

## MEMORIA ESCRITA DEL PROYECTO

CFGS Desarrollo de Aplicaciones Multiplataforma

# GestResi

**Autor: Moisés González López**

**Tutor: Armando Cea**

**Fecha de entrega: 28/04/2023**

**Convocatoria: 2ºSemestre-2022/2023**

**Documentos del proyecto:**

**[https://drive.google.com/drive/folders/1WpYRTHxcq00O6ae5sUygl4RcqC9b9UPT?usp=share\\_link](https://drive.google.com/drive/folders/1WpYRTHxcq00O6ae5sUygl4RcqC9b9UPT?usp=share_link)**



## Índice de contenidos

<b>1. INTRODUCCIÓN .....</b>	<b>3</b>
1.1. Motivación .....	3
1.2. Abstract.....	4
1.3. Objetivos propuestos (generales y específicos) .....	4
1.4. Contexto laboral.....	4
<b>2. METODOLOGÍA USADA .....</b>	<b>6</b>
<b>3. TECNOLOGÍAS Y HERRAMIENTAS UTILIZADAS EN EL PROYECTO.....</b>	<b>9</b>
<b>4. ESTIMACIÓN DE RECURSOS Y PLANIFICACIÓN .....</b>	<b>11</b>
<b>5. ANÁLISIS DEL PROYECTO .....</b>	<b>13</b>
<b>6. DISEÑO DEL PROYECTO .....</b>	<b>20</b>
<b>7. DESPLIEGUE Y PRUEBAS .....</b>	<b>31</b>
<b>8. CONCLUSIONES .....</b>	<b>37</b>
<b>9. VÍAS FUTURAS.....</b>	<b>38</b>
<b>10. BIBLIOGRAFÍA/WEBGRAFÍA.....</b>	<b>40</b>
<b>11. ANEXOS .....</b>	<b>42</b>
11.1. MANUAL DE USUARIO .....	42
11.2. IMAGENES .....	44

# 1. Introducción

GestResi es un proyecto que se presenta como aplicación de escritorio, creado en JAVA a través del IDE NetBeans y que se construye con una interfaz gráfica sencilla para un fácil manejo de la gestión de datos, de disponibilidad de productos y de cuidados, realizados por el personal de una residencia, para el mejor manejo de la información entre el personal que trabaja en la residencia.

## 1.1. Motivación

Esta elección está motivada debido a la fuerte relación familiar y de amistad con trabajadores de residencia los cuales me han hablado de la carencia de este tipo de programas y teniendo en cuenta el gran desarrollo de este sector y las previsiones de futuro con una curva de envejecimiento de la población española y el aumento de la esperanza de vida.

Desde el inicio de la pandemia provocada por el SARS-CoV-2, y la afectación en especial a las personas mayores y que tenían como vivienda una residencia al convivir y hacer uso de estancias comunes, se ha hecho latente e imprescindible una mejor gestión de los recursos materiales de los cuales deben hacer un uso tanto los trabajadores como los usuarios de estos servicios residenciales, asimismo se debe hacer un esfuerzo superlativo para que el personal que trabaja en estos centros de atención cuenten con la información de forma sencilla y accesible, haciendo que se mejore también la gestión de los recursos laborales.

En la inmensa mayoría de estos centros la gestión de la información se da de forma manuscrita, dificultando, este hecho la comprensión por parte de todo el personal de lo descrito por otros compañeros en estos partes o informes manuscritos.

Mas allá de lo descrito anteriormente, puedo destacar que siempre me he sentido atraído por la mejora de los procesos y este sector está poco desarrollado en este aspecto.

Mi interés en este proyecto a parte de lo mencionado va también relacionado a la gran cantidad de centros de asistencia residencial que hay desplegados por todo el territorio y abre un gran mercado con muchas posibilidades de mejora.

El conseguir desarrollar un proyecto para facilitar la vida de estos trabajadores desde cero ha provocado en mí una gran satisfacción a la par que me ha creado un nuevo nexo

de unión con mis familiares y amigos, los cuales me han ayudado en el desarrollo y las pruebas de calidad.

## 1.2. Abstract

GestResi is a program created for Windows, which consists of a SQLite database, which provides care center workers with a simple graphical interface for managing the different information generated during a work shift.

Giving the possibility of creating an excel document with the material available in inventory.

It offers the possibility of creating, modifying, or deleting both the users of the center, the care offered, the products or medicines in the store, or even the distributors.

It also gives the possibility of generating a pdf report of the care and products used during the work shift.

## 1.3. Objetivos propuestos (generales y específicos)

Como objetivo general de GestResi es ofrecer a los trabajadores de los centros asistenciales residenciales un programa funcional y sencillo, que facilite el trabajo y la gestión de la información, que no requiera de amplios conocimientos técnicos para poder usarlo y que mejore el flujo de información entre los diferentes turnos.

Objetivos específicos se destacan:

- Facilitar instrucciones para poder trabajar con el programa
- Acercarse a la exposición de una idea ante posibles clientes
- Gestión de base de datos desde Java
- General Excel para inventario
- Poder elaborar informes de turno automáticos
- Generar un ejecutable para Windows

## 1.4. Contexto laboral

GestResi se puede considerar el punto de partida para la creación de un entorno más completo en el cual integrar nuevas funcionalidades y la elaboración de una interfaz gráfica más elaborada, en este aspecto destacar que este proyecto se han usado iconos e imágenes de uso libre y para próximas versiones se puede contratar personal

especializado como diseñadores gráficos para incorporarlas a una plantilla que se debe encargar del desarrollo y mantenimiento del programa.

Como primer escenario, y más posible, sería el ofrecerlo a las diferentes residencias y centros donde se tiene un contacto estrecho, para poder dar a conocer este programa y que estos mismos, sean los encargados de promocionarlos por los diferentes centros amigos y de los que forman parte de su confederación.

Asimismo, se mantendrían contactos con grupos de residencias a nivel nacional.

Se plante por último una aplicación móvil con la que los trabajadores, y creando un servidor con la base de datos puedan conectarse a la red wifi de la residencia y hacer las anotaciones pertinentes desde sus dispositivos móviles, garantizando así la seguridad y la ley de protección de datos, al hacerlo inaccesible de otra manera, facilitando aún más el proceso y economizando tiempos, esta aplicación podría suponer un gasto bajo en las diferentes tiendas de aplicaciones (app Store y Google play store) pero sería un gran aporte económico ya que sería mucho el personal que la descargaría, debido a la alta rotación en estos trabajos por la gran exigencia física y mental que tienen, y en caso de una buena implantación en el mercado y ello ayudaría a una mejora continua de las aplicaciones al crecimiento de la empresa y a la contratación de nuevo personal.

## 2. Metodología usada

Desde un principio he optado por la utilización de una de las metodologías Agile más asentadas en la actualidad, la metodología Kanban, ya que su utilización en mi trabajo está muy extendida y la implantación es de largo recorrido, su estilo visual y altamente entendible es muy útil en este tipo de desarrollo. Al representar las tareas por medio de tarjetas, los cuales representan los tres estados de las mismas, representados por colores como son, las tareas pendientes, en curso y por realizar.

Estas tareas se representan en las columnas y conforme cambian su estado se cambia el color y la columna que ocupa para una mayor y mejor visibilidad de los avances del proyecto, a la vez que el cambio de un color fuerte a uno más suave en las tarjetas hace que el humor mejore, por la posible finalización del proyecto.

He empleado, para llevar a cabo esta metodología la página KanbanFlow que al tratarse de una aplicación web gratuita y simple se adapta mejor al proyecto. Cuenta con una interfaz gráfica intuitiva y amigable, haciendo que la curva de aprendizaje sea poco pronunciada, es posible usarla incluso cuando jamás se ha tenido un contacto previo, aun estando en ingles no reporta problema alguno.

En esta aplicación web contamos para realizar el tablero con cuatro columnas, a las mencionadas anteriormente agregamos una columna para tareas diarias, la cual me ha ayudado mucho en la consecución de objetivos y la organización durante el desarrollo del proyecto.

La planificación durante el proyecto es muy importante para mí ya que he debido conjugar el proyecto, las asignaturas, el trabajo, la vida familiar y el desarrollo de las distintas competencias en las que estoy embarcado, haciendo que haya podido sobrevivir.

Las fases en las que se ha dividido el ciclo de vida de la aplicación son la planificación, el desarrollo, el seguimiento y el cierre de los cuales hablare a continuación:

- **Planificación:** En este primer paso, se inició incluso anteriormente a recibir el apto a la por parte del Ilerna, donde los trabajadores de residencia me comentaron de la necesidad de una aplicación para llevar a cabo el trabajo, con diferentes reuniones me dieron a conocer las funcionales esenciales y empecé a hacer una planificación usando Kanban para organizarme, durante este etapa también se eligió, la base de datos, sus tablas iniciales, el idioma a utilizar así como el IDE con el cual me sentía

más cómodo, así mismo busque recursos e imágenes de uso libre para el uso en la interfaz gráfica. Asimismo, se tuvo muy en cuenta los tiempos para poder entregar a tiempo el proyecto.

- Desarrollo: Es el núcleo del proceso y en el realizó el código de la aplicación, la interfaz gráfica y la elaboración de la memoria escrita. Como explicare más adelante en esta memoria, se fue desarrollando el proyecto de con un modelo iterativo Incremental, para de esta manera encontrar posibles fallos en el código o en las funcionalidades que generarían un mayor retraso en el proyecto en caso de estar mucho más avanzado.
- Seguimiento: Como he comentado en el punto anterior se han realizado pruebas durante el proceso de codificación, antes de dar por clausurado el proyecto y la aplicación, he realizado unas pruebas antes de la entrega final, pero se debe seguir con el proceso de control una vez el usuario lo tiene a disposición ya que el usuario final es capaz de usarlo de maneras que el desarrollador no se había planteado, para ello una vez obtuve el ejecutable lo distribuí por diferentes equipos y usuarios para que probaran las funcionalidad completa y se ajustaba a los objetivos marcados en primer término.
- Cierre: Tras las pruebas y comprobada la ausencia de errores o comportamientos indeseados, que implicaran el no disponer de un programa estable. En este momento se cierra el desarrollo del proyecto y se pasa una fase de mantenimiento en el cual poder pensar en la implementación de nuevas funcionalidades.

Durante la fase de desarrollo de este proyecto, descrita anteriormente he elegido el modelo iterativo incremental debido a el profundo desconocimiento de llevar un proyecto de software des de cero y con la facilidad que da Java y su desarrollo en clases, he optado por este método para ir probando las diferentes capacidades del Software conforme se va desarrollando y poder saber hasta donde se ha llegado con seguridad.

Con este modelo divido el programa en bloques o módulos los cuales se van diseñando, y probando antes de agregar un nuevo módulo o funcionalidad, usando el módulo o módulos anteriores como inicio para el siguiente, con ello consigo aislar los errores y facilitar la detección de estos, como he comentado la falta de experiencia me ha hecho ser precavido y diseñar bajo este modelo ya que la acumulación de errores o comportamientos no deseados en el código al llegar al final del proyecto podrían hacer desechar un proyecto valido.

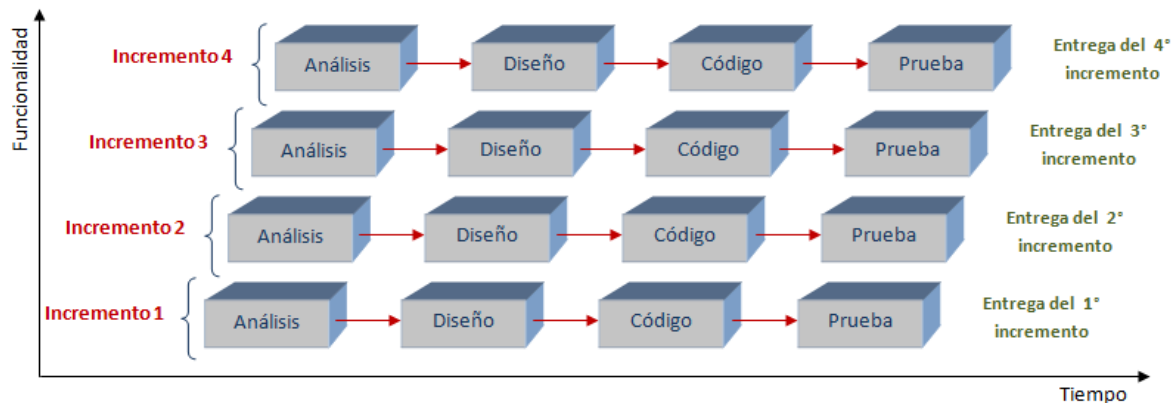


Figura 1: El Modelo Incremental

Representación del modelo iterativo incremental. Fuente: Librado, Bonifacio. Gervacio. (Septiembre 2013). Sistema de seguimiento al crédito(Tesis). Santiago de Queretaro: Tesis. [www.uteq.edu.mx/tesis/ITIC/0303.pdf](http://www.uteq.edu.mx/tesis/ITIC/0303.pdf).

Después de analizar las posibles desventajas y evaluar el nivel de seguridad, dado que esta aplicación se ejecutará de forma local, esta desventaja no supone un gran inconveniente, por lo que designaremos este riesgo como un riesgo asumido y seguiremos con este modelo de desarrollo.

También tendremos en cuenta que a partir de este software podemos ir desarrollando mejoras conforme queramos hacer crecer y seguir desarrollando las funcionalidades como hemos hecho durante el proyecto.

Los diferentes incrementos por los que he ido haciendo pasar mi aplicación han sido muchos, aunque los desglosare en los más llamativos y que coinciden prácticamente con las clases en las que está estructurado el proyecto.

1. Conexión con bases de datos.
2. Diseño de interfaz de acceso.
3. Registro, acceso a la aplicación.
4. Diseño de interfaz de usuario.
5. Registro de distribuidores.
6. Registro de usuarios.
7. Registro de almacén y cuidados.
8. Funcionalidad de generación de Excel.
9. Registro de turno.
10. Generación de datos turno y PDF.
11. Restringir funcionalidades dependiendo del tipo de usuario.



### 3. Tecnologías y herramientas utilizadas en el proyecto

Durante este proyecto se han utilizado diferentes tecnologías y herramientas las cuales desde el primer momento y ante todo he ido buscando la gratuidad de las herramientas y la calidad de estas al estar altamente contrastadas y ser de uso común en el mundo del desarrollo, asimismo he usado diferentes herramientas online para varios aspectos del proyecto, todas ellas gratuitas, aunque sí que en algunas de ellas ha hecho falta registrarse para poder hacer uso de algunas funcionalidades aunque están restringidas la capacidad total de ellas, como indicare más adelante.

Entramos al detalle con ellas:

- **Lenguaje JAVA:** Lenguaje de programación de alto recorrido que fue lanzado por primera vez en 1995 y que sigue vivo y en continua evolución que aumenta sus funcionalidades y la facilidad de programación gracias a las librerías, e integrado en gran variedad de productos.
- **Lenguaje SQL:** Con este lenguaje se consigue establecer consultas, actualizar, eliminar, agregar datos a una base de datos, a través de este lenguaje se agregan productos o tablas a la base de datos, es un lenguaje estandarizado, que se ha estudiado durante el curso.
- **Apache NetBeans:** Con este IDE (Entorno de Desarrollo Integrado) he llevado a cabo la codificación del proyecto ya que también posee la capacidad de realizar una interfaz gráfica y con capacidad para programar los diferentes elementos que vamos agregando según la necesidad, igualmente permite la integración de imágenes que faciliten la experiencia y el rápido aprendizaje en el uso de la aplicación. Gracias a varios cursos que he realizado en el que se usa este IDE, ha ayudado a que me haya decidido por este entorno y este lenguaje.
- **Eclipse:** Dado los problemas con los que me he encontrado para conseguir crear un .jar ejecutable con las librerías incluidas en NetBeans, introduje esta herramienta dando muy buen resultado.
- **MySQL Workbench:** Para gestionar la base de datos he optado por esta solución dado la facilidad de conexión entre esta y la programación JAVA. Desde el principio hay que contar con herramientas que faciliten esta integración, ayudado con la librería `java.sql.Connection`, podemos conectarnos a la base de datos desde el mismo IDE en las diferentes pruebas de integración incremental.

- **SQLite:** Una vez terminado el proyecto con MySQL y conociendo que la mayoría de residencias no cuentan con un personal con conocimientos técnicos para la implantación de una tecnología como MySQL y XAMPP. Y la gran capacidad que cuenta SQLite para una empresa mediana integre esta tecnología que crea la base de datos directamente al iniciar el programa.
- **WinRAR:** Usado durante el momento de compresión de los archivos para subirlos a Google Drive. WinRAR permite comprimir y descomprimir archivos en multitud de formatos de una forma rápida y sencilla.
- **Microsoft Office:** Durante todo el proceso se ha usado diversas herramientas de esta distribución ofimática, que por implantación es la más usada, entre las herramientas de esta distribución, las utilizadas para este proyecto, son Word para la elaboración de esta memoria, Excel para la su integración en la elaboración de inventarios y PowerPoint para el desarrollo de la presentación.
- **OBS Studio:** El desconocimiento de este tipo de herramientas me ha llevado a optar por la herramienta de edición y grabación de video, ya que permite grabar la pantalla y es ideal para la presentación del proyecto.
- **XAMPP:** Usado para crear una un servidor virtual a través del cual se conecta la app con la base de datos MySQL. También facilita el diseño, la creación y modificación de la base de datos y de sus tablas gracias a MyPHP.
- **Launch4j:** Tras conseguir crear el .jar ejecutable lo pase a .exe y para ello necesitaba esta tecnología.
- **Inno Setup Compiler:** En consecuencia, cree un instalador para que se creara una carpeta con los archivos necesarios y facilitar la instalación.
- **Kanban Flow:** Como he indicado anteriormente gracias a esta página ha sido muy fácil tener el control de las tareas diarias, así como las finalizadas, en proceso o por hacer, los colores y esta metodología han conseguido incluso la finalización del proyecto en tiempo y forma.
- **Canva:** Gracias a esta aplicación web que permite la creación de diagramas, esquemas y más formatos para lo que usado en la elaboración de esta memoria.
- **easyUML:** Plugin de NetBeans para conseguir crear diagramas UML, así como crear el diagrama de clases
- **Pixabay:** Esta página web me ha permitido encontrar recursos gráficos gratuitos y sin licencia para poderlos usar en el proyecto.

- **Paint:** aunque puede parecer que esta herramienta es anticuada, al tener la herramienta instalada de serie en mi sistema operativo he decidido aprovecharla para retocar, cortar o redimensionar imágenes para el proyecto, y al ser una herramienta básica y sin muchas funciones extras no hace falta un gran nivel técnico para usarla, de igual manera permite la utilización de imágenes en muchos formatos diferentes e incluso cambiar de formato.
- **Xiconeditor.com:** Es un editor en línea de iconos para poder crear imágenes en formato .ico usado por Windows para los accesos directos o el reconocimiento de una aplicación instalada.
- **Pngwing:** Se trata de una página en la que descargar logos en formato png, sin fondo, me ha sido muy útil en la presentación.
- **Adobe Acrobat:** Es el lector y editor de archivos .pdf líder y con el he podido consultar el temario de las asignaturas, así como las diapositivas y para a través de la edición de código de la aplicación consigo generar los informes de turno para una mejor visualización de los datos.

## 4. Estimación de recursos y planificación

Para este apartado se va a utilizar un diagrama de Gantt en el cual se puede observar el Tiempo estimado inicialmente para cada integración y haciendo un cálculo de tal forma que diera tiempo a hacer las diferentes entregas y centrándome en la entrega final ordinaria para la consecución efectiva del proyecto.

Para este fin he calculado una media de 3 horas diarias en las que he incluido tanto codificación, pruebas, el desarrollo de esta memoria, la realización de la presentación y tanto la grabación como el editado del video. Asimismo, se estimó que sobrara tiempo para hacer modificaciones y mejoras, así como posibles retoques, mas no conté con otros tiempos como reuniones previas.

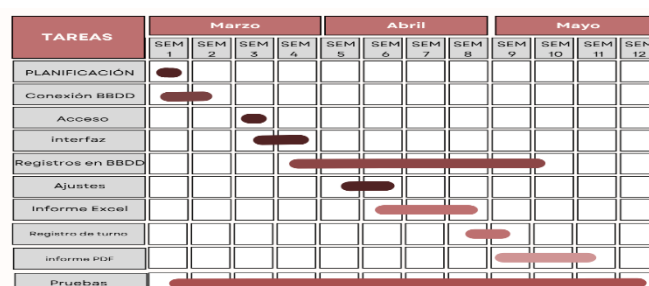


Diagrama de Gantt con el tiempo estimado. Fuente: elaboración propia en canva.com

Durante este tiempo y como he mencionado con anterioridad he usado el modelo iterativo incremental, lo que me ha permitido solucionar los errores que se han ido viendo en cada incremento, e incluso anteriores. Durante este proceso encontré un problema en el diseño de la base de datos, el cual no se notaba en las pruebas incrementales, pero ha hecho que empiece la codificación prácticamente de nuevo con su consiguiente incremento de tiempo.

Se añade también un diagrama de Gantt con los tiempos reales tras la finalización del proyecto. Este sin lugar a duda ha variado sobre todo ya que al aparecer problemas no deseados e inesperados en ejecución ha provocado que la dedicación sea mayor también cabe destacar que no conté bien el tiempo para la entrega ordinaria.

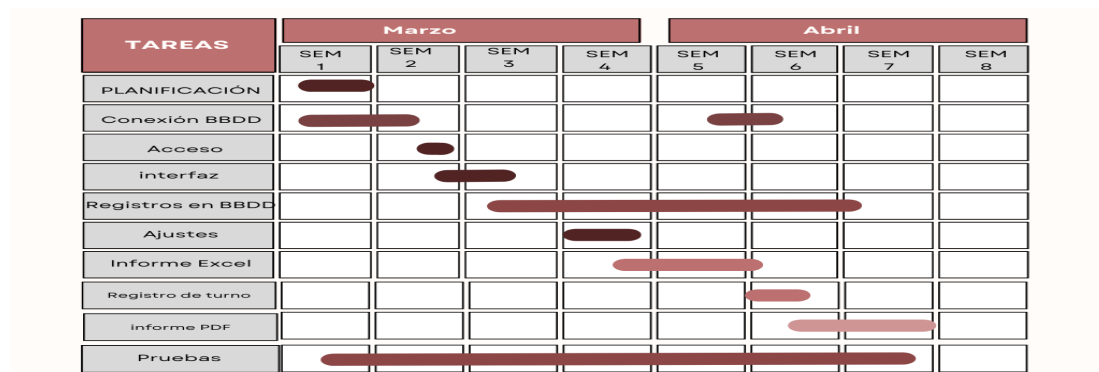


Diagrama de Gantt con el tiempo estimado. Fuente: elaboración propia en canva.com

Modelo de Marta borreguero

A parte de los recursos temporales hay que comentar que he tenido obviamente que hacer uso de tanto mi ordenador portátil con las siguientes características:

Procesador 11th Gen Intel(R) Core (TM) i7-1165G7 @ 2.80GHz 2.80 GHz

RAM instalada 16,0 GB (15,7 GB usable)

Tipo de sistema: Sistema operativo de 64 bits, procesador basado en x64

Una Tablet Huawei M5 y mi dispositivo móvil para poder conectarme en el trabajo ya que la red de mi trabajo es corporativa y no es para uso de dispositivos personales.

## 5. Análisis del proyecto

- **Requisitos Funcionales.**

Este cuadro representa los requisitos funcionales, los cuales aportaran capacidad a la aplicación para realizar diferentes tareas. Son una descripción clara de lo que un usuario puede esperar durante el uso de la aplicación y el entorno del sistema creado para tal efecto.

	Requisito	Descripción
RF1	Crear una aplicación funcional	Desarrollar una aplicación que pueda ser usado en gran variedad de equipos que no provoque errores.
RF2	Impedir errores de usuario	Dar las instrucciones claras para que el usuario no pueda hacer que se provoque errores, lanzando mensajes para que el usuario no se pueda equivocar.
RF3	Login	Crear un sistema para identificar usuarios
RF4	Menú	Crear un menú sencillo que de acceso a las diferentes pantallas
RF5	Pantallas	Creación de pantallas con las diferentes funcionalidades fácilmente identificables
RF6	Inventario	Posibilidad de crear una hoja Excel con el inventario disponible en el almacén para un control de este
RF7	Informe	Hacer que se puedan generar informes en PDF para visualizar el

		trabajo/ cuidado realizado por un empleado durante el turno
--	--	--

- **Requisitos no Funcionales.**

Estos requisitos no hacen referencia a funcionalidades que pueda disfrutar el usuario si no a la experiencia y fiabilidad.

	Requisito	Descripción
RNF1	Bajos requisitos	Desarrollar una aplicación que no requiera muchos requisitos de sistema
RNF2	Diseño	Conseguir que el diseño sea atractivo y relacionado con el entorno al que va dirigido
RNF3	Ejecutable	Conseguir que sea multiplataforma
RNF4	Mantenimiento	Dar soporte y seguir con el desarrollo de la aplicación para mejorarla
RNF4	Seguridad	Dado que los datos se almacenan en una base de datos local le confiere al sistema una seguridad y protección de datos

- **Diagrama Entidad Relación.**

Este diagrama representa la base de datos creada para la aplicación, sus tablas y las relaciones entre ellas. Las tablas contienen diferentes columnas que desde un principio se ha intentado nombrar para facilitar todo tipo de consultas, la visibilidad y la conexión con la aplicación.

Desde el primer momento se ha estado en contacto con personal de residencias para que la base de datos se ajuste a las necesidades reales del trabajo y para ello se ha creado este diagrama a partir de estas necesidades y los datos relacionados.

Después de la prueba continua y avance en el desarrollo de la aplicación se ha llegado a la conclusión que había que agregar una nueva funcionalidad para que los usuarios empleados no pudieran estropear la base de datos y es por ello que se crean roles en la tabla de empleados y así salvaguardar la privacidad y la integridad de los datos.

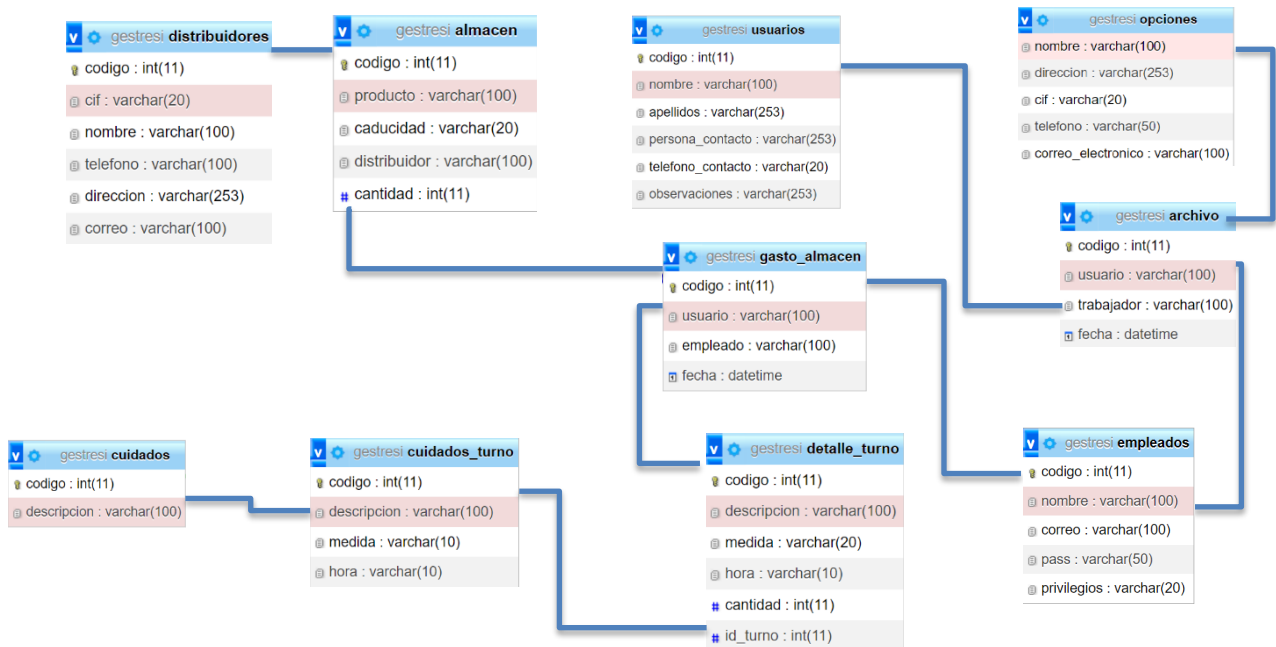


Diagrama ER. Elaboración propia con ayuda de phpMyAdmin.

## • Casos de uso

El objetivo de este diagrama es que se pueda revisar de un vistazo los posibles caminos que puede realizar una persona que use la aplicación, iniciando desde la pantalla de en la que debe ingresar un correo y contraseña y dependiendo de si tiene un rol administrador y empleado se les da acceso a diferentes funciones.

Como administrador, este le lleva a un menú principal, desde el cual puedes acceder a las diferentes pantallas Almacén, Usuarios, Opciones, Distribuidores, Cuidados, Turno, Archivo y Empleados, los cuales los cuales se conectan con la BBDD de manera bidireccional excepto el archivo que solo recibe información y genera informe de turno, entre tanto el usuario puede desde almacén generar un inventario, mientras el inventario se genera a partir de la base de datos, el informe del turno se genera desde de que se guarda el turno y se almacena en la carpeta de descargas del equipo, la consulta se realizara desde esa carpeta, la dirección de las flechas es algo importante a destacar en este diagrama y hay que tenerlo muy en cuenta a la hora de observar las posibles funcionalidades que se desarrollan a partir de cada paso que puede dar el usuario de la aplicación, asimismo cabe destacar que se puede

cerrar/salir de la aplicación en cualquier momento ya que las diferentes pantallas están integradas con el menú principal.

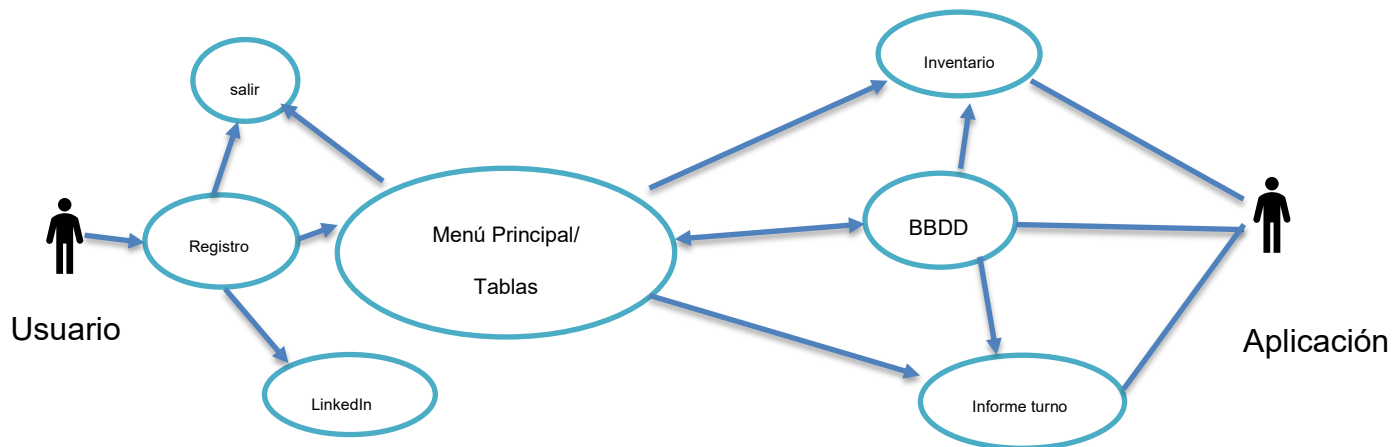


Diagrama de clases de uso. Elaboración propia desde el mismo programa Word

A continuación, y a partir de este diagrama podemos especificar los siguientes casos de uso para el usuario con privilegios de Administrador:

Especificación de caso de Uso	
Nombre	Ventana emergente
Descripción	Se muestra una ventana con el mensaje para dar un Empleado y contraseña inicial y poder hacer uso de la aplicación.
Secuencia Principal	Se abre la aplicación y se crea la base de datos lo que lanza un mensaje con el usuario administrador y la contraseña de este que será eliminado una vez creado el nuevo y eliminado el primero.

Especificación de caso de Uso	
Nombre	Login
Descripción	Se muestra una pantalla de inicio con dos campos para rellenar con correo y contraseña
Secuencia Principal	Se abre la aplicación y se carga la pantalla de registro si se introduce usuario y contraseña valida se entra al menú principal. Si no lanza un mensaje de error



Especificación de caso de Uso	
Nombre	LinkedIn
Descripción	Se abre enlace a mi página de LinkedIn
Secuencia Principal	Desde la pantalla de registro se puede acceder a esta página y será lo único que se podrá hacer sin un usuario y contraseña validos

Especificación de caso de Uso	
Nombre	Turno
Descripción	Desde el menú principal la primera pantalla que se abre para introducir los datos del turno
Secuencia Principal	Al abrir la pantalla se abre la pantalla para grabar los datos por usuario, tanto gasto de material como cuidados recibidos

Especificación de caso de Uso	
Nombre	Almacén
Descripción	Acceso a los datos de inventario como a la introducción de nuevos productos o modificación y creación de un inventario en Excel.
Secuencia Principal	Al clicar sobre el botón le da acceso a esta pantalla desde la que se puede gestionar el almacén con diferentes botones

Especificación de caso de Uso	
Nombre	Usuarios
Descripción	Acceso y consulta de los datos de los usuarios que pertenecen a la residencia, así como introducir nuevos, modificar los datos...
Secuencia Principal	Al clicar sobre el botón con este nombre se carga el listado de usuarios sobre una tabla existente.

Especificación de caso de Uso	
Nombre	Distribuidores
Descripción	Acceso y consulta de los datos de los distribuidores abastecen a la residencia, así como introducir nuevos, modificar los datos...
Secuencia Principal	Al clicar sobre el botón con este nombre se carga el listado de distribuidores sobre una tabla existente.

Especificación de caso de Uso	
Nombre	Cuidados
Descripción	Acceso y consulta de los datos de los cuidados que se ofrecen en la residencia, así como introducir nuevos, modificar los datos...
Secuencia Principal	Al clicar sobre el botón con este nombre se carga el listado de cuidados sobre una tabla existente.

Especificación de caso de Uso	
Nombre	Archivo
Descripción	Acceso y consulta de los datos de los turnos guardados y poder generar un informe en PDF para guardarlos
Secuencia Principal	Al clicar sobre el botón con este nombre se carga el listado de turnos sobre una tabla existente.

Especificación de caso de Uso	
Nombre	Empleados
Descripción	Acceso, consulta creación y eliminado de empleados asignándoles privilegios
Secuencia Principal	Al clicar sobre el botón con este nombre se carga el listado de Empleados sobre una tabla existente y a partir de los botones se puede eliminar o crear empleado.

A continuación, se detallan los casos de uso para un usuario con privilegios Trabajador:

Especificación de caso de Uso	
Nombre	Login
Descripción	Se muestra una pantalla de inicio con dos campos para rellenar con correo y contraseña
Secuencia Principal	Se abre la aplicación y se carga la pantalla de registro si se introduce usuario y contraseña valida se entra al menú principal. Si no lanza un mensaje de error

Especificación de caso de Uso	
Nombre	Turno
Descripción	Desde el menú principal la primera pantalla que se abre para introducir los datos del turno
Secuencia Principal	Al abrir la pantalla se abre la pantalla para grabar los datos por usuario, tanto gasto de material como cuidados recibidos

Especificación de caso de Uso	
Nombre	Archivo
Descripción	Acceso y consulta de los datos de los turnos guardados y poder generar un informe en PDF para guardarlos
Secuencia Principal	Al clicar sobre el botón con este nombre se carga el listado de turnos sobre una tabla existente.

- **Diagrama de clases**

Este diagrama representa las clases creadas en el entorno de desarrollo para crear las funcionalidades, en ellas se ve las clases objeto, una por cada una de las tablas de la base de datos en la cual se encapsulan las variables, las clases de tipo DAO (data access object) en las cuales se crean los metodos para darle funcionalidad a la aplicación, las clases de la interfaz donde esta la clase login la cual da la funcionalidad para el registro en el sistema y la interaccion de inicio, y la clase Sistema

que contiene todos los datos y funcionalidades de cada parte de la interfaz, y la interacción con el usuario, asimismo agregue en esta el método para crear pdf.

Después cuento con una clase conexión para crear la funcionalidad de conexión con la base de datos, la cual es indispensable y se encarga de crear la base de datos al iniciar el programa por primera vez, una clase llamada base de datos en la que creo las consultas sql para crear las diferentes tablas de la base de datos y crear el usuario admin para poder acceder a la base de datos y e introducir unos primeros datos en la tabla opciones para que no se creen mas de un registro en esta tabla.

Aparte de estas se crea una clase GenerarExcel para crear los inventarios en este formato y que conecta con la base de datos para extraer los datos y exponerlos.

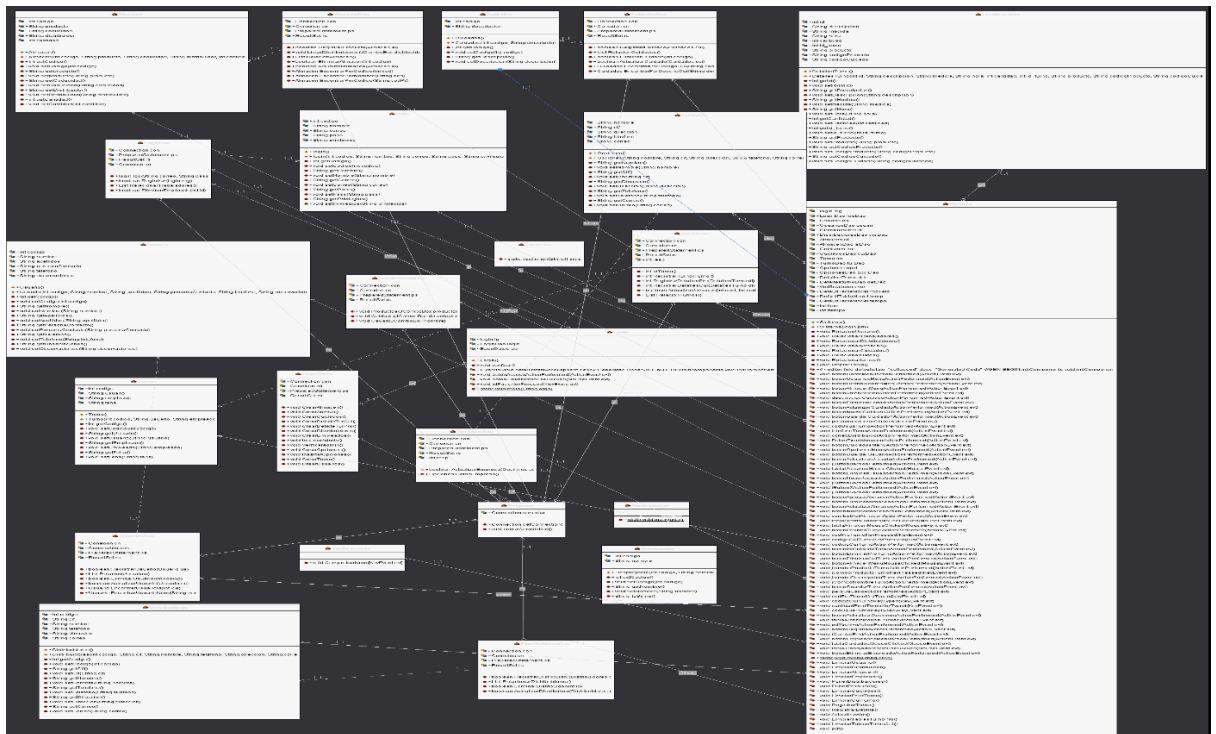


Diagrama de clases de GestResi. Fuente: realización propia en easyUML.

## 6. Diseño del proyecto

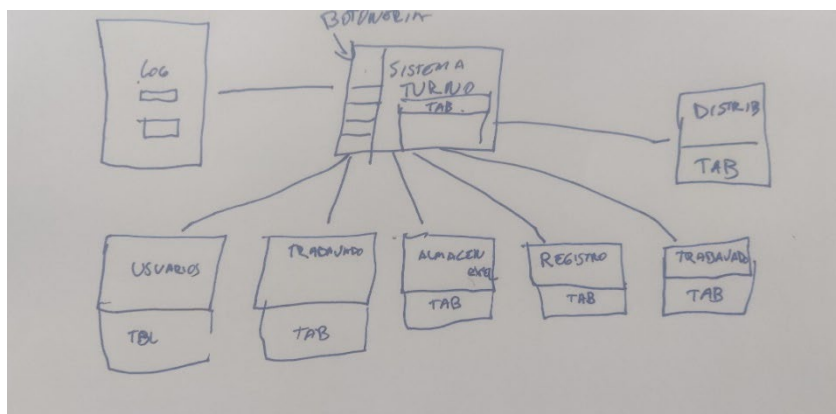
Desde el primer momento tome como diseño inicial uno fácil de entender y que no diera mucha oportunidad para el error en el uso de usuarios inexpertos y no sufrió cambios desde primer momento, tan solo se agregó una pantalla más y se cambió la configuración de la base de datos.

También Aproveche para decidir por que empezar a crear el proyecto, y aunque cambio en un momento avanzado empecé con el diseño de las diferentes tablas de nuestra base de datos con una serie de reuniones con personal trabajador de una residencia que me hablaban de necesidades básicas de la aplicación a desarrollar.

La estructura básica no sufrió cambios hasta que en uno de los momentos cuando diseñaba la estructura del turno me di cuenta que debía separar los cuidados de los productos para una mejor aplicación y estructura de la base de datos. Provocando el incremento de las tablas necesarias de 8 (almacen\_cuidados, archivo, detalle\_turno, distribuidores, empleados, opciones, turno y usuarios) pasaron a 10( almacén, cuidados, archivo, detalle\_turnopro, detalle\_turnocui, distribuidores, empleados, opciones, turno y usuarios), para almacenar mejor los detalles de turno a nivel productos de almacén y los detalles de cuidados ofrecidos por los trabajadores a los usuarios durante el turno y no provocar errores y datos nulos al ser conceptos diferentes.

Tras tener claro lo necesario para que fuese funcional la base de datos se creó el diseño y en el cual conté con ayuda y opiniones de trabajadores de estos centros sociosanitarios, que tras la primera muestra de este en modelo se da el visto bueno.

A continuación, muestro el diseño inicial en un pequeño croquis hecho a mano.



Diseño inicial de la interfaz de usuario GestResi. Fuente Elaboración propia a papel y bolígrafo.

Empiezo con el diseño de la interfaz de registro desde Apache NetBeans 12.6. Creando una clase llamada "Login" de tipo JFrameForm dentro del paquete interfaz y se agregan los componentes que enumero a continuación, un JLabel que ocupa tofo el frame para poner una imagen de fondo, la cual saco del banco de imágenes de Pixabay.com, para lo cual creo un Package donde voy a ir introduciendo las imágenes necesarias para el proyecto llamado "Img", tras ello agrego otro JLabel para introducir el icono de la aplicación, elaborado desde mi terminal móvil con la app Logo Maker, tras ello agrego un nuevo JPanel para agregar los diferentes componentes para poder interactuar con la app, estos son, dos JButton , uno para hacer el registro y uno para que de acceso a mi perfil de LinkedIn con el logo de la red social, un JTextField para que el empleado introduzca el correo, un JPasswordField para el acceso y tres JLabel, uno para una imagen extraída de Pixabay.com y editada con el programa de Windows Paint, para que se vea que el programa está bloqueado si no hay registro y los otros para agregar la descripción de los JTextField y JPasswordField.



Diseño de la interfaz de registro GestResi. Fuente Elaboración propia

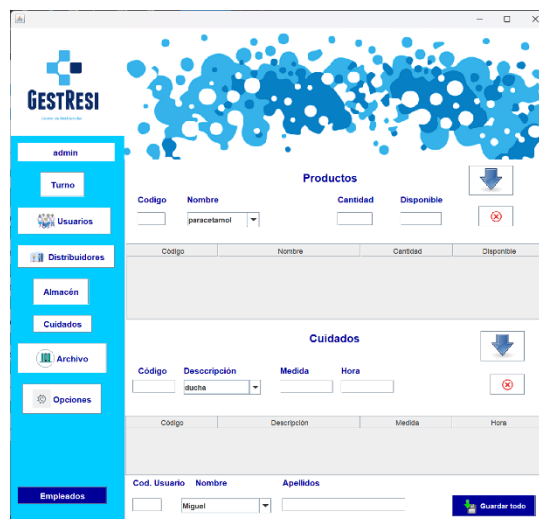
Después Empecé con el diseño de un nuevo JFrameForm al que llamé Sistema ya que es el que soporta la interfaz principal de interacción del empleado con la aplicación.

En la cual agregué un JPanel lateral en el cual incluí diferentes elementos como son ocho JButton, uno para el acceso a cada apartado de la aplicación (Turno, Usuarios, Distribuidores, Almacén, Cuidados, Archivos, Opciones y Empleados), un JPanel con un JLabel para la imagen del logo y un JTextField en el cual ira el nombre del Empleado que haga los registros.

A continuación, agregué un JLabel con la imagen decorativa para la parte superior, también extraída del banco de imágenes de Pixbay.com. y editada con Paint de Windows, en este punto quiero hacer un inciso para indicar que la paleta de colores de la aplicación fue elegida a partir de la creación del Logo y consultada con los trabajadores de la residencia.

Y agregue un JTabbedPane con 8 pestañas correspondientes a los botones del lateral. Los cuales describo a continuación.

- Pestaña Turno: compuesta por trece JLabel utilizadas para describir las zonas y los diferentes JTextField y los JComboBox, diez JTextField, dos de ellos invisibles desde la app final son todos editables pero al introducir código en alguno de ellos edita automáticamente otros, tres JComboBox para facilitar la búsqueda de productos, cuidados o usuarios, dos JScrollPane uno para los productos que se usen durante el turno y otro para los cuidados administrados durante este y cinco botones, uno para introducir productos en la tabla correspondiente, otro para eliminar los productos de la tabla en caso necesario, otro botón para introducir cuidados en su tabla y el respectivo para eliminarlo, y por ultimo un botón para guardar el turno y generar un archivo .pdf con la información del turno. A continuación, agrego una imagen de la pantalla principal con el menú y la pestaña turno.



Diseño menú inicio y pestaña turno de registro GestResi. Fuente Elaboración propia

- Pestaña Usuarios: esta pestaña hace referencia no a los usuarios de la aplicación si no a los usuarios de las residencias a las que va dirigida. Esta pestaña cuenta con cinco JLabel para describir cinco de los seis JTextField, ya que uno (codigousuario) permanecerá oculto al usuario final de la aplicación, estos y los visibles serán Nombre, Apellidos, Persona de contacto, Teléfono de contacto y Observaciones, también cuenta con una JTable donde se observa la relación de usuarios ya guardados y que al seleccionar un usuario rellena los JTextField, y cuatro botones los cuales he agregado imágenes para una mejor identificación, estos son uno para guardar el usuario, para eliminar el usuario, para actualizar los datos y para eliminar los campos

de los JTextField y poder introducir un nuevo usuario, es un diseño simple y fácil de comprender. Muestro pantalla de referencia



Diseño menú inicio y pestaña turno de registro GestResi. Fuente Elaboración propia

- **Pestaña Distribuidores:** esta pestaña es prácticamente idéntica a la anterior cuenta con cinco JLabel que identifican a cinco de los seis JTextField ya que uno permanecerá oculto para el usuario final, una JTable para mostrar el listado de usuarios, y cuatro botones los cuales dan funcionalidad a este apartado como son guardar distribuidor, actualizar eliminar y borrar los campos para poder introducir un nuevo distribuidor.
- **Pestaña Almacén:** cuenta con cuatro JLabel las cuales describen tres de los cuatro JTextField, uno permanece oculto, el JLabel restante describe un JComboBox el cual toma referencia de nombre de los distribuidores, es autocompletable, asimismo, agrego una JTable para poder visionar los productos en el almacén y al seleccionar uno de estos productos en la tabla los JTextField se rellenaran, así como los mismos botones de las pestañas anteriores para guardar, actualizar y eliminar productos, limpiar los campos para poder agregar nuevo e introduzco un nuevo botón para generar un informe inventario en Excel.
- **Pestaña Cuidados:** esta pestaña solamente cuenta con un JLabel para describir uno de los dos JTextField como en pestañas anteriores he ocultado uno para el usuario final, una JTable para ver los cuidados disponibles, y que al seleccionar uno rellene los campos de los JTextField, y cuatro botones para guardar, actualizar y borrar cuidado, así como un botón para eliminar el contenido de los campos y poder introducir cuidados nuevos.

A continuación, muestro las tres pantallas





Diseño menú y pestañas Distribuidores, Almacén y cuidados GestResi. Fuente Elaboración propia

- Pestaña Archivo: esta es la más simple ya que solo cuenta con una JTable que relaciona los turnos guardados y un botón para consultar el pdf generado al guardar el turno que seleccionemos de la JTable.
- Pestaña Opciones: en esta pestaña es la destinada a tener los datos de la residencia para poderlos agregar en su momento al pdf del turno. Cuenta con diferentes campos para agregar los datos de la residencia y solo cuenta con un botón de actualizar para que no se generen más registros en la base de datos.
- Pestaña Empleados: esta Pestaña fue la última en agregarse y se agregó al final de la codificación del código pero la agrego en este apartado para que estén todos los elementos visuales del programa juntos cuenta con cuatro JTextField para la identificación de empleados y creación de rol de cada uno de ellos, distinguiéndose en dos y agregándolos a un JComboBox para que no haya error a la hora de identificar los roles y asignarlos a los empleados uno de los JTextField permanece oculto al usuario final de la aplicación, cuenta con dos botones uno para agregar un nuevo usuario y otra para eliminar usuario, asimismo, contiene una JTable para visualizar la relación de empleados y su rol.

A continuación, se muestran las pantallas restantes.



Diseño menú y pestañas Archivo, Opciones y Empleados de GestResi. Fuente Elaboración propia

Para empezar con el momento de la codificación del proyecto y antes que nada Descargué el programa XAMPP y realice la instalación, para crear la base de datos,

la cual generé toda, desde la conexión local generada por la aplicación y la facilidad de uso con apenas necesidad de crear consultas en la creación de nuestra base de datos MySQL, desde donde creé cada una de las tablas y asigné el tipo de dato de cada una de ellas, a partir de esta conexión.

### **Clases:**

1. **Clase Conexión:** Empiezo con esta clase ya que sin ella no puedo crear la conexión con la base de datos y conseguir ninguna funcionalidad en mi aplicación, su única funcionalidad es esta.
2. **Clase Usuarios:** En esta clase me encargo de encapsular los valores de la tabla de la base de datos con el mismo nombre. Se añaden los métodos getter y setter para poder acceder a ellos, así como los constructores uno vacío y otro con todas las variables.
3. **Clase UsuariosDao:** haciendo uso de la clase conexión creo los métodos que ejecutan las consultas en la base de datos, para crear usuario (RegistrarUsuario), para generar una lista de usuarios (RelacionUsuarios), para eliminar usuario (EliminarUsuario) pasándole el parámetro del código para identificar el usuario a eliminar, actualizar usuario (ActualizarUsuario), y dos métodos para encontrar el usuario por nombre y por código (estos métodos se agregan cuando se trabaja con la interfaz del turno).

A continuación, paso a la clase sistema para crear las funcionalidades llamando a esta clase y a la clase Usuarios, para poder llamar a sus métodos y a los valores encapsulados y agrego la funcionalidad al botón para crear usuario, picando sobre él se me abre el código para insertarlo, donde primero compruebo que los valores de los JTextField estén rellenos y en caso de no estar genere un mensaje para que el usuario final revise esto. Continuo con relacionar usuarios para generar el listado en la tabla, para ello uso DefaultTableModel y List de la clase java.util la cual le asigno el método "RelacionUsuario" creado en la clase. Continúo con actualizar el usuario donde agrego las comprobaciones pertinentes. Y que lance mensajes si las condiciones no se cumplen. Tras ello en Sistema creo los métodos para limpiarla tabla y limpiar los JTextField y los agrego a la funcionalidad del botón y después le agrego Relacionar usuario para cuando se actualice se quede vacío los campos y la tabla rellena con los usuarios actualizados. Después agrego el método EliminarUsuario al botón correspondiente, para ello agrego también que compruebe el código del usuario y que lance un mensaje de confirmación, igualmente agrego los métodos limpiar tabla limpiar usuario y relacionar usuario para que quede preparado para una nueva acción.

4. **Clase Distribuidores:** creo la clase para encapsular los valores de esta tabla en la base de datos, su constructor tanto vacío como con todas las variables, así como los métodos getter y setter para tener acceso a ellos desde fuera de la clase.
5. **Clase DistribuidoresDao:** Reutilizando el código de la clase UsuariosDao y cambiando los parámetros en las consultas SQL creo los métodos RegistrarDistribuidor, EliminarDistribuidor, ActualizarDistribuidor y RelacionarDistribuidor.

Como en el clase DAO anterior continué en la interfaz de Sistema e importé las clases para crear las funcionalidades de los botones y la tabla, copié y edité los códigos de los botones de la pestaña anterior, importando para ello las clases creadas creando mensajes personalizados de comprobación, se agrega el método limpiarTabla, asimismo, se crean los métodos RelacionarDistribuidores y Limpiar distribuidores para que cada vez que se haga una acción se muestre la lista de distribuidores en la tabla y los campos de los JTextField queden vacíos para poder efectuar una nueva acción.

6. **Clase Almacen:** creo la clase y encapsulo las variables de la respectiva tabla en la base de datos y creo los constructores tanto vacíos como con todas las variables y los métodos getter y setter
7. **Clase AlmacenDao:** en esta clase y copiando el código de la clase anterior se crean los métodos ActualizarAlmacen, EliminarAlmacen, RegistrarProducto y RelacionAlmacen, pero en este caso creo también un método para que con acceso a distribuidores estos se listen en el JComboBox, llamado ListarDistribuidores, como en la clase UsuariosDao se crean dos métodos para consultar por nombre (EncontrarPorNombre) y para consultar por código (EncontrarPorCodigo) y como en la anterior ocasión se crearon cuando se trabajó con la pestaña del turno.

Vuelvo a la clase de la interfaz sistema, importé las clases y lo primero que creo es el método Poner distribuidores para que se listen en el JComboBox (PonerDistribuidores) y en al añadir funcionalidad en los diferentes botones se usa SelectedItem para dar que seleccione el nombre de los distribuidores al crear o actualizar un producto del almacén, creo limpiar almacén para que se ejecute después de cada opción y creo las comprobaciones de condiciones para agregar nuevos productos, actualizarlos, eliminarlos y para poder agregar datos nuevos, uso también AutoCompleteDecorator para que al introducir algún carácter que coincida con el nombre de un distribuidor.

8. **Clase GenerarExcel:** creo esta clase dentro del paquete informes y para ello es importante importar `org.apache.poi`, `java.io`, `java.awt`, `java.text`, `java.sql`, `java.util` y `javax.swing`. de primeras introduje una imagen, pero tras los problemas a la hora de crear el archivo `.jar`, localizar la imagen o portarla para ponerla en el documento me generaba error y no conseguía crear el Excel por lo que dejé sin implementar la imagen, aunque se podrá actualizar esta parte en cualquier momento. Elijo la fuente Arial y el tamaño 14 para el título, creo un Array String para los títulos de las columnas le asigno la fuente Arial y tamaño 12 y le pongo un fondo azul y le agrego bordes, después lanzo una consulta y recorro los datos de la consulta para que rellene los datos en su respectiva columna. Por último, hago que guarde el archivo generado en descargas y le asigno un nombre el cual lo concateno con la fecha y la hora para que dé siempre un archivo nuevo. Al probarlo me doy cuenta que se ve muy pequeño por lo que le agrego un zoom a la hora de abrirlo. Tras ello lo agrego al botón correspondiente en Interfaz.Sistema, e importo la clase.
9. **Clase Cuidados:** En esta clase encapsulo, como en las clases anteriores los datos correspondientes de a la base de datos con el mismo nombre en este caso son menos solo dos e igualmente agrego el constructor vacío y el constructor con las variables además de los correspondientes métodos getter y setter para acceder a los datos desde otra clase.
10. **Clase CuidadosDao:** en Esta clase genero los métodos con conexión a la base de datos, `RegistrarCuidado` (para crear cuidado nuevo), `RelacionarCuidados` (para consultar todos cuidados guardados en la base de datos), `EliminarCuidado` (para eliminar un cuidado de la base de datos), `ActualizarCuidado` (para cambiar la descripción de algún cuidado), `EncontrarPorCodigoCui` y `EncontrarPorDescopCui`. Estos dos últimos se agregan cuando se elabora las funcionalidades de la pestaña Turno.  
  
Voy a la clase de interfaz Sistema, importo las clases y creo los métodos `RelacionarCuidados` y `LimpiarCuidados` y a partir de ahí empiezo a dar funcionalidades a los botones de la pestaña, creando condiciones que de no cumplirse mandara mensajes personalizados para no dejar campos vacíos, le doy también `mouseclick event` a la tabla como a las anteriores para que al seleccionarla se rellenen los campos de los `JTextField`.
11. **Clase Opciones:** en esta clase encapsulo los valores de la tabla de la base de datos con el mismo nombre y creo los constructores y los métodos getter y setter correspondientes.

**12. Clase OpcionesDao:** aquí solo creo dos métodos uno para actualizar los datos y otro para rellenar los datos de la empresa en los diferentes JTextField.

Me dirijo a la interfaz sistema, importo las clases, creo el método RelacionarDatos y agrego la funcionalidad al botón actualizar creando una condición que de no cumplirse no se actualice y lance un mensaje.

**13. Clase login:** En esta clase encapsulo las variables correspondientes a la tabla empleados y creo los constructores y los métodos getter y setter para acceder a estos datos.

**14. Clase LoginDao:** en esta clase creo los métodos para la gestión del acceso y de los empleados creo el método para conseguir el acceso al consultar el correo y la pass del empleado de la base de datos, un método para registrar nuevos empleados, para relacionar los empleados y para eliminarlos.

Paso primero a la interfaz login y creo el método verificar en el cual se comprueba la condición para que el correo y la password coincida y se lanza un mensaje de bienvenida y se da acceso a sistema, en caso contrario se lanza un mensaje para que se sepa que no coinciden. Tras ello voy a sistema y creo el método relacionar trabajadores para llenar la tabla. Relleno el JComboBox con los privilegios y creo Relacionar trabajadores para listarlos en la tabla le doy funcionalidad a los botones poniendo las condiciones pertinentes para que lance un mensaje en caso de faltar algún campo o pida confirmación para eliminar algún empleado.

**15. Clase Turno:** encapsulo las variables que se relacionan con tabla de la base de datos del mismo nombre, creo el constructor vacío y otro con todas las variantes menos la fecha ya que utilizare java.util para obtenerla, así, como los métodos getter y setter.

**16. Clase TurnoDao:** creo los diferentes métodos para poder crear turnos con los datos introducidos por los empleados, así como restar los productos gastados durante el turno, registrar los detalles tanto de la parte de productos como de cuidados con hora y medida en su caso.

Me dirijo a sistema y voy dando funcionalidades tanto a los botones que permiten introducir datos como los JComboBox y para que al introducir un código válido se rellenen los campos de productos, de cuidados o de usuario, le creo también evento a clicar mouse sobre la tabla para poder eliminar un cuidado o un producto. Creo también el método para generar pdf y como en el caso del Excel introduzco una imagen, pero debo modificar el código al crearse un error al no encontrar la imagen a partir de su compilación en un ejecutable .jar por lo cual decidí quitarlo. Este método

se ejecutará al clicar sobre el botón guardar turno siempre y cuando los campos de cuidados, productos y usuarios estén rellenos

- 17. Clase DetallesTurno:** en esta clase encapsulo todos los elementos que están en las tablas detalleturnocui y detallesturnopro con sus constructores y sus métodos getter y setter para tener acceso a ellos y los uso en los métodos creados en TurnoDao.
- 18. Clase DetallesTurnoDao:** en esta clase creo los métodos para rellenar los JComboBox de la pestaña turno.
- 19. Clase Verificaciones:** creo la clase para generar un método para impedir que escriban letras en un lugar determinado para números y evitar que se creen problemas con la base de datos.
- 20. Clase Principal:** creo esta clase main para lanzar el programa.

En sistema añado funcionalidad para que al clicar sobre los botones laterales del menú y que se cargue la información en la tabla correspondiente a la pestaña, tras ello agrando el JPanel que contiene la imagen de decoración de la cabecera para que tape las pestañas y solo sean accesibles a través de los botones.

#### **Privilegios:**

Tras la comprobación del proyecto añado en la interfaz sistema la condición con una sobrecarga de método para que para bloquear todos los botones excepto Turno y Archivo para los empleados no administrador y hago que se muestre el nombre del empleado en el JTextField destinado para el efecto y desde el que se consigue el nombre del empleado para el turno.

#### **Cambio a SQLite:**

Tras finalizar el proyecto y comprobar la dificultad para un usuario final del programa en la instalación de la base de datos y su gestión desde fuera de la aplicación, consulte como crear la conexión e integrar SQLite al proyecto y dado que las consultas eran compatibles, la facilidad ya que se genera automáticamente la conexión y la mayoría de los datos se generan en un documento pdf que se archiva en Descargas. Para una residencia es más que suficiente las capacidades que ofrece este tipo de base de datos, para lo cual duplique el proyecto eliminé la clase conexión y cree un package llamado com.mycompany.sqlite y dentro la clase Conexión. Para conservar la mayor parte del código y que coincidiera con el nombre anterior de la clase conexión, también cree una nueva clase en el package clases llamada BaseDatos para crear las tablas automáticamente si no existen así como introducir un usuario con Rol Administrador y los primeros datos en la tabla opciones para que puedan actualizarse sin dar error, tras comprobar su buen funcionamiento con SQLiteStudio procedo a crear también un

mensaje que se mostrara tras el primer inicio y mientras el primer usuario con Privilegios Administrador no sea eliminado. Para que el usuario final pueda acceder y crear los primeros usuarios sin problemas.

**21. Clase BaseDatos:** Tras crear la conexión de la base de datos SQLite veo la necesidad de crear las diferentes tablas con sus atributos, y crear un usuario administrador con el primer inicio y rellenar campos de la tabla opciones que son los datos de la residencia para que se pueda actualizar desde la aplicación y no de problemas. Para ello creo esta clase con los métodos con las consultas necesarias que son llamados al ejecutar el programa. Asimismo, cabe destacar que en el trabajo con SQLite me fue necesaria el cierre de la conexión después de cada consulta mientras que con MySQL no era necesario.

## 7. Despliegue y pruebas

Como se ha especificado en anteriores apartados, para conseguir una mayor rapidez y facilitar la identificación de todo tipo de errores o comportamientos no deseados, he ido haciendo pruebas conforme se iba agregando código a la aplicación y llevando a cabo un modelo de desarrollo iterativo incremental, solventando y cambiando a cada momento que era identificado cada posible error durante todo el proceso de desarrollo del proyecto, para que la aplicación se ejecute sin problemas.

Una vez cambie la base de datos a una SQLite descargué SQLiteStudio para comprobar que se creaba tanto la base de datos, como las tablas y los registros en ella sin problemas, anteriormente mientras usaba MySQL utilice XAMPP para visualizar los cambios, esto me aseguraba una visión constante de que las operaciones de lectura y escritura se producían a la perfección.

Para mostrar de una forma más esquematizada las pruebas realizadas una vez finalizado el proyecto, muestro un cuadro para que se comprenda mejor.

Todas estas pruebas se han efectuado en modelo caja negra y desde diferentes equipos.

PRUEBA	DESCRIPCIÓN
1.Creacion Base de datos	<b>Objetivo Comprobado:</b> Se crea la base de datos al abrir por primera vez la aplicación



	<p><b>Requisitos Comprobados:</b> La creación de la base de datos en el inicio se ejecuta como estaba definido.</p> <p><b>Pruebas Efectuadas:</b> Se comprueba en el archivo de ejecución la creación de la base de datos.</p>
2. Mensaje de inicio	<p><b>Objetivo Comprobado:</b> Aparece el mensaje tras el primer inicio.</p> <p><b>Requisitos Comprobados:</b> el mensaje se sitúa en el sitio adecuado y muestra el texto indicado.</p> <p><b>Pruebas Efectuadas:</b> se ejecuta la aplicación por primera vez o se elimina la base de datos para que genere de nuevo el mensaje y la base de datos.</p>
3. Integración de los diferentes elementos como botones y tablas	<p><b>Objetivo Comprobado:</b> Los botones y elementos se mantienen en el lugar indicado</p> <p><b>Requisitos Comprobados:</b> Los elementos aparecen en los sitios designados en un principio.</p> <p><b>Pruebas Efectuadas:</b> Se ejecuta la aplicación y se comprueba la situación de los diferentes elementos en el conjunto</p>
4. Funcionamiento del botón de Registro	<p><b>Objetivo Comprobado:</b> El botón da acceso a los usuarios registrados.</p> <p><b>Requisitos Comprobados:</b> El botón ejecuta la consulta correctamente y da acceso a la siguiente pantalla.</p> <p><b>Pruebas Efectuadas:</b> Se introducen correo y contraseña de varios de los usuarios creados y se da al botón de registro</p>
5. Botón LinkedIn	<p><b>Objetivo Comprobado:</b> Se abre el perfil de LinkedIn.</p> <p><b>Requisitos Comprobados:</b> Al presionar el botón abre el perfil de LinkedIn al presionar el botón.</p> <p><b>Pruebas Efectuadas:</b> Se abre la aplicación y se presiona en el botón de LinkedIn.</p>
6. Nombre de empleado en interfaz	<p><b>Objetivo Comprobado:</b> El nombre aparece en la interfaz.</p> <p><b>Requisitos Comprobados:</b> El nombre coincide con el nombre de empleado que ha iniciado sesión.</p> <p><b>Pruebas Efectuadas:</b> Se hace el registro y se comprueba que en la localización aparece el nombre correspondiente.</p>
7. Roles o Privilegios Correctos	<p><b>Objetivo Comprobado:</b> El empleado tiene acceso según su Rol o Privilegio</p> <p><b>Requisitos Comprobados:</b> Los Botones están deshabilitados o habilitados según el privilegio que posea.</p>



	<p><b>Pruebas Efectuadas:</b> Se hace el registro y se comprueba que depende del usuario tiene habilitado una serie de botones u otros para que un empleado no autorizado no cambie cosas.</p>
8.Botones Menú	<p><b>Objetivo Comprobado:</b> Los botones funcionan correctamente.</p> <p><b>Requisitos Comprobados:</b> Cada botón abre la pestaña adecuada en la interfaz y efectúa el comportamiento esperado.</p> <p><b>Pruebas Efectuadas:</b> Se hace el registro y se comprueba que los botones del menú abren cada una de las pestañas cargando los datos en las tablas.</p>
9.Tabla de la Pestaña Distribuidores	<p><b>Objetivo Comprobado:</b> La tabla de la pestaña se completa con el listado de distribuidores y al clicar sobre uno los JTextField se rellenan automáticamente.</p> <p><b>Requisitos Comprobados:</b> al presionar el botón correspondiente en el menú se cargan los datos en la tabla y al seleccionar un elemento de la tabla esta rellena los campos.</p> <p><b>Pruebas Efectuadas:</b> Se selecciona el botón de Distribuidores y se observa cómo están rellenos los datos en la tabla y hago clic sobre alguno de ellos estos para comprobar que se rellenan los JTextField .</p>
10.Botones pestaña Distribuidores	<p><b>Objetivo Comprobado:</b> Los botones funcionan y crean, actualizan, eliminan distribuidores y limpian los campos de los JTextField.</p> <p><b>Requisitos Comprobados:</b> Se comprueba que no se produce ningún error al crear un distribuidor, actualizarlo o eliminarlo y se actualiza la tabla.</p> <p><b>Pruebas Efectuadas:</b> Se comprueba cada uno de los botones y los mensajes de advertencia creados durante un proceso de creación, actualización o eliminado de distribuidores.</p>
11.Tabla pestaña Usuarios	<p><b>Objetivo Comprobado:</b> La tabla de la pestaña se completa con el listado de Usuarios y al clicar sobre uno los JTextField se rellenan automáticamente.</p> <p><b>Requisitos Comprobados:</b> al presionar el botón correspondiente en el menú se cargan los datos en la tabla y al seleccionar un elemento de la tabla esta rellena los campos.</p> <p><b>Pruebas Efectuadas:</b> Se selecciona el botón de Usuarios y se observa cómo están rellenos los datos en la tabla y hago</p>

		clic sobre alguno de ellos estos para comprobar que se rellenan los JTextField.
12.Botones Usuarios	pestaña	<p><b>Objetivo Comprobado:</b> Los botones funcionan y crean, actualizan, eliminan distribuidores y limpian los campos de los JTextField.</p> <p><b>Requisitos Comprobados:</b> Se comprueba que no se produce ningún error al crear un distribuidor, actualizarlo o eliminarlo y se actualiza la tabla.</p> <p><b>Pruebas Efectuadas:</b> Se comprueba cada uno de los botones y los mensajes de advertencia creados durante un proceso de creación, actualización o eliminado de distribuidores.</p>
13.Tabla Almacén	Pestaña	<p><b>Objetivo Comprobado:</b> La tabla de la pestaña se completa con el listado de productos y al clicar sobre uno los JTextField y el JComboBox se rellenan automáticamente.</p> <p><b>Requisitos Comprobados:</b> al presionar el botón correspondiente en el menú se cargan los datos en la tabla y al seleccionar un elemento de la tabla esta rellena los campos.</p> <p><b>Pruebas Efectuadas:</b> Se selecciona el botón de Almacén y se observa cómo están rellenos los datos en la tabla y hago clic sobre alguno de ellos estos para comprobar que se rellenan los JTextField y el JComboBox.</p>
14.JComboBox		<p><b>Objetivo Comprobado:</b> Los desplegables funcionan y son autocompletable.</p> <p><b>Requisitos Comprobados:</b> Se comprueba q no se produce ningún error, tienen los elementos que deben y se puede seleccionar autocompletando.</p> <p><b>Pruebas Efectuadas:</b> Se comprueba cada uno de los desplegables, se observan los elementos que contienen y se escribe en ellos para comprobar que da opciones que concuerden con las letras introducidas.</p>
15.Botón Generar Excel		<p><b>Objetivo Comprobado:</b> Se genera un archivo Excel</p> <p><b>Requisitos Comprobados:</b> al presionar el botón correspondiente en la pestaña Almacén se genera un inventario en Excel con el material disponible y con la forma y nombre adecuados en la carpeta descargas.</p> <p><b>Pruebas Efectuadas:</b> Se clica en el botón correspondiente y observamos como se abre el archivo y lo localizamos en la carpeta de descargas.</p>
16.Botones Almacén	Pestaña	<p><b>Objetivo Comprobado:</b> Los botones funcionan y crean, actualizan, eliminan productos y limpian los campos de los JTextField.</p>

		<p><b>Requisitos Comprobados:</b> Se comprueba que no se produce ningún error al crear un producto, actualizarlo o eliminarlo y se actualiza la tabla.</p> <p><b>Pruebas Efectuadas:</b> Se comprueba cada uno de los botones y los mensajes de advertencia creados durante un proceso de creación, actualización o eliminado de productos.</p>
17.Tabla Cuidados	pestaña	<p><b>Objetivo Comprobado:</b> La tabla de la pestaña se completa con el listado de cuidados y al clicar sobre uno el JTextField se rellena automáticamente.</p> <p><b>Requisitos Comprobados:</b> al presionar el botón correspondiente en el menú se cargan los datos en la tabla y al seleccionar un elemento de la tabla esta rellena los campos.</p> <p><b>Pruebas Efectuadas:</b> Se selecciona el botón de Cuidados y se observa cómo están rellenos los datos en la tabla y hago clic sobre alguno de ellos estos para comprobar que se rellena el JTextField.</p>
18.Botones Cuidados	pestaña	<p><b>Objetivo Comprobado:</b> Los botones funcionan y crean, actualizan, eliminan cuidados y limpian los campos de los JTextField.</p> <p><b>Requisitos Comprobados:</b> Se comprueba que no se produce ningún error al crear un cuidado, actualizarlo o eliminarlo y se actualiza la tabla.</p> <p><b>Pruebas Efectuadas:</b> Se comprueba cada uno de los botones y los mensajes de advertencia creados durante un proceso de creación, actualización o eliminado de distribuidores.</p>
19.Agregar y eliminar Productos al Turno		<p><b>Objetivo Comprobado:</b> se agregan y eliminan productos en la tabla correspondiente.</p> <p><b>Requisitos Comprobados:</b> al presionar el botón correspondiente en la parte superior en turno se agregan productos, eliminan o lanza mensaje de advertencia según corresponda.</p> <p><b>Pruebas Efectuadas:</b> Se agregan diferentes productos con el botón correspondiente y se elimina alguno con su botón, así como se comprueba si no se selecciona ninguna cantidad o se seleccionan más de la que hay en almacén.</p>
20.Agregar y eliminar Cuidados al Turno		<p><b>Objetivo Comprobado:</b> se agregan y eliminan cuidados en la tabla correspondiente.</p> <p><b>Requisitos Comprobados:</b> al presionar el botón correspondiente en la parte superior de la tablase agrega o eliminan, según corresponda</p>

	<p><b>Pruebas Efectuadas:</b> Se agregan diferentes cuidados con el botón correspondiente y se elimina alguno con su botón.</p>
21.Guardar Turno	<p><b>Objetivo Comprobado:</b> se genera el turno y se crea un archivo pdf con los datos del turno.</p> <p><b>Requisitos Comprobados:</b> al presionar el botón correspondiente de guardar todo, se genera el archivo y se guarda el turno o no si faltan datos por rellenar.</p> <p><b>Pruebas Efectuadas:</b> Se agregan diferentes productos, cuidados y se presiona en el botón correspondiente vemos si se genera el turno. Para lo cual lanza un mensaje y abre el pdf de lo contrario solo lanza un mensaje de que falta información.</p>
22.Tabla pestaña Archivo	<p><b>Objetivo Comprobado:</b> La tabla de la pestaña se completa con el listado de turnos.</p> <p><b>Requisitos Comprobados:</b> al presionar el botón correspondiente en el menú se cargan los datos en la tabla</p> <p><b>Pruebas Efectuadas:</b> Se selecciona el botón de Archivo y se observa cómo están rellenos los datos en la tabla</p>
23.Botón PDF pestaña Archivos	<p><b>Objetivo Comprobado:</b> Se abre un pdf del turno seleccionado.</p> <p><b>Requisitos Comprobados:</b> al presionar el botón correspondiente, tras seleccionar un elemento de la tabla se abre el correspondiente archivo pdf guardado.</p> <p><b>Pruebas Efectuadas:</b> Se selecciona un elemento de la tabla y se pulsa el botón lo que provoca que se abra el documento que está relacionado con ese registro.</p>
24.Consultar y actualizar opciones.	<p><b>Objetivo Comprobado:</b> se cargan los datos y se actualizan al pulsar el botón</p> <p><b>Requisitos Comprobados:</b> al presionar el botón correspondiente en el menú se cargan los datos en los JTextField y al modificar los datos y pulsar el botón estos se actualizan</p> <p><b>Pruebas Efectuadas:</b> Se selecciona el botón de Opciones y se observa cómo están rellenos los datos en sus respectivos lugares, se cambian los datos y se comprueba que se han actualizado</p>
25.Tabla Empleados pestaña	<p><b>Objetivo Comprobado:</b> La tabla de la pestaña se completa con el listado de empleados y al clicar sobre uno los JTextField y el JComboBox se rellenan automáticamente.</p> <p><b>Requisitos Comprobados:</b> al presionar el botón correspondiente en el menú se cargan los datos en la tabla y</p>

		<p>al seleccionar un elemento de la tabla esta rellena los campos.</p> <p><b>Pruebas Efectuadas:</b> Se selecciona el botón de Empleados y se observa cómo están rellenos los datos en la tabla y hago clic sobre alguno de ellos estos para comprobar que se rellenan los JTextField y el JComboBox</p>
26.Botones Empleados	pestaña	<p><b>Objetivo Comprobado:</b> Los botones funcionan y crean y eliminan los empleados</p> <p><b>Requisitos Comprobados:</b> Se comprueba que no se produce ningún error al crear un empleado o eliminarlo y se actualiza la tabla.</p> <p><b>Pruebas Efectuadas:</b> Se comprueba cada uno de los botones y el mensaje de confirmación durante el proceso de eliminado de productos.</p>
27.Crear ejecutable		<p><b>Objetivo Comprobado:</b> Se crea el ejecutable a partir del .jar y funciona de forma esperada</p> <p><b>Requisitos Comprobados:</b> Se crea el .exe y este se ejecuta de la forma esperada, generando la base de datos y los datos primeros</p> <p><b>Pruebas Efectuadas:</b> Se comprueba todos os casos de usos posibles desde el ejecutable.</p>

## 8. Conclusiones (1-2 páginas)

A la conclusión de este proyecto puedo destacar que ha sido una gran experiencia ya q no contaba con experiencia previa en la creación de un proyecto desde cero, y este ha sido una gran manera de sacar a relucir y poner a prueba los conocimientos adquiridos durante el módulo de grado superior, como son Bases de Datos, Programación, Diseño de interfaces, acceso a datos y empresa, para desarrollar un modelo de negocio, asociado a este proyecto que aunque no se ve reflejado en esta memoria sí que se ha trabajado en él.

Cabe destacar en este proyecto que se ha contado con el apoyo de compañeros y amigos para la gestión de tiempos y para la colaboración en el proceso de diseño de la base de datos, dando el conocimiento necesario para que la aplicación fuese mínimamente funcional e interesante para el público al que va destinado.

Detrás del trabajo realizado tengo que destacar la adaptación del proyecto a mis circunstancias personales, la adaptación del modelo de desarrollo a un modelo adecuado en

horario y forma, para que este proyecto no me diera problemas y poder corregir y probar de forma unitaria cada uno de los apartados e ir incrementando el proyecto.

Ha habido dos puntos cruciales en el desarrollo del proyecto, más allá de la elección y la metodología los cuales han provocado retrasos pero se han conseguido solventar, a mitad del proyecto cuando iba a crear las funcionalidades para la pestaña Turno, me lanzaba errores al no haber diseñado bien las tablas de la base de datos, esto me llevo a modificar parte del proyecto y a la conclusión de la importancia de un buen diseño desde un principio, tras solventar este problema y tras una actualización del sistema operativo el programa XAMPP dejo de funcionar correctamente como servidor local de la base de datos lo que me llevo a la conclusión que para este tipo de proyecto y dada la cantidad de datos a archivar por la aplicación y su despliegue en local era más que suficiente la utilización de una base de datos SQLite, mucho más sencilla para ejecutar la aplicación.

En definitiva, el proyecto ha sido muy enriquecedor en lo personal y a nivel didáctico, me ha llevado a desarrollar lo aprendido y a buscar soluciones que hagan funcional y fácil cada paso para un usuario básico, que al fin y al cabo es a quien va dirigida esta aplicación.

## 9. Vías futuras

Se planifican dos vías para el futuro a partir del desarrollo efectuado, las cuales presento a continuación:

1. **Distribución, soporte y mejora:** en esta vía se planifica una distribución en fase beta de la aplicación hacia varios usuarios finales (Residencias), y a partir de los informes y sugerencias planteadas por estas añadir funcionalidades, mejorar las existentes y corrección de errores que los usuarios finales puedan encontrar, tras la implantación en alguna residencia se recurrirá a una reunión concertada para poder conocer la experiencia de uso y todo lo que puedan aportar al proyecto.
2. **Aplicación móvil y Tablet:** Al tener ya construida la aplicación con la base de datos en MySQL se mantiene la opción de implantar este sistema en una base de datos centralizada en un ordenador de la residencia que sirva como servidor y crear una aplicación móvil para que los empleados puedan incluir desde sus dispositivos toda la información referente a su turno así como consultar, los turnos anteriores, para ello se plantea la posibilidad de que solo se pueda conectar desde el móvil a la base de datos desde la red WIFI de la Residencia, provocando ello que los datos no estén expuestos en servidores externos y el tráfico de estos sea local, de igual manera se abre la

posibilidad a que la residencia cuente con una Tablet destinada para este fin y que los empleados no tengan que hacer uso de sus dispositivos.

Estas son las principales vías de desarrollo que me he planteado para continuar con el proyecto y mejorar la experiencia, tanto de uso como el aumento de características, para crear un ecosistema propio de sistema de gestión de residencias.

## 10. Bibliografía/Webgrafía

1. Crear archivo con fecha y hora:  
<https://es.stackoverflow.com/questions/166168/crear-archivo-con-fecha-y-hora-del-sistema>
2. Qué son los requisitos no funcionales: ejemplos, definición, guía completa:  
<https://visuresolutions.com/es/blog/non-functional-requirements/>
3. Data Access Object (DAO) Pattern  
<https://www.oscarblancarteblog.com/2018/12/10/data-access-object-dao-pattern/>
4. Plantillas diagrama de Gantt.  
<https://www.canva.com/p/templates/EAFPycRKQPI-gr-fico-diagrama-gantt-para-actividades-corporativo-rojo/>
5. Join MySQL: las distintas formas de unión entre tablas  
<https://www.premiumleads.com/blog/desarrollo/join-mysql-la-distintas-formas-de-union-entre-tablas/>
6. Problema con ImageIcon -> Ruta Absoluta  
<https://foro.chuidiang.org/java-j2se/problema-con-imageicon-gt-ruta-absoluta/>
7. Sistema de venta Java (swing) y Mysql – básico  
<https://informatica-programacion.com/sistema-de-venta-java-swing-y-mysql-basico/17/2022/>
8. Java - cambiar tamaño de JFrame  
<https://www.lawebdelprogramador.com/foros/Java/1126424-cambiar-tamano-de-JFrame.html>
9. Crear instalador java Netbeans desde cero.  
<https://www.tusolutionweb.org/crear-instalador-java-netbeans-con-base-de-datos-mysql/>
10. Error no encuentra librerías al generar .exe desde .jar java  
<https://es.stackoverflow.com/questions/261090/error-no-encuentra-librer%c3%adas-al-generar-exe-desde-jar-java>
11. Validaciones de JTextField Java Swing  
<https://informatica-programacion.com/validaciones-de-jtextfield-java-swing/18/2023/>
12. Reportes en Excel Java (Swing)  
<https://informatica-programacion.com/reportes-en-excel-java-swing/14/2023/>
13. ¿Crear una aplicación portable con java y SQLite?



- <https://www.programadornovato.com/01-crear-una-aplicacion-portable-con-java-y-sqlite-%F0%9F%93%A6/>
14. Añadir a Inno Setup un instalador de Java en caso tal de que la Pc no cuente con Java  
<https://es.stackoverflow.com/questions/209155/a%c3%b1adir-a-inno-setup-un-instalador-de-java-en-caso-tal-de-que-la-pc-no-cuente-con>
  15. CONVERTIR UN .JAR UN EJECUTABLE .EXE  
<https://hashblogeando.wordpress.com/2019/01/31/convertir-un-jar-un-ejecutable-exe/>
  16. Fix Java Virtual Machine Launcher Error, Could Not Create the Java Virtual Machine on Windows 11/10  
<https://youtu.be/08uKHfRbc7U>
  17. Crear una aplicación portable con JAVA y SQLite  
<https://www.youtube.com/watch?v=AuCdBBIbPdU>
  18. Java SQLite CRUD  
<https://www.youtube.com/watch?v=YkalyDj0wKo>
  19. Adjuntar Base de Datos a aplicación Java para convertirla en EXE  
[https://www.javamexico.org/foros/java\\_standard\\_edition/adjuntar\\_base\\_de\\_datos\\_aplicacion\\_java\\_para\\_convertirla\\_en\\_exe](https://www.javamexico.org/foros/java_standard_edition/adjuntar_base_de_datos_aplicacion_java_para_convertirla_en_exe)
  20. Conectar Java y Sqlite en Netbeans  
<https://evilnapsis.com/2022/09/28/conectar-java-y-sqlite-en-netbeans/>
  21. Crear Archivos Jar en Eclipse  
<https://www.youtube.com/watch?v=e-TWiuP5N4s>
  22. Diagrama UML Automático en NETBEANS con EasyUML  
<https://www.youtube.com/watch?v=MiysL0MC2ck>
  23. Migrar proyecto Netbeans a Eclipse  
<https://www.jc-mouse.net/java/migrar-proyecto-netbeans-a-eclipse>
  24. Java – Crear proyectos Maven en NetBeans y Eclipse  
<https://decodigo.com/java-crear-proyectos-maven-en-netbeans-y-eclipse>
  25. Así puedes diseñar y personalizar tus propios iconos para Windows  
<https://www.softzone.es/windows/como-se-hace/crear-personalizar-iconos/>
  26. Creador de iconos online  
<https://www.xiconeditor.com/>
  27. Kambanflow  
<https://kanbanflow.com/>
  28. Pngwing.  
<https://www.pngwing.com/>

## 11. Anexos

### 11.1. Manual de Usuario

GestResi Necesita unos requisitos técnicos bajos, necesita como requisitos version minida de JRE de 1.8 ya que es bajo el que está construido y JDK.

GestResi está ideada para un usuario con un nivel básico de informática, por lo cual es muy fácil su uso y su curva de aprendizaje y más teniendo en cuenta el rol o privilegios de cada empleado, para empezar, debe ejecutar el archivo y una vez iniciado le saldrá un mensaje para que el primer inicio sea con el correo “admin” y contraseña “admin”, lo cerramos y se abrirá el registro donde iniciamos con estos datos.

Se divide la aplicación es utilidades para empleado Administrador y Trabajador, por ello lo indicare por separado.

#### **Administrador:**

El menú aparece con todos los botones accesibles y el usuario puede acceder a todas las características por botones en el menú lateral:

1. Turno: se dispone de desplegables que hacen una búsqueda fácil de productos, cuidados o usuarios, al elegir un producto o cuidado debe introducirlo en la tabla correspondiente dando clic en el botón de la flecha. Si se ha introducido uno por error puede eliminarlo con el botón correspondiente. Tras ello puede seleccionar el guardar turno para guardar todo lo hecho.
2. Usuarios: puede introducir los datos de los diferentes usuarios, crear nuevos, consultarlos, modificarlos... según la botonería y tabla existente.
3. Distribuidores: puede introducir los datos de los diferentes distribuidores, crear nuevos, consultarlos, modificarlos... según la botonería y tabla existente.
4. Almacén: puede introducir los datos de los diferentes productos, crear nuevos, consultarlos, modificarlos... y crear un Excel con el material disponible según la botonería y tabla existente.
5. Cuidados: puede introducir los datos de los diferentes cuidados, crear nuevos, consultarlos, modificarlos... según la botonería y tabla existente.
6. Archivo: aquí puede observar en la tabla los turnos anteriores y consultar el Archivo pdf que se creó.
7. Opciones: Aquí puede modificar los datos de la residencia.

8. Empleados: aquí puede crear nuevos empleados asignándoles un rol o eliminarlos con los botones, también puede consultar los datos de los empleados existentes en la tabla.

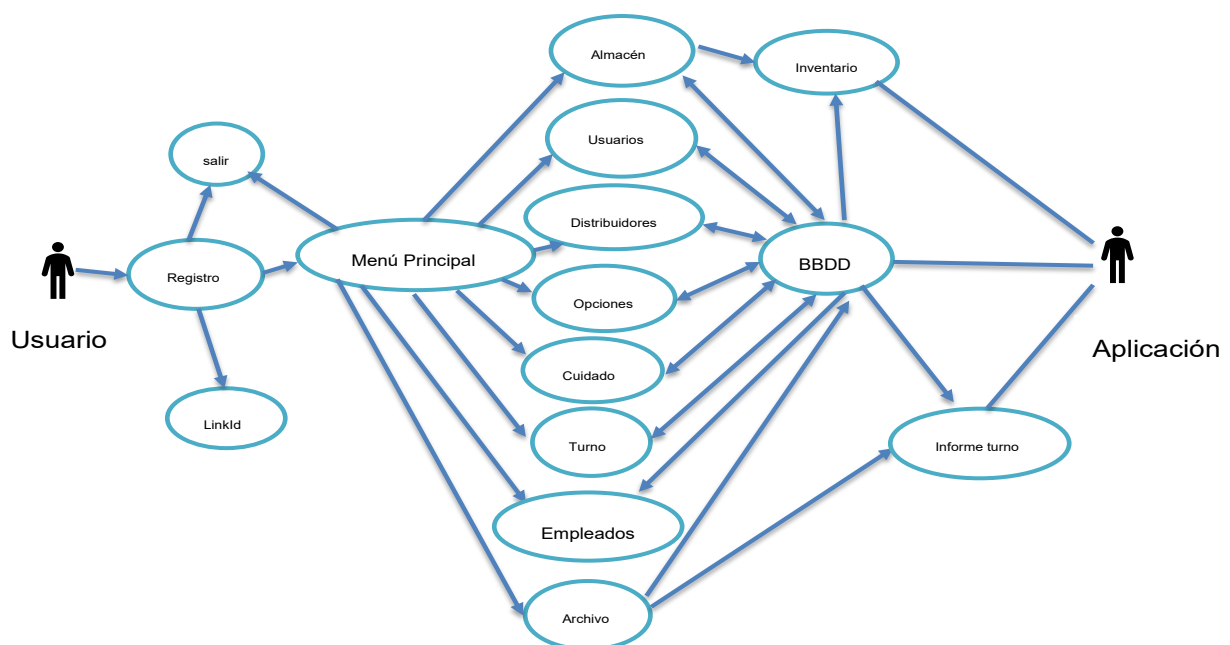
### Trabajador:

En caso del registro como trabajador tiene limitado el uso de la aplicación a solo dos botones:

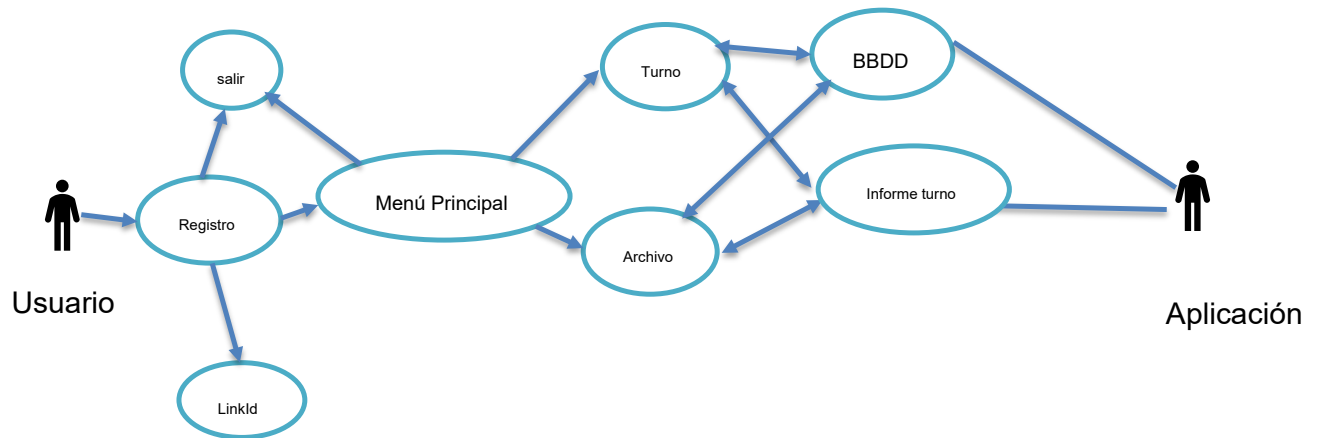
1. Turno: se dispone de desplegables que hacen una búsqueda fácil de productos, cuidados o usuarios, al elegir un producto o cuidado debe introducirlo en la tabla correspondiente dando clic en el botón de la flecha. Si se ha introducido uno por error puede eliminarlo con el botón correspondiente. Tras ello puede seleccionar el guardar turno para guardar todo lo hecho.
2. Archivo: aquí puede observar en la tabla los turnos anteriores y consultar el Archivo pdf que se creó.

## 11.2. Imágenes:

- Casos de uso Administrador:



- Casos de uso Trabajador:



- Diagramas de clases:





!!!!Gracias por leerme!!!!

