración |
| #4 | Instalación de Zotero | Configuración |
| Instalación |
| #5 | Uso de herramientas | Investigación |

Tabla A1: Tareas - Sprint 1

**SPRINT 2 – 30/11/2020 – 13/12/2020**

En estas dos semanas el trabajo se ha centrado en elegir el IDE y en el diseño de la interfaz y el algoritmo. Es importante decidir cuál va a ser el comportamiento de la aplicación y qué debe mostrar al usuario para saber cómo diseñar la base de datos.

Una vez decidido, se ha creado la base de datos en el servidor, se generan las tablas y se añaden algunos datos provisionales para empezar a programar.

El código implementado en este sprint desarrolla la conexión con la base de datos y en mostrar las órdenes en pantalla. En este caso, las órdenes que muestra no son las reales, sólo es un comienzo de cómo mostrar los datos en una tabla en pantalla.

Se ha comenzado a documentar la memoria del título empezando por la introducción y los objetivos del proyecto.

También se ha continuado con el Anexo documentando el Sprint 1.

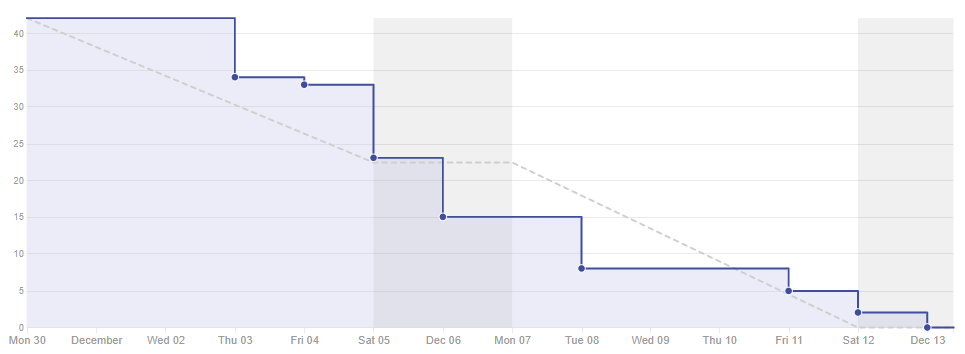


Figura A2: Burndown Report - Sprint 2

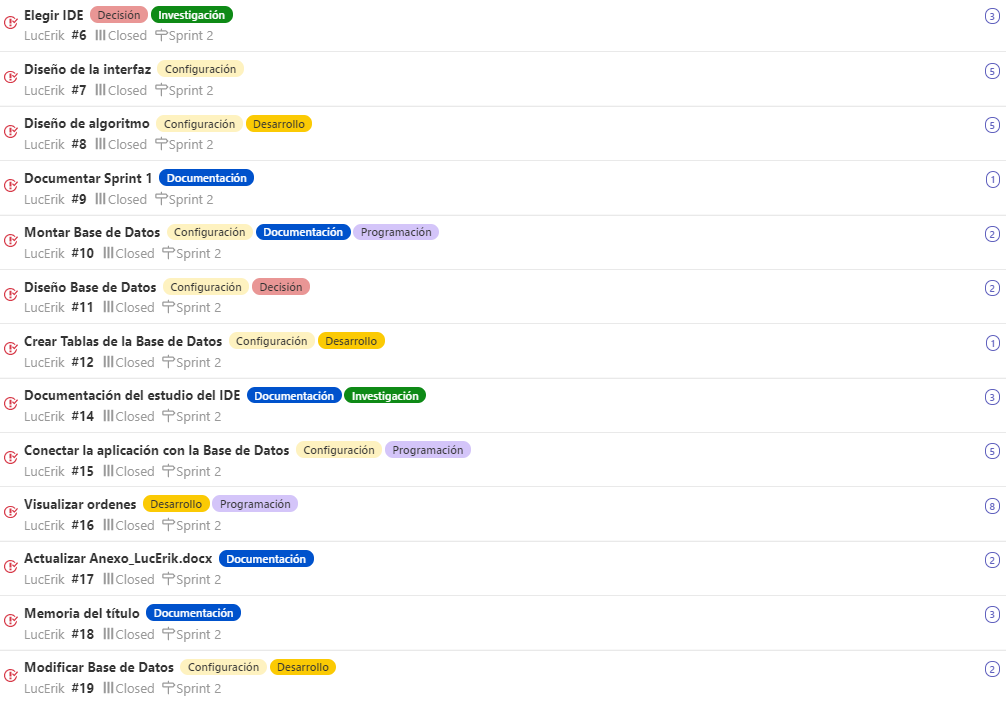


Figura A3: Issues - Sprint 2

|  |  |  |
| --- | --- | --- |
| **TAREA** | **NOMBRE DE LA TAREA** | **TAG** |
| #6 | Elegir IDE | Decisión |
| Investigación |
| #7 | Diseño de la interfaz | Configuración |
| #8 | Diseño del algoritmo | Configuración |
| Desarrollo |
| #9 | Documentar Sprint 1 | Documentación |
| #10 | Montar Base de Datos | Configuración |
| Documentación |
| Programación |
| #11 | Diseño de Base de Datos | Configuración |
| Decisión |
| #12 | Crear tablas de la Base de Datos | Configuración |
| Desarrollo |
| #14 | Documentación del estudio del IDE | Documentación |
| Investigación |
| #15 | Conectar la aplicación a la Base de Datos | Configuración |
| Programación |
| #16 | Visualizar órdenes | Desarrollo |
| Programación |
| #17 | Actualizar Anexo\_LucErik.docx | Documentación |
| #18 | Memoria del título | Documentación |
| #19 | Modificar Base de Datos | Configuración |
| Desarrollo |

Tabla A2: Tareas – Sprint 2

**SPRINT 3 – 14/12/2020 – 23/12/2020**

En estos 10 días, el trabajo se ha centrado principalmente en la parte de programación.

Se han creado 2 scripts que se ejecutan en segundo plano:

* El primero de ellos ha sido para actualizar los materiales a medida que se van produciendo talones.
* El segundo, actualiza el número de cubiertas fabricadas y se decrementa el número de talones disponibles en la máquina.

Se ha elaborado de un documento donde se explica de manera detallada y gráfica como es el proceso productivo de una cubierta para una mejor compresión del proyecto.

Se han hecho modificaciones en la base de datos y se ha creado un fichero en Excel con todos los datos de la tabla HUECOS y MAQUINAS para la base de datos inicial definitiva.

Se diseña el algoritmo para generar las órdenes en función de las necesidades de cada máquina y se crea el Script que obtiene los datos de las diferentes tablas y genera las órdenes.

Se crea la ventana Buscar Material. Cuando no existe material suficiente en las RTBS, existe la posibilidad de que haya en otras máquinas.  
El programa permite buscar ese material en otras ubicaciones, para ello abre una ventana nueva en la que se muestra dónde puede haber stock de ese material.

Se modifica la ventana órdenes. Muestra las órdenes en pantalla.  
Recoge los datos necesarios para saber qué material hay que llevar, a qué máquina y qué cantidad.  
Además muestra la cantidad de cubiertas que todavía puede hacer con el material que tiene y las cubiertas que le faltan por hacer de ese material.

Se implementa la función del botón Recoger Material. Al pulsar en el botón "Buscar Material", se abre una nueva ventana (Buscar Material) que nos muestra los lugares donde puede haber más material de ese tipo. Al pulsar el botón "Recoger Material", recoge el número de carros que se quitan de un lugar para actualizar la base de datos.

También se ha continuado con el Anexo documentando el Sprint 2.

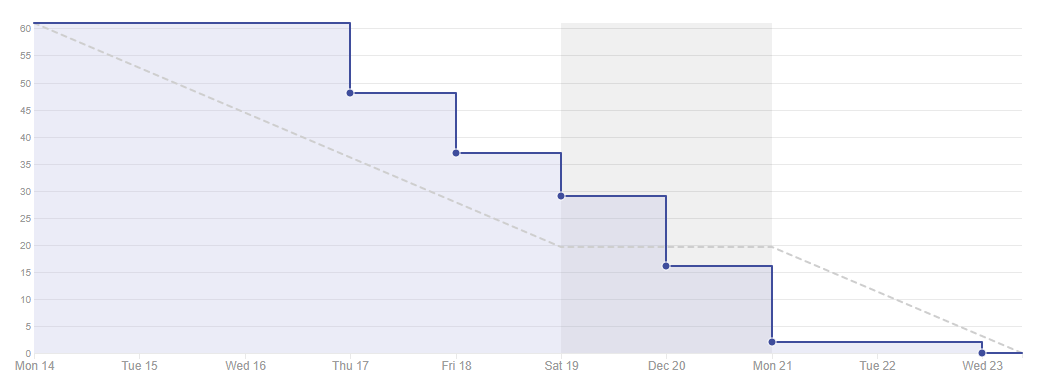


Figura A3: Burndown Report - Sprint 3

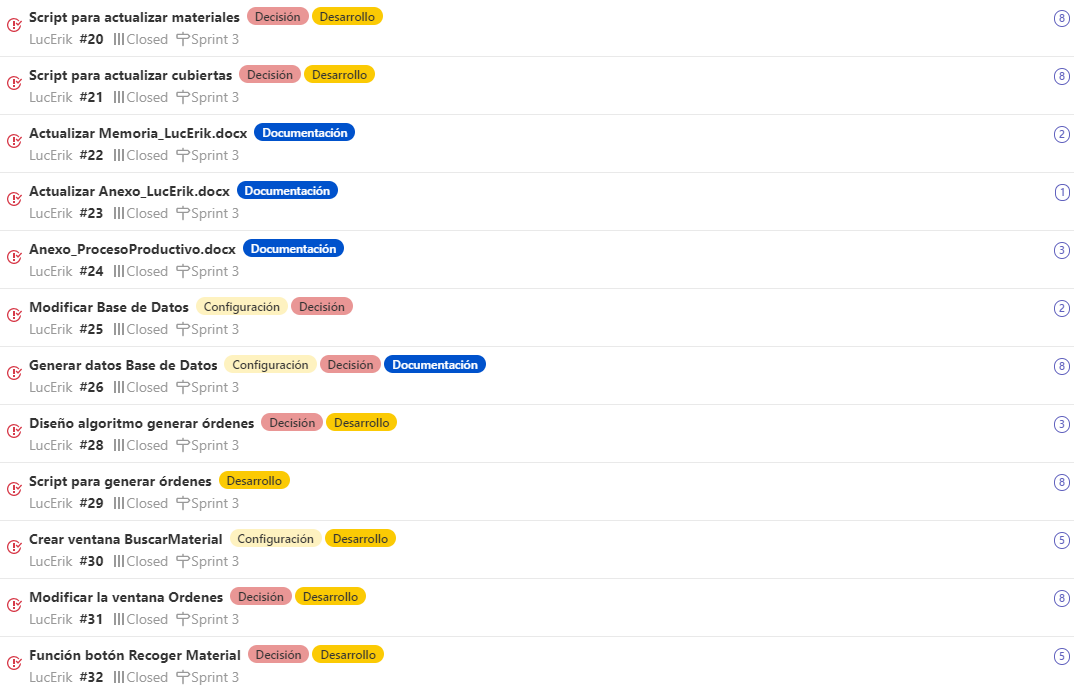


Figura A4: Issues - Sprint 3

|  |  |  |
| --- | --- | --- |
| **TAREA** | **NOMBRE DE LA TAREA** | **TAG** |
| #20 | Script para actualizar materiales | Decisión |
| Desarrollo |
| #21 | Script para actualizar cubiertas | Decisión |
| #22 | Actualizar memoria LucErik | Documentación |
| #23 | Actualizar anexo LucErik | Documentación |
| #24 | Crear anexo Proceso Productivo | Documentación |
| #25 | Modificar base de datos | Configuración |
| Decisión |
| #26 | Generar datos de la Base de Datos | Configuración |
| Decisión |
| Documentación |
| #28 | Diseñar algoritmo generar órdenes | Decisión |
| Desarrollo |
| #29 | Crear script para generar órdenes | Desarrollo |
| #30 | Crear ventana buscar material | Configuración |
| Desarrollo |
| #31 | Modificar la ventana órdenes | Decisión |
| Desarrollo |
| #32 | Función botón recoger material | Decisión |
| Desarrollo |

Tabla A3: Tareas – Sprint 3

1. Estudio de viabilidad

En este apartado se van a calcular los costes y beneficios del proyecto el un supuesto caso que se desarrollara para una empresa.

**VIABILIDAD ECONÓMICA**

En primer lugar se detallan los costes ya continuación los beneficios

**EMPLEADOS**

En este apartado se detallan los costes que supone tener empleados para el desarrollo de la aplicación.

**Especificación de Requisitos**

* 1. Introducción

En este apartado se van a definir los requisitos funcionales y no funcionales que se establecieron al comienzo y que debe cumplir el proyecto.

* 1. Objetivos generales

El proyecto tiene como objetivo crear una aplicación que muestre un listado de órdenes a un operario.

Dichas órdenes deben proporcionar al operario la mayor cantidad de información posible para facilitarle el trabajo.

Además, le permite la opción de buscar el material que se necesita en ubicaciones alternativas.

* 1. Catálogo de requisitos

**REQUISITOS FUNCIONALES**

* **R.F-1: Logan de Usuarios:** El programa permite a los usuarios introduciendo sus credenciales y acceder a sus funcionalidades.
* **R.F-2: Visualizar las órdenes:** La aplicación muestra las órdenes que el operario debe cumplir.
* **R.F-3: Buscar material:** Muestra otras ubicaciones alternativas a la que muestra por defecto o en el caso de que no muestre ninguna.
  + **R.F-3.1: Recoger material:** Cuando encuentra otras ubicaciones, permite recoger el material de dichas ubicaciones. Actualizando las ubicaciones de donde recojo el material y a donde lo llevo.
* **R.F-4: Llevar material:** Al seleccionar una orden, y pulsar el botón llevar material, actualiza el stock de la máquina a la que lo lleva y de la ubicación de donde lo recoge.
* **R.F-5: Actualizar:** Posee un botón que actualiza el listado de órdenes que se muestran en pantalla.

**REQUISITOS NO FUNCIONALES**

* **R.N.F-1: Usabilidad:** La aplicación debe ser sencilla e intuitiva y permitir al usuario su aprendizaje de manera rápida.
* **R.N.F-2: Eficiencia:** La actualización de los datos ante las acciones del usuario debe ser lo más rápido posible, así como el almacenamiento de los datos en la base de datos.
* **R.N.F-3: Disponibilidad: La aplicación debe estar disponible siempre que el usuario lo necesite.**
* **R.N.F-4: Escalabilidad:** La aplicación debe estar preparada para cambios que puedan surgir
* **R.N.F-5: Confiabilidad:** Cumplirá las funciones para las que se ha creado.
* **R.N.F-6: Mantenibilidad:** Permitirá cambios de manera sencilla.
  1. Especificación de requisitos

**CASOS DE USO**

|  |  |  |
| --- | --- | --- |
| CU-01 | Login de Usuario | |
| Requisitos relacionados | RF1 | |
| Descripción | Permite al usuario acceder a la aplicación | |
| Precondiciones | El usuario debe estar en la base de datos | |
| Acciones | **Paso** | **Acción** |
| **1** | EL usuario introduce su número de operario y el número de servicio. |
| **2** | Pulsa el botón Conectar |
| **3** | Si el número de operario es correcto se muestra su nombre y el botón Acceder |
| Postcondiciones | El número de operario y el número de servicio debe estar en la base de datos | |
| Excepciones | Si el número de operario o el servicio no es correcto no permite acceder a la ventana de órdenes. | |
| Importancia | Alta | |

Tabla B1 – Caso de Uso 1

|  |  |
| --- | --- |
| CU-02 | Mostrar Órdenes |
| Requisitos relacionados | RF2 |
| Descripción | Permite al usuario visualizar las órdenes |
| Precondiciones | El usuario se debe haber logueado |
| Acciones | - |
| Postcondiciones | - |
| Excepciones | - |
| Importancia | Alta |

Tabla B2 – Caso de Uso 2

|  |  |  |
| --- | --- | --- |
| CU-03 | Buscar Material | |
| Requisitos relacionados | RF3 | |
| Descripción | Permite saber si existe un material en otra ubicación aparte de en la que se muestra | |
| Precondiciones | El usuario debe haber seleccionado una orden de la lista | |
| Acciones | **Paso** | **Acción** |
| **1** | El usuario selecciona una orden |
| **2** | Pulsa el botón Buscar Material |
| **3** | Se muestra una nueva ventana con las ubicaciones alternativas. |
| Postcondiciones | El número de operario y el número de servicio debe estar en la base de datos | |
| Excepciones | Si no hay ninguna orden seleccionada el botón no está habilitado | |
| Importancia | Alta | |

Tabla B3 – Caso de Uso 3

|  |  |  |
| --- | --- | --- |
| CU-04 | Recoger Material | |
| Requisitos relacionados | RF3.1 | |
| Descripción | Recoge el material de una ubicación alternativa a la que se muestra en la orden y lo lleva a la máquina seleccionada | |
| Precondiciones | De la ubicación elegida, el usuario selecciona el número de carros de una lista desplegable | |
| Acciones | **Paso** | **Acción** |
| **1** | El usuario elige una ubicación |
| **2** | El usuario selecciona una cantidad de esa ubicación |
| **3** | Pulsa el botón Recoger |
| Postcondiciones | Se actualiza el stock del material en el origen y en el destino | |
| Excepciones | - | |
| Importancia | Alta | |

Tabla B4 – Caso de Uso 4

|  |  |  |
| --- | --- | --- |
| CU-05 | Llevar Material | |
| Requisitos relacionados | RF4 | |
| Descripción | Recoge el material de una ubicación y lo lleva a la máquina seleccionada | |
| Precondiciones | El usuario debe haber seleccionado una orden de la tabla | |
| Acciones | **Paso** | **Acción** |
| **1** | El usuario elige una orden |
| **2** | Pulsa el botón Llevar Material |
| Postcondiciones | Se actualiza el stock del material en el origen y en el destino | |
| Excepciones | Si no se ha seleccionado ninguna orden no se puede llevar material. | |
| Importancia | Alta | |

Tabla B5 – Caso de Uso 5

|  |  |
| --- | --- |
| CU-06 | Actualizar Órdenes |
| Requisitos relacionados | RF2 |
| Descripción | Permite al usuario actualizar las órdenes |
| Precondiciones | - |
| Acciones | - |
| Postcondiciones | - |
| Excepciones | - |
| Importancia | Media |

Tabla B6 – Caso de Uso 6

**DIAGRAMA DE CASOS DE USO**

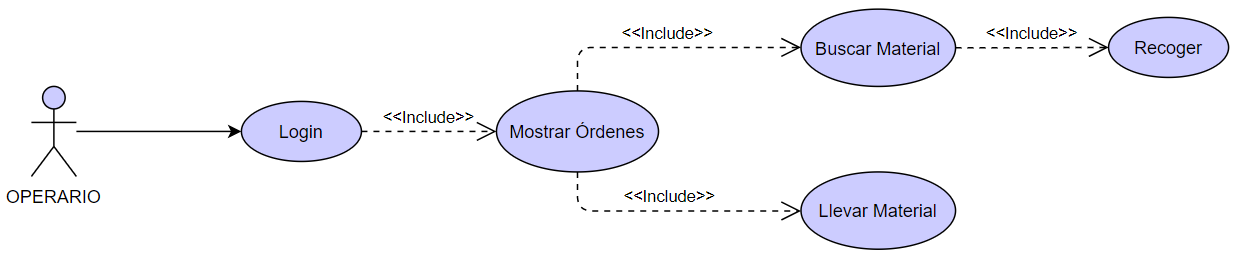


Figura B1 – Diagrama de Casos de Uso

**Especificación de Diseño**

1. Introducción

En este apartado se va a definir cómo se han implementado y diseñado las diferentes partes de la aplicación

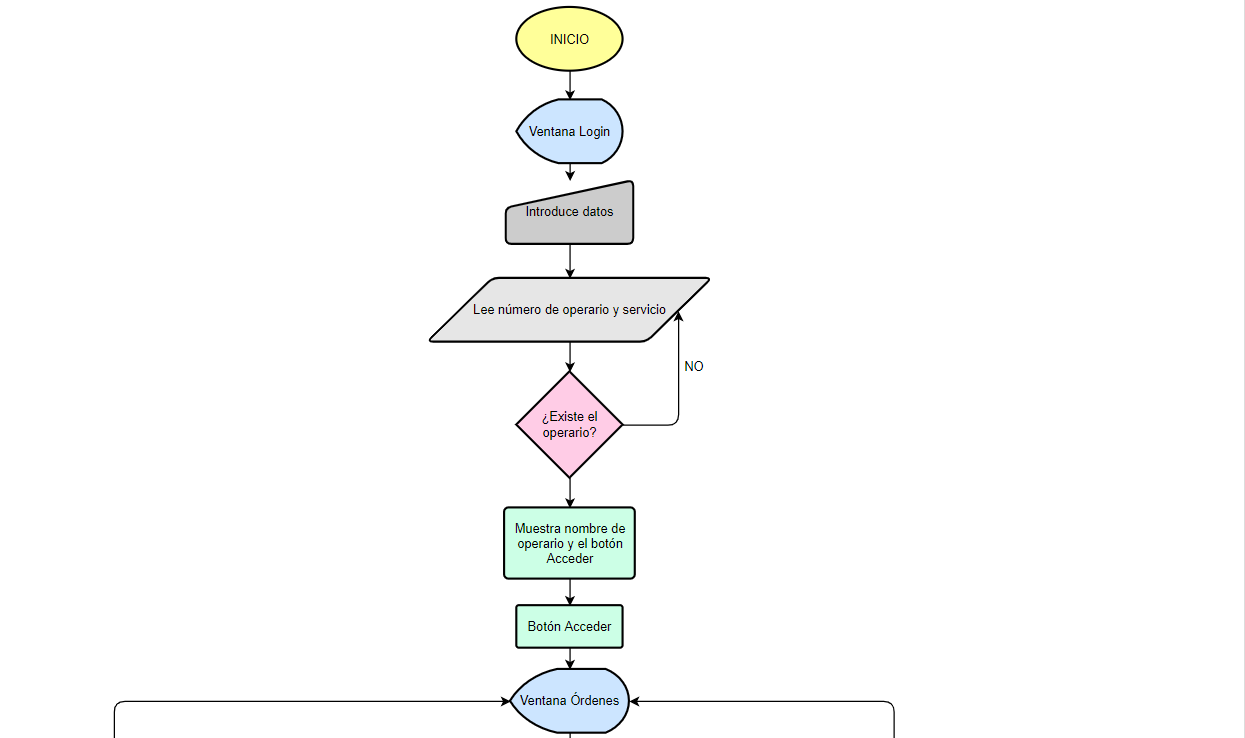
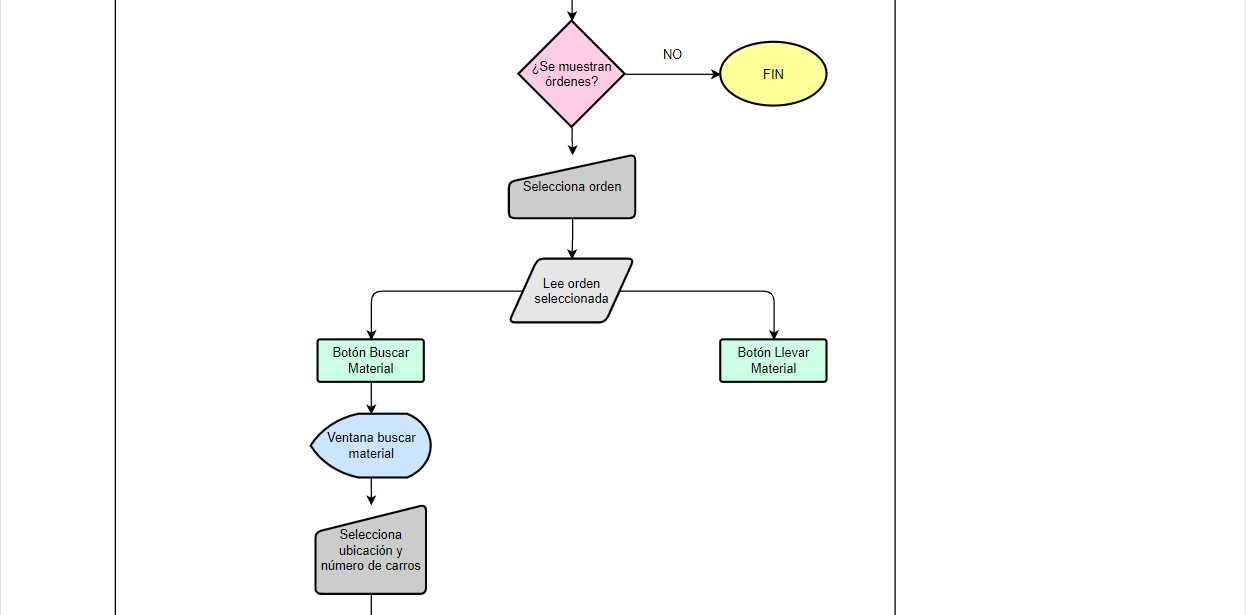
1. Diseño de datos

A continuación se muestra el listado de las tablas de la base de datos.

* **OPERARIOS:** Guarda el número del operario, su nombre y el turno que tiene asignado.
* **SERVICIOS:** Contiene el número de identificador del servicio y el nombre de cada uno de ellos.
* **LISTADO\_RTBS:** Guarda un listado con el nombre las RTBS que hay.
* **PRODUCCIÓN\_TALONES:** Almacena el pedido del turno para cada una de las RTBS. Identificador de la RTB, el material que tiene que hacer, la cantidad, los talones que lleva producidos y los totales que se tienen que hacer.
* **MATERIALES:** A medida que se van produciendo talones, en la máquina materiales, almacena o actualiza el stock de talones.
* **MAQUINAS:** Guarda por cada máquina los huecos que tiene para dejar talones, el material de talones con el que está fabricando cubiertas actualmente, el material que usará después, las cubiertas totales que debe producir del material actual y las que lleva fabricadas, el número de cubiertas que deberá fabricar del siguiente material y la prioridad de la máquina frente a otras.
* **HUECOS:** Para cada una de las máquinas, almacena qué material tiene en cada hueco y qué cantidad.

1. Diseño procedimental

A continuación se muestra el funcionamiento del programa mediante diagramas de flujo.



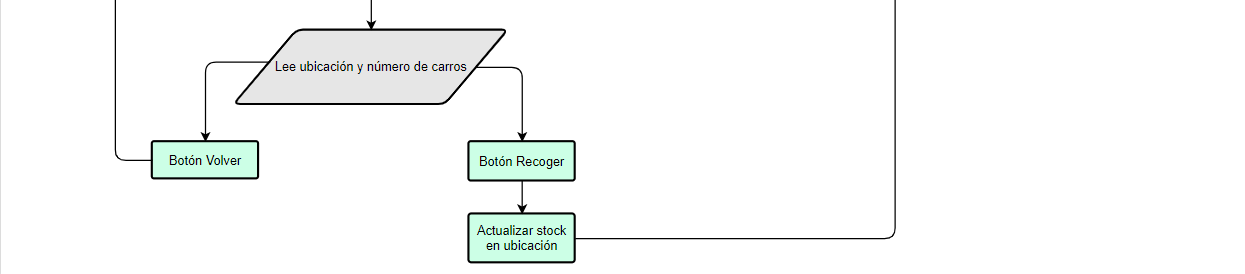


Figura C1 – Diagrama de flujo

1. Diseño arquitectónico

En este apartado se va a detallar la estructura del proyecto y cómo se ha diseñado.

El programa cuenta con dos partes bien diferenciadas: la parte de diseño y la parte de programación.

En la parte de diseño están los ficheros que contienen el diseño de la interfaz gráfica.

En la parte de programación se importa la parte de diseño. Existen dos ficheros que no necesitan de interfaz gráfica puesto que se ejecutan en segundo plano.

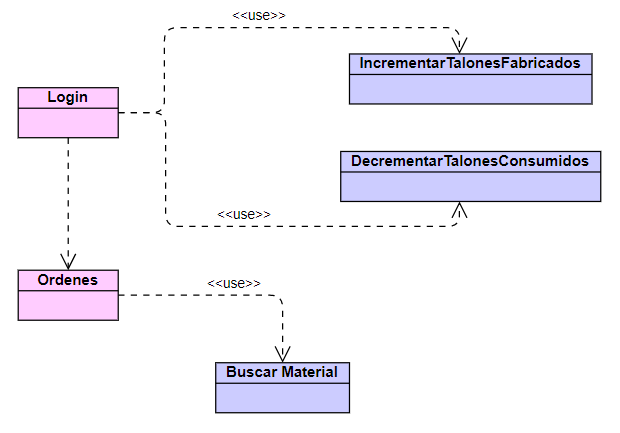


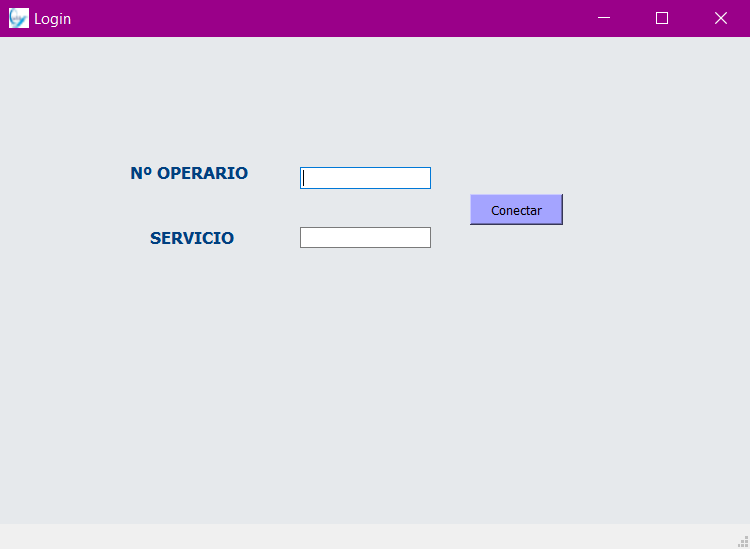
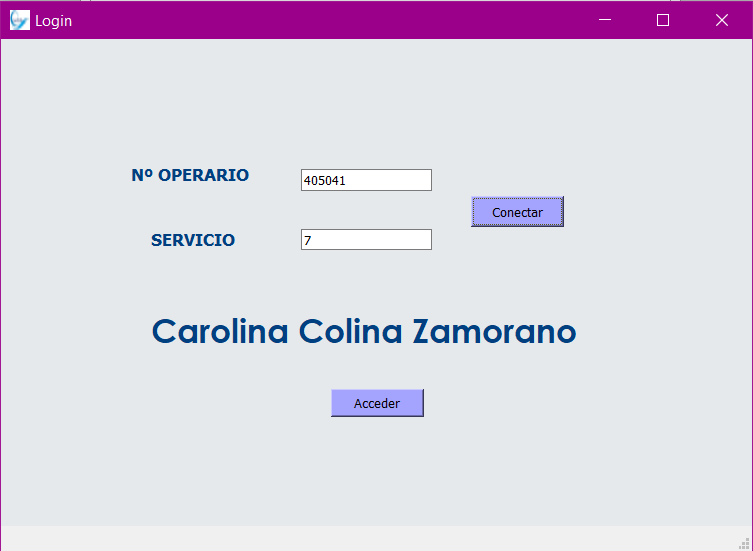
Figura C2 – Diagrama de clases

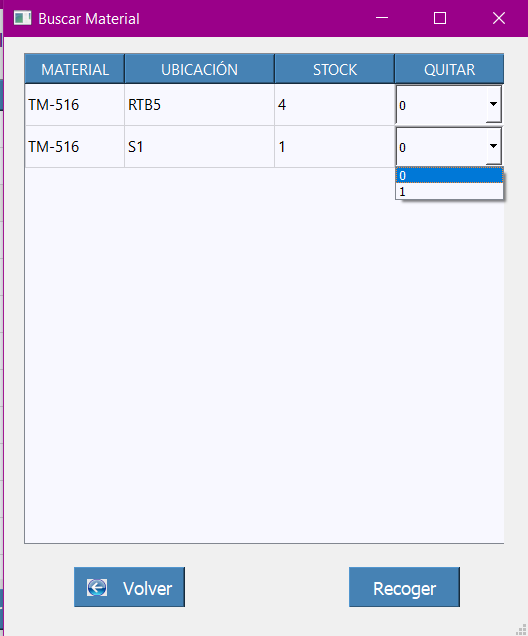
1. Diseño de interfaces

En el segundo sprint, en la tercera y cuarta semana se decidió cuál iba a ser el diseño de la interfaz.

Al tener decidido qué ventanas se iban a mostrar al usuario, cómo se iban a mostrar los datos y qué botones o funcionalidades iba a tener, facilitó el proceso de desarrollo de la aplicación.

Una vez introducido el número de operario y el servicio, se muestra el nombre y el botón Acceder que nos lleva a la ventana órdenes. En dicha ventana existe la opción de buscar material el otra ubicación distinta a la que proporciona la orden.





**Documentación técnica de Programación**

1. Introducción

En este apartado describiremos cómo se estructuran los directorios, el manual del programador y lo necesario para poder ejecutar el proyecto.

1. Estructura de directorios

El proyecto consta de cuatro directorios:

* **Documentación:** En este directorio se incluye la memoria del proyecto, el anexo de la memoria, un documento que explica el proceso productivo para una mejor comprensión del programa y una hoja de cálculo donde se hace una simulación de la situación inicial de la base de datos antes de arrancar el programa y que se haga ningún cambio.
* **Imágenes:** Aquí se alojan todas las imágenes que se usan en el proyecto.
* **sqlite:** En este directorio se encuentra la base de datos.
* **src:** Este directorio contiene el código del programa.

1. Manual del programador

**INSTALACIÓN DE PYTHON**

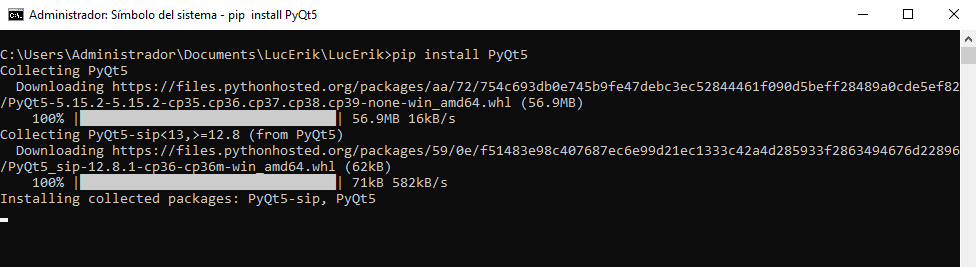
Para poder ejecutar nuestro programa es necesario tener instalado Python. Para ello, accedemos a la siguiente dirección: <https://www.python.org/downloads/> y descargamos la última versión.

Una vez descargado, se ejecuta y se instala.

Durante la instalación, es importante marcar la casilla para añadir la ruta de instalación al PATH de Windows.

**INSTALACIÓN PyQt5**

Es necesario instalar la librería PyQt5 para Python. De esta manera se importa lo necesario para que la interfaz gráfica se ejecute correctamente.



1. Compilación, instalación y ejecución del proyecto

Para poder ejecutar el proyecto, es necesario descargar el repositorio de GitHub que se encuentra en <https://github.com/CarolinaCCZ/LucErik>, una vez descargado se descomprime y está listo para usar.

Para el desarrollo del proyecto se ha usado PyCharm, aunque su instalación no es obligatoria puesto que se puede ejecutar desde la consola.

**IMPORTAR Y EJECUTAR EN PYCHARM**

En el caso de que se desee usar esta herramienta, es tan sencillo como abrir el proyecto: *File -> Open* y seleccionamos la carpeta donde se encuentra nuestro proyecto.

Como ya hemos instalado la librería PyQt5 no nos pedirá instalarlas de nuevo.

Para que al ejecutar coja el fichero que debe ejecutarse en primer lugar, seleccionamos *Run -> Configuration* y nos aseguramos de que en *Script path* se encuentre el fichero Login.py y en *Working directory* la ruta de nuestro proyecto.

**EJECUTAR EL PROYECTO**

Una vez importado el proyecto en PyCharm, sólo queda ejecutarle (*Run*).

En el caso de que queramos ejecutarle desde la consola, es necesario ubicarnos en la ruta del proyecto (*cd Path*) y con el comando p*ython Login.py* se ejecutará sin problema.

Existe otra opción que es ejecutarlo desde la máquina virtual que se proporciona.

En el escritorio se encuentra un acceso directo al proyecto con nombre LucErik.

Usuario: Administrador

Contraseña: LucErik2020

**Documentación de Usuario**

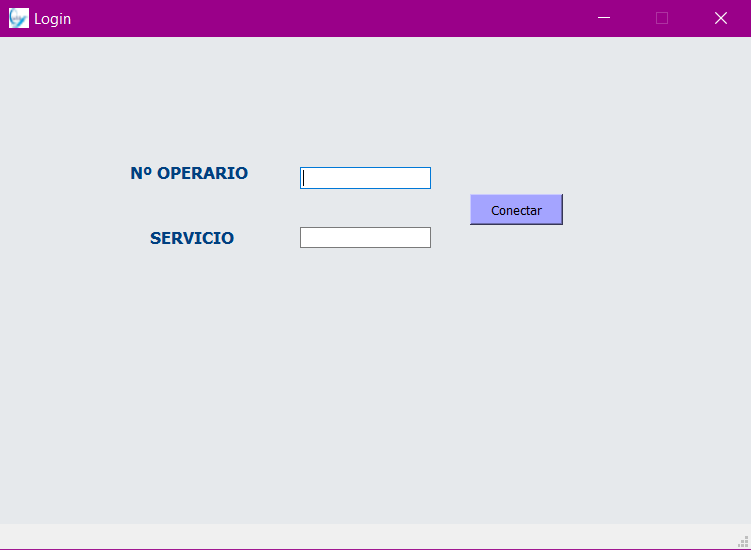
1. Instalación

El usuario no va a tener que instalar nada para poder ejecutar LucErik. Sólo tendrá que ejecutar el archivo que se le proporciona.

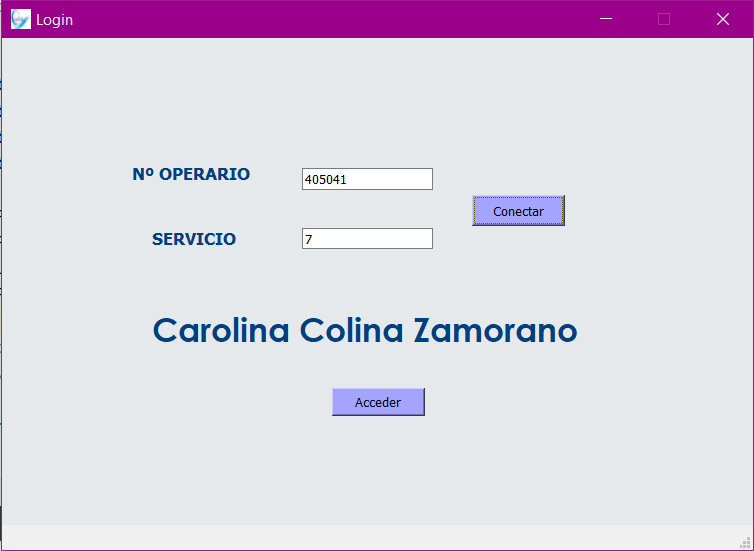
1. Manual de usuario

**LOGIN**

Esta es la primera ventana que se muestra al ejecutar nuestro programa:

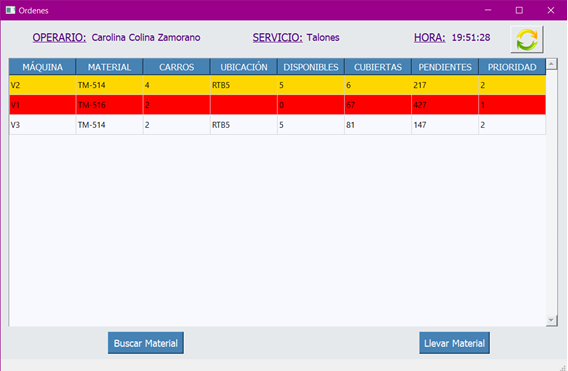


Ahora el usuario debe introducir su número de operario y el número del servicio. Si las credenciales introducidas son correctas, se mostrará el nombre del usuario y el botón Acceder estará disponible.



**ÓRDENES**

Una vez pulsado el botón Acceder, se abre la ventana órdenes.



En esta ventana se muestran las órdenes, las cuales indican qué material hay que llevar, a qué máquina y donde se encuentra dicho material.

Para cada una de las órdenes se muestra la siguiente información:

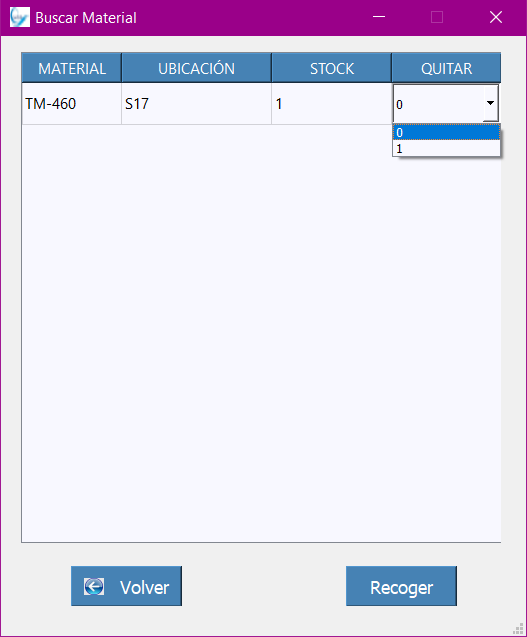
* **Máquina:** Máquina que necesita el material.
* **Material:** Código del material.
* **Carros:** Total de carros que hay que llevar a la máquina en función de los carros que necesita y los que caben en la máquina.
* **Ubicación:** Localización del material en las RTBS. Siempre va a mostrar la ubicación que más carros contenga.
* **Disponibles:** Cantidad de carros disponibles en esa RTB.
* **Cubiertas:** Cantidad de cubiertas que puede hacer con el material que tiene actualmente en la máquina.
* **Pendientes:** Cantidad de cubiertas que aún le quedan por hacer,
* **Prioridad:** Prioridad que se le da a la máquina en función de su importancia.

Como se muestra en la imagen las órdenes que se muestran adoptan distintos colores. Aquellas que tienen el color amarillo significa que el material disponible en la máquina le permite hacer 20 cubiertas o menos, por lo que es necesario llevar el material cuanto antes. Las órdenes de este color, siempre se encuentran en los primeros lugares de la tabla por ser las más prioritarias.

Las órdenes que se muestran en rojo, significa que no existe material de esa medida en ninguna RTB. Es posible que exista ese material en otras máquinas.

Para saber si existe material de esa medida en otra ubicación que no sea en las RTBS, es decir, en otra máquina, la aplicación nos da la posibilidad de buscarlo pulsando en el botón *“Buscar Material”*.

**BUSCAR MATERIAL**



En esta ventana se muestra un listado de ubicaciones donde existe el material actualmente. Como se ve en la imagen, en la máquina S17 existe el material TM-460.

El stock nos muestra el máximo de carros que podemos retirar de la máquina, puesto que no podemos dejar una máquina vacía para llevar material a otra. En este caso, la máquina S17 dispondrá de 2 carros y uno de ellos se estará usando, por lo que no le podemos retirar.

Si pulsamos el botón *Volver*, retornamos a la ventana de órdenes sin hacer ninguna función.

Si seleccionamos de la lista desplegable el número de carros que vamos a retirar y pulsamos el botón *Recoger*, se actualizan los huecos de la máquina de la que lo retiramos y de la máquina a la que lo llevamos.

Se cierra la ventana para buscar material y volvemos a la ventana órdenes.

En este momento, es necesario actualizar las órdenes, para ello el operario deberá pulsar el botón *Actualizar*, que se encuentra en la parte superior derecha de la ventana.



La ventana órdenes no se actualiza automáticamente hasta que el operario se asegure de haber llevado el material a la máquina correcta, de esta forma la orden no se borra de la lista hasta que el operario actualiza.

**Se puede encontrar el manual de usuario en**

Seguridad Social

Python

Wikidot

online.visual-paradigm