

DOCUMENTACIÓN DEL PROYECTO MANAGE MY LIFE

GESTOR DE TAREAS Y FINANZAS PERSONALES



MANAGE MY LIFE

Manuel Serrano Scholz
2ºDAM
IES Fernando Aguilar Quignon

Índice de contenido

INTRODUCCIÓN.....	4
Descripción del Proyecto.....	4
Aplicación móvil.....	4
Aplicación escritorio.....	5
Informes.....	6
Base de Datos.....	12
Servidor Multihilo.....	15
FINALIDAD.....	15
Finalidad.....	15
Para qué sirve.....	16
REQUISITOS.....	17
Requisitos Funcionales.....	17
Casos de Uso.....	25
DESCRIPCIÓN GRÁFICA.....	26
Aplicación Móvil.....	26
Aplicación de Escritorio.....	37
Prototipos de Diseño.....	49
Prototipo de Android.....	49
Prototipo de Swing.....	50
Medios Requeridos.....	51
Requerimientos Hardware.....	51
Requerimientos Software.....	51
Recursos Humanos.....	52
Requerimientos mínimos.....	52
Recogida de Información.....	53
Preparación de Entorno de Ejecución e Instalación.....	55
Uso de Librerías.....	58
Planificación Temporal.....	61

Índice de Ilustraciones

Ilustración 1: Flujo de funcionamiento de generación de Informes.....	6
Ilustración 2: Ejemplo de Informe de Tareas por Categoría.....	7
Ilustración 3: Ejemplo de Informe de información de las Tareas Pendientes.....	8
Ilustración 4: Ejemplo de Informe con número de subtareas por tarea.....	8
Ilustración 5: Ejemplo de Informe con cantidad de productos por tipo.....	9
Ilustración 6: Ejemplo de Informe con la información de los distintos gastos.....	10
Ilustración 7: Ejemplo de Informe con los usuarios y su número de tareas terminadas.....	11
Ilustración 8: Ejemplo de Informe con los usuarios y su número de gastos.....	12
Ilustración 9: Diagrama Entidad-Relación de la Base de Datos.....	13
Ilustración 10: Diagrama Modelo Relacional de la Base de Datos.....	14
Ilustración 11: Diagrama de Caso de Uso representando los requisitos funcionales.....	25
Ilustración 12: Diseño de la pantalla de Inicio de Sesión y Registro.....	26
Ilustración 13: Diseño de configuración de conexión y navegación de información.....	27
Ilustración 14: Diseño de la pantalla de perfil y modificar perfil.....	28
Ilustración 15: Diseño de la pantalla de tareas e información de las tareas.....	29
Ilustración 16: Diseño de la pantalla de Subtareas y Agregar Subtareas.....	30
Ilustración 17: Diseño de la pantalla de Creación de Tareas.....	31
Ilustración 18: Diseño de la pantalla de Tareas Terminadas.....	32
Ilustración 19: Diseño de la pantalla de Calendario.....	33
Ilustración 20: Diseño de la pantalla de Gastos y Crear Gastos.....	34
Ilustración 21: Diseño de las pantallas de Productos y Crear Productos.....	35
Ilustración 22: Swing - Diseño de la pantalla de Inicio de Sesión.....	37
Ilustración 23: Swing - Diseño de la pantalla de Registro.....	38
Ilustración 24: Swing - Diseño de la pantalla de Perfil.....	39
Ilustración 25: Swing - Diseño de la pantalla de Tareas.....	40
Ilustración 26: Swing – Diseño de la pantalla de insertar tareas.....	41
Ilustración 27: Swing – Diseño de la pantalla de información de tareas.....	42
Ilustración 28: Swing - Diseño de la pantalla de tareas terminadas.....	43
Ilustración 29: Swing – Diseño de la pantalla del Calendario.....	44
Ilustración 30: Swing - Diseño de la pantalla de Gastos.....	45
Ilustración 31: Swing – Diseño de la pantalla de Productos.....	46
Ilustración 32: Swing – Diseño de la pantalla de insertar productos.....	47
Ilustración 33: Swing – Diseño de la pantalla de Informes.....	48
Ilustración 34: Diseño Prototipo de Android.....	49
Ilustración 35: Diseño Prototipo de Android 2.....	49
Ilustración 36: Diseño Prototipo de Swing.....	50
Ilustración 37: Diseño Prototipo de Swing 2.....	50
Ilustración 38: Diseño Prototipo de Swing 3.....	50
Ilustración 39: Prototipo de planificación temporal.....	61
Ilustración 40: Planificación cumplida.....	61

1. INTRODUCCIÓN

1.1. Descripción del Proyecto

El proyecto **Manage My Life** consiste en el desarrollo e implementación de un producto que permita al usuario interactuar con distintos componentes que le permiten gestionar sus tareas, gestionar sus gastos y obtener datos sobre su rendimiento laboral y económico.

Se desarrollará como una herramienta destinada a la interacción del usuario para gestionar sus tareas diarias, la aplicación responde a la necesidad de muchos usuarios de poder gestionar sus tareas de forma intuitiva, fácil y rápida, permitiendo crear pequeños objetivos (en adelante, llamadas subtareas) en cada una de ellas, a la vez que da la posibilidad de manejar sus gastos permitiendo almacenar todos sus gastos en una sencilla tabla o lista, y en cada uno de estos gastos, un conjunto de productos con sus respectivos datos y precios.

Este proyecto va a estar diferenciado en 4 partes que se explicarán durante todo el proceso de documentación, estas partes constan de una aplicación móvil, una aplicación de escritorio, una aplicación Servidor Multihilo como intermediario entre las aplicaciones móvil y escritorio encargada de las consultas de datos y una base de datos MySQL como gestor de persistencia de datos de cada usuario.

1.2. Aplicación móvil

El proyecto será programado para dispositivos Android, la aplicación permitirá al usuario iniciar sesión en su cuenta o registrarse en donde el usuario podrá visualizar un formulario con los siguientes campos: Usuario, Nombre, Contraseña y Balance.

El usuario podrá acceder a su perfil donde estará toda su información y desde donde podrá modificar y consultar sus datos, estos datos serán las tareas pendientes, las tareas terminadas y su balance actual.

La aplicación permite al usuario programar distintas tareas en distintas **categorías** (trabajo, estudios, ocio..) permitiendo al usuario crear una **agenda programada**, el usuario podrá indicar un **nombre** a la tarea, una **descripción**, indicar que **tipo de tarea** es, indicar si la tarea es **prioritaria** o no, cuando se debe cumplir y **programar recordatorios**.

El usuario también podrá visualizar un **calendario** donde podrá ver las distintas tareas catalogadas por fechas, permitiendo visualizar la cantidad de tareas que tiene cada día mostrando los días que contengan tareas con un color destacado, al pinchar en uno de esos días se mostrará al usuario un **alerta temporal de 3 segundos** en la parte de notificaciones del terminal, indicando la cantidad de tareas que tiene para ese día, el usuario puede interactuar con esta alerta deslizando para descartarla de la pantalla manualmente.

Cada tarea tendrá ciertos campos, el nombre de la tarea, tipo de tarea, descripción de la tarea, conocer su prioridad, su estado, fecha de inicio, fecha de realización y permitirá al usuario crear dentro de la tarea **subtareas**, estas subtareas serán pequeñas tareas que habrá que realizar para cumplir con toda la tarea, pequeños objetivos que permitirán al usuario crear tareas mas complejas, el usuario podrá interactuar con estas subtareas de varias formas, agregándolas indicando un nombre a través de un botón flotante e indicando si se han terminado para finalizarlas de la lista deslizando a la izquierda.

Al terminar una tarea, todas las subtareas asociadas a ella también terminarán, también se le permitirá eliminar cada una de las subtareas deslizando la tarea hacia la derecha

En la gestión de tareas también habrá una pantalla distinta que permitirá al usuario ver una lista con las **tareas terminadas**, como si de una pantalla de logros se tratase.

La aplicación permitirá al usuario en una lista interactiva poder **programar y clasificar todos sus gastos**, se permitirá crear grupos de gastos y productos, creando así listas de gastos con sus propios productos en cada uno de ellos, de esta forma hacer mas sencilla y ordenada la gestión de los gastos por parte del usuario, con la finalidad de mostrar al usuario con distintos informes y gráficos información sobre los distintos tipos de gastos y productos, como por ejemplo en las distintas compras o pagos de facturas de servicios de luz, agua o Internet.

La aplicación también permite a través del navegador lateral entrar en las distintas pantallas de la aplicación.

Toda la aplicación será diseñada respetando las normas de Material Design.

1.3. Aplicación escritorio

La aplicación de escritorio será diseñada en JAVA con la biblioteca gráfica SWING, esta aplicación permitirá al usuario iniciar sesión/registrarse en pantallas distintas, estas trabajarán exactamente igual que en la parte Android explicada anteriormente.

Permitirá visualizar los datos del usuario a través de un panel de perfil donde se le mostrará las tareas pendientes, las tareas terminadas, el balance actual (respecto a los gastos registrados) y la opción de modificar su perfil a través de una pequeña pantalla donde podrá registrar nuevos datos e incluso insertar una imagen de perfil.

Al igual que en la aplicación Android, la interacción con la aplicación se realizará a través de un panel lateral (en este caso fijo) donde se podrá mover entre las tareas, tareas terminadas, gastos, informes y perfil.

En la gestión de tareas se mostrará una tabla diseñada para mostrar el nombre de cada tarea registrada, su estado y prioridad y se le permitirá insertar tareas a través de un botón rellenando un formulario. Al interactuar con la tabla podrá visualizar (dependiendo de la tarea seleccionada) su información al igual que las subtareas (pequeños objetivos de cada tarea) si es que tiene alguna, con la cual podrá interactuar también, añadiendo, eliminando y actualizando a gusto del usuario

Se le permite modificar y eliminar la tarea en la pantalla de información de la tarea seleccionada. Se le proporciona al usuario componentes de selección de fechas y selección de tipos de tarea intuitivas y fáciles para una mejor interacción con el cliente.

Podrá visualizar una tabla de tareas terminadas y una pantalla con una tabla con los distintos gastos, al interactuar con cada uno de ellos podrá visualizar otra tabla con los productos de ese gasto, permitiendo añadir y eliminar productos mostrando en su caso el nombre y el precio de cada uno de ellos.

Toda esta información será recolectada para la generación de informes, gráficos y estadísticas que se mostrarán a través de la ultima pantalla de la aplicación, Informes.

En informes se le permitirá al usuario visualizar los distintos informes con información personal y pública de otros usuarios para informar al usuario sobre su recorrido en la aplicación (tareas terminadas, tipo de tareas que se ha escrito en la aplicación, cantidad de usuarios que utilizan la aplicación) entre otros datos en informes diferenciados.

Se permitirá al usuario acceder a la aplicación de escritorio y a la aplicación móvil al mismo tiempo, permitiendo crear cierta sincronización en el uso (si insertamos una tarea en la aplicación móvil podremos verlo al actualizar la pantalla de la aplicación de escritorio, sin necesidad de cerrar el programa).

1.4. Informes

La aplicación de escritorio permitirá imprimir en pantalla (utilizando la aplicación por defecto que tenga el usuario para leer PDF) distintos informes que podrá seleccionar desde el panel de Informes (ultima opción del menú lateral) en las que podrá contemplar distintos botones, cada uno de estos botones permitirá visualizar distintos informes.

La funcionalidad de la obtención de los informes a través del servidor funciona siguiendo este esquema

Este esquema representa el flujo de datos tanto el envío como el recibo entre servidor/cliente, representando los 2 tipos de datos que envía el cliente al servidor (tipo de informe a imprimir e identificador del usuario activo), toda la funcionalidad del servidor, tanto el recibo, como la generación de informes a través de la librería JasperReport, como la conversión posterior, conversión a array de bytes y codificación en Base64 para un envío mas seguro a través de sockets.

FLUJO DE FUNCIONAMIENTO DE GENERACIÓN DE INFORMES

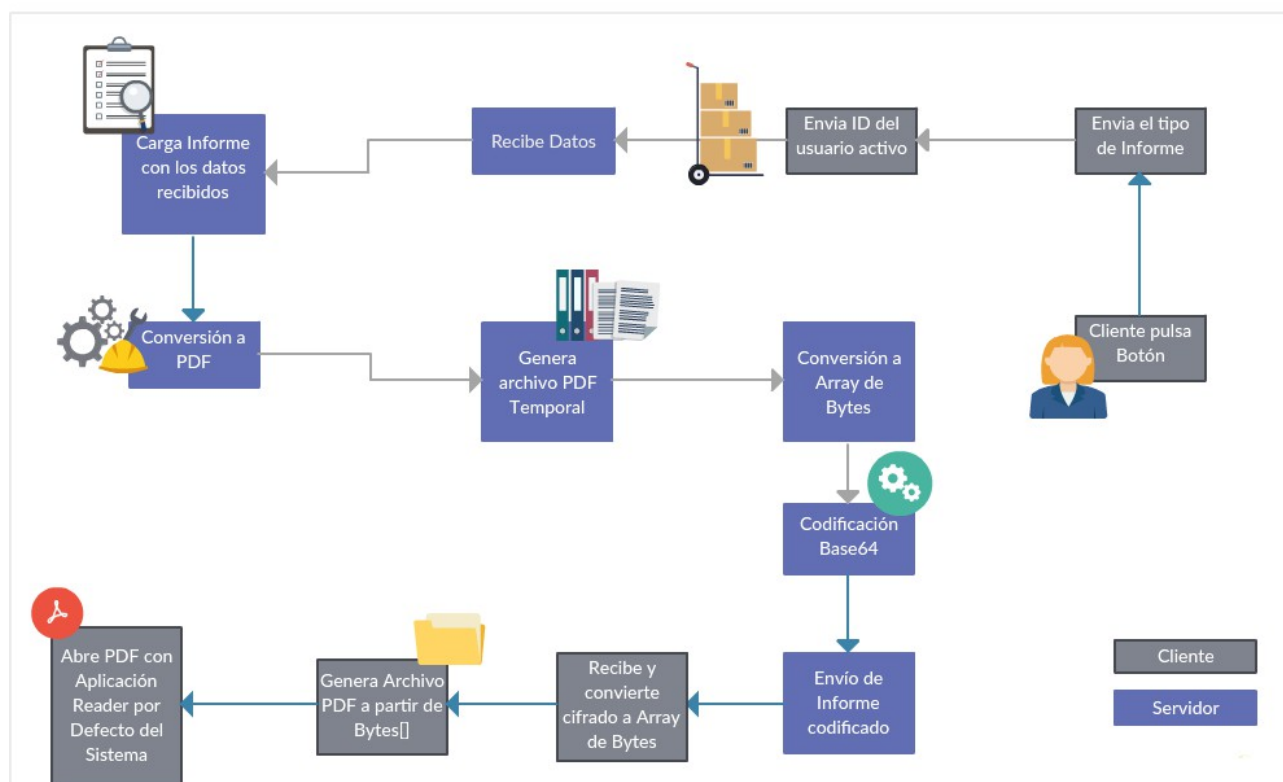


Ilustración 1: Flujo de funcionamiento de generación de Informes

El usuario recibe esta codificación, el cual sigue el mismo procedimiento a la inversa, recoge la codificación, convierte a array de bytes, genera el PDF y lo abre con la aplicación de lectura de PDF que tiene el sistema por defecto, todo a través de un botón y con total transparencia al usuario.

Distintos tipos de informes que podrá generar el usuario:

- **Cantidad de Tareas por Categoría**

Informe/Gráfico con el numero de tareas por cada categoría.

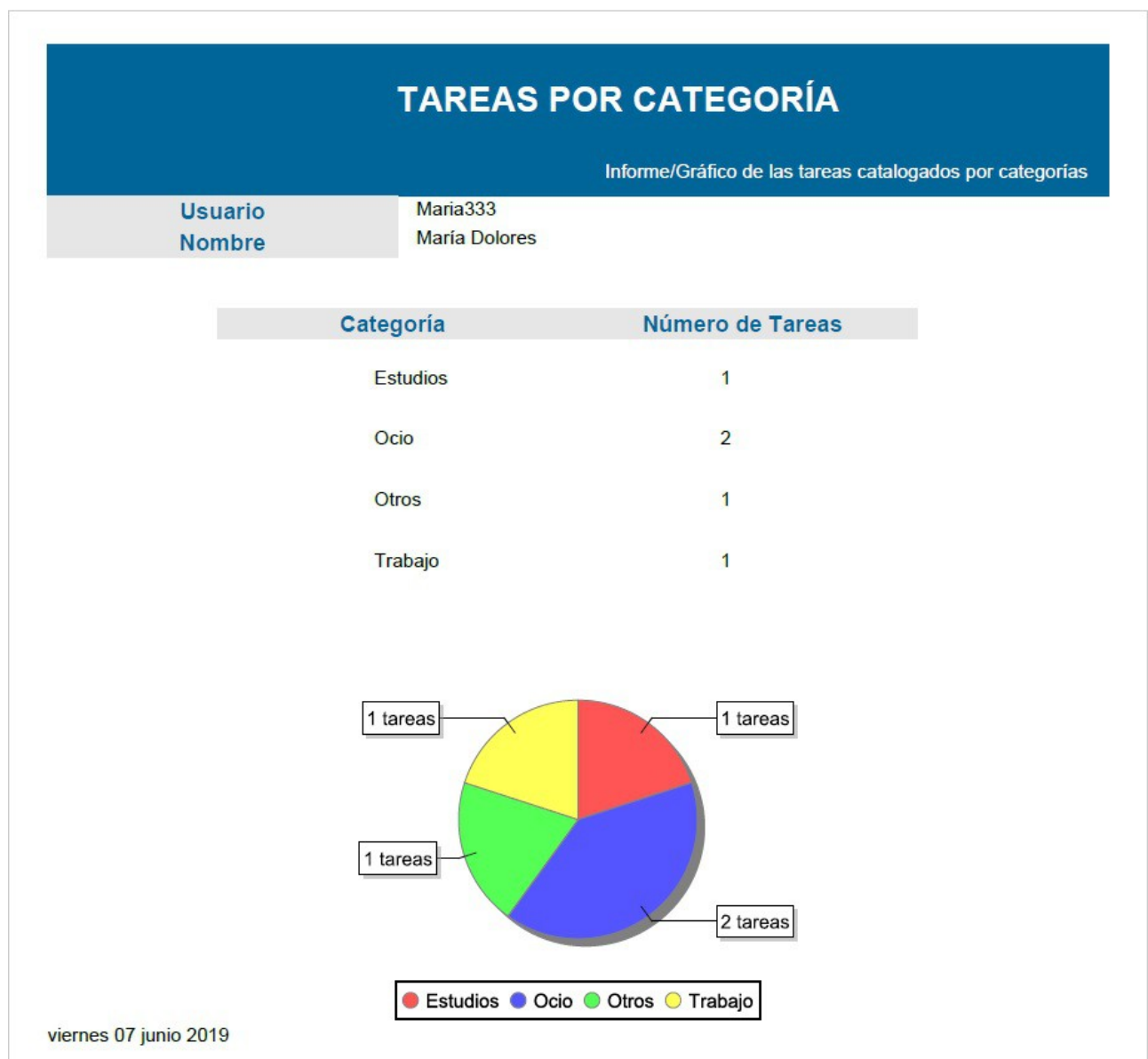


Ilustración 2: Ejemplo de Informe de Tareas por Categoría

- **Informe de todas las tareas pendientes**

El usuario podrá visualizar en un informe de texto todas las tareas que quedan pendientes con toda su información y si son prioritarias o no

TAREAS PENDIENTES			
Informe/Gráfico de Tareas Pendientes			
Usuario	Maria333		
Nombre	María Dolores		
Nombre de la Tarea	Categoría	Terminar	Prioritaria
Estudiar para el examen de	Estudios	17/06/19	Prioritario
Comprar billetes de avión	Trabajo	29/06/19	No Prioritario
Limpiar Piso	Otros	27/06/19	No Prioritario
Entregar Proyecto beBIR	Ocio	14/06/19	No Prioritario

viernes 07 junio 2019

Ilustración 3: Ejemplo de Informe de información de las Tareas Pendientes

- **Informe de nombre de tareas y su número de subtareas**

El usuario podrá visualizar el nombre de la tarea y su cantidad de número de subtareas.

TAREAS Y SUBTAREAS	
Informe/Gráfico de tareas junto con número de subtareas	
Usuario	Maria333
Nombre	María Dolores
Nombre de la Tarea	Número de Subtareas
Comprar billetes de avión	1
Entregar Proyecto beBIR	3
Estudiar para el examen de	3
Limpiar Piso	2
Preparar tarta de cumpleaños	2

viernes 07 junio 2019

Ilustración 4: Ejemplo de Informe con número de subtareas por tarea

- **Informe de numero de productos por tipo de producto.**

El usuario podrá visualizar la cantidad de productos que tiene el usuario por el tipo de producto (Producto y Servicio) junto con un gráfico



Ilustración 5: Ejemplo de Informe con cantidad de productos por tipo

- **Informe de Gastos**

El usuario podrá visualizar toda la información de sus gastos en forma de informe.

GASTOS			
Informe de Gastos			
Usuario	Maria333		
Nombre	María Dolores		
Nombre del Gasto	Tipo	Cantidad de Productos	Precio del Gasto
Facturas de la Luz	Servicio	2	69.5
Lista de la Compra	Producto	2	4.0
Reparacion del Coche	Servicio	2	30.5
Tarta de Cumpleaños	Producto	2	11.5
viernes 07 junio 2019			

Ilustración 6: Ejemplo de Informe con la información de los distintos gastos

- **Informe de Usuarios y sus tareas terminadas**

El usuario podrá ver la información de otros usuarios para conocer su nombre y su cantidad de tareas terminadas como si de un *ranking* se tratase

TAREAS TERMINADAS POR LOS USUARIOS DEL SISTEMA

Informe/Gráfico de usuarios registrados junto sus tareas completadas

Usuarios	Número de Tareas Terminadas
Alejandro Ramírez	5
María Dolores	2
Ana María	1
Manuel Serrano	1
Gloria Fernandez Ram	1

viernes 07 junio 2019

Ilustración 7: Ejemplo de Informe con los usuarios y su número de tareas terminadas

- **Informe de Usuarios y sus gastos**

El usuario podrá ver la información de otros usuarios para conocer su nombre y su cantidad de gastos como si de un *ranking* se tratase

USUARIOS Y GASTOS	
Informe con los usuarios y su cantidad de gastos	
Usuarios	Número de Gastos
María Dolores	4
Mari Carmen	2
Ana María	1
Alejandro Ramírez	1
Gloria Fernandez Ram	1

viernes 07 junio 2019

Ilustración 8: Ejemplo de Informe con los usuarios y su número de gastos

Estos informes se van a generar en el servidor a través de la librería *Jasper Report* trabajada durante el curso en el modulo 'Desarrollo de Interfaces', la cual trabaja a través de consultas SQL a la base de datos con la que se esta trabajando.

1.5. Base de Datos

Se trabajará con una base de datos *MySQL* que será el encargado de almacenar todos los datos de la aplicación y de los clientes, es una de las partes mas importantes del proyecto al igual que el servidor (encargado de comunicar la base de datos con los clientes) ya que **Manage My Life** es una aplicación totalmente interactiva con el usuario.

Esta trabajará con 5 tablas relacionadas entre ellas y con distintas columnas los cuales estarán llenas de datos escritas por los usuarios.

Estos datos se utilizarán a través de informes que mostrarán gráficos y estadísticas al usuario para poder visualizar y comparar sus resultados y ver su recorrido en la aplicación.

Modelo Entidad Relación

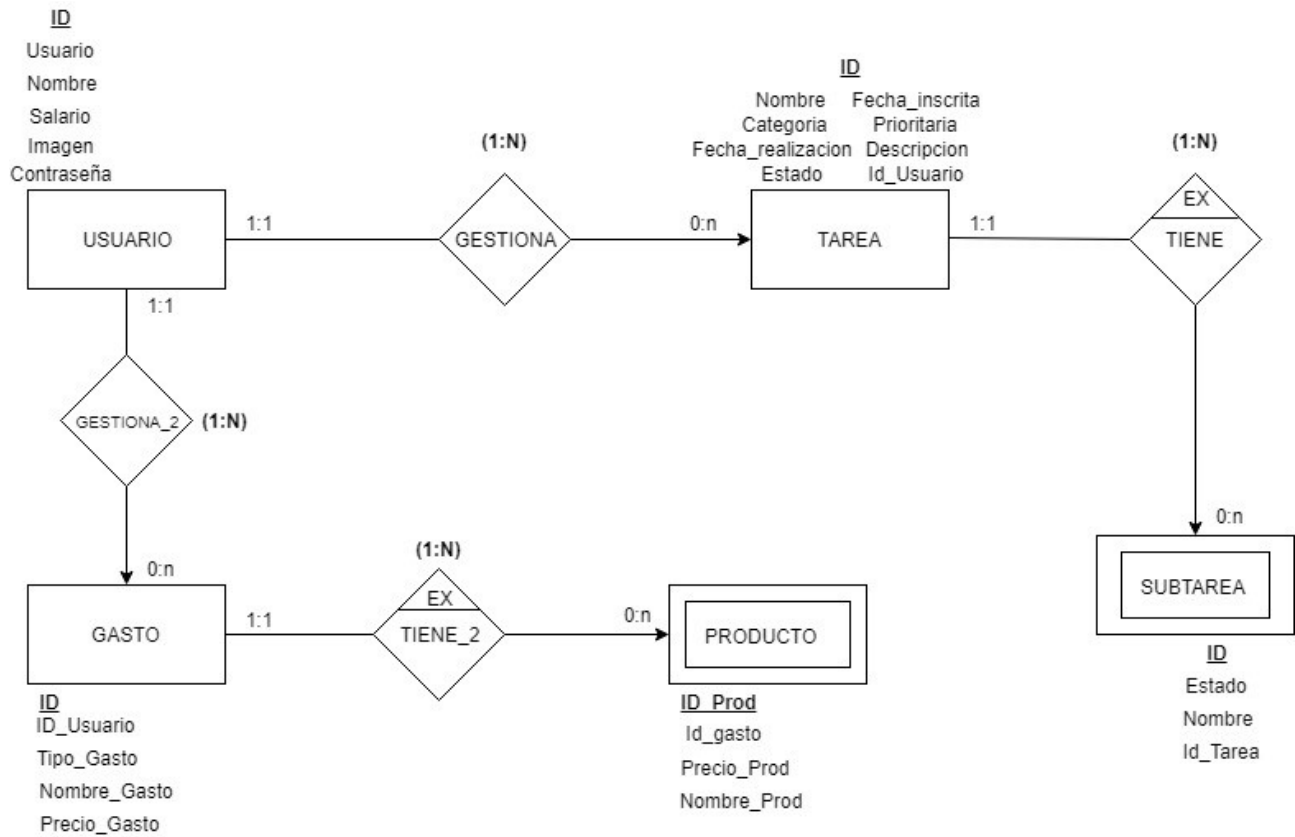


Ilustración 9: Diagrama Entidad-Relación de la Base de Datos

Modelo Relacional

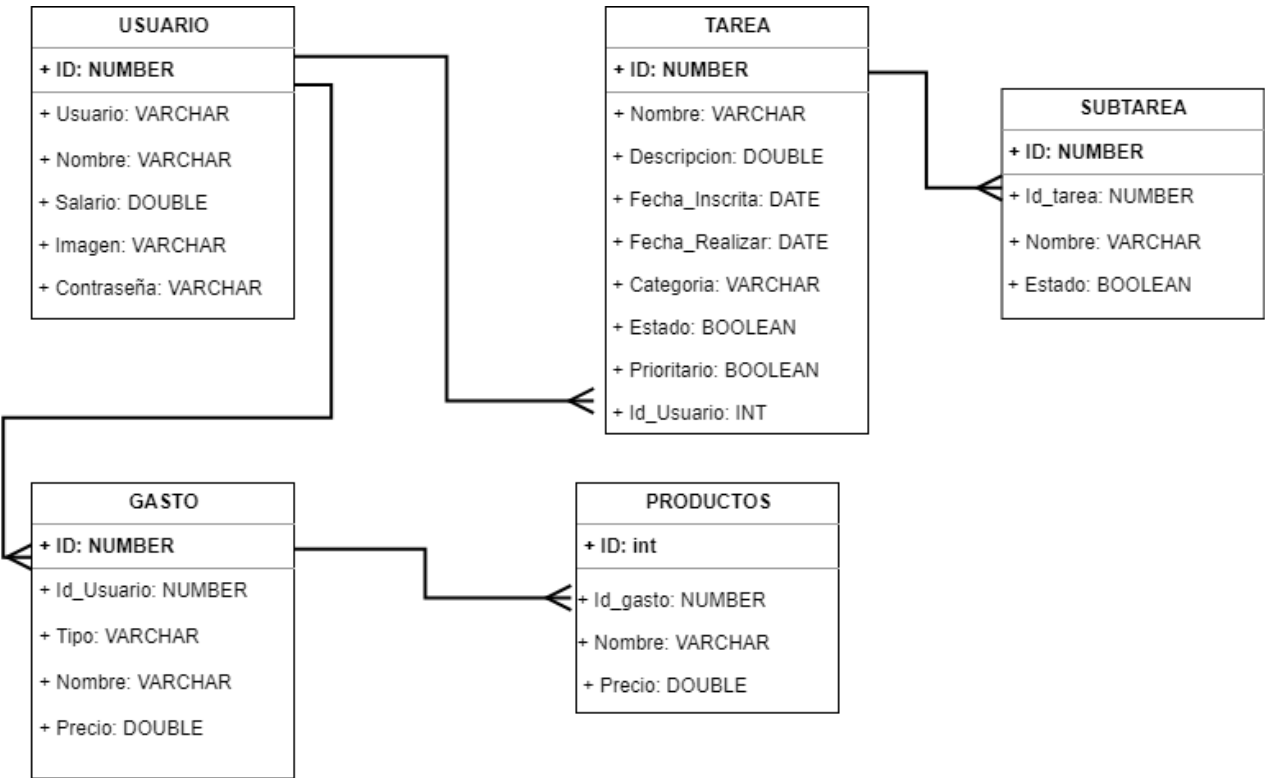


Ilustración 10: Diagrama Modelo Relacional de la Base de Datos

1.6. Servidor Multihilo

La aplicación móvil y la aplicación escritorio trabajaran con la base de datos pero estos trabajarán con un servidor multihilo como intermediario para permitir utilizar la aplicación en distintas "instancias" distintos usuarios.

Todas las funcionalidades y gestión de la información, al igual que la interacción de datos de parte del usuario en las aplicaciones se gestionarán a través de un servidor centralizado multihilo, que permitirá gestionar distintas conexiones en distintas instancias y trabajará como intermediario entre las aplicaciones clientes y la base de datos, este será el encargado de registrar todo tipo de manipulaciones en la base de datos (Inserciones, eliminaciones, actualizaciones, recogida de datos). También será el encargado de enviar datos a los distintos clientes.

Trabjará con un flujo de objetos, los objetos que se enviarán entre servidor y cliente son objetos "compartidos" los cuales deberán tener la misma estructura de atributos.

Cualquier gestión y funcionalidad del servidor se gestionará internamente con un objeto compartido previo el cual nos servirá para indicar que tipo de consulta/petición queremos realizar, esta petición será distinta en los distintos métodos de los clientes y a su vez, si el servidor lo requiere, obtendrá un objeto del tipo necesario (Objeto tarea, objeto usuario, objeto ...), el cual lo utilizará para realizar la consulta indicada (*INSERT*, *UPDATE*, *DELETE*, *SELECT*), el servidor puede (o no) devolver datos, el cual tendrá como referencia al objeto que recibieron, tanto para reenviar el objeto con valores actualizados como datos de confirmación.

La aplicación Android trabajará a través del servidor desde la clase *AsyncTask* permitiendo trabajar con las conexiones (envío y recibo de datos cliente-servidor) desde el *background* evitando bloqueos largos de la interfaz del usuario y permitiendo trabajar con otras funcionalidades mientras se envían/reciben y cargan los distintos datos.

2. FINALIDAD

2.1. Finalidad

Es un hecho que con las nuevas tecnologías y con la cantidad de distracciones que se forman a nuestro alrededor se nos olviden algunas de las tareas que teníamos pendientes a realizar en el día u olvidemos alguna fecha importante, Manage My Life permite al usuario administrar mucho mejor su tiempo, desde recordar fechas de celebraciones a recordar fechas de entregas de trabajo o reuniones importantes.

Es una aplicación ideal para jóvenes estudiantes, las tareas se acumulan y permite al estudiante priorizar algunas de ellas, llevar un calendario y poder gestionar mucho mejor su economía personal para ser mas eficiente tanto a la hora de su rendimiento académico, laboral y económico.

También es ideal para personas de todo tipo, desde jóvenes hasta empresarios o gente sin oficio que quieren recordar alguna fecha en especial o algún cobro pendiente que le deban y de esa forma, no olvidarse jamas de alguna deuda pendiente.

El usuario podrá acceder a su vez a la aplicación de escritorio en el cual podrá gestionar la agenda de tareas en forma de tablas interactivas y uso de una interfaz accesible.

Podrá visualizar tanto desde la aplicación móvil como la aplicación de escritorio distintos informes que informará al usuario de distintos parámetros de su cuenta personal de esta forma descubriendo como ha aumentado su productividad en los distintos sectores.

Para cumplir este objetivo se pretende desarrollar esta aplicación con el lenguaje de programación *JAVA*, haciendo uso del entorno de desarrollo *Android Studio* para la aplicación móvil y uso del entorno *Netbeans IDE* para el desarrollo de la aplicación escritorio, usaremos el conjunto de clases *Swing* para el diseño de la interfaz.

La aplicación pues debe permitir al usuario agregar, eliminar, modificar y consultar tareas. Agregar pequeños objetivos a las tareas si el usuario lo desea. Deberá permitir agregar y consultar sus gastos en el apartado financiero de la aplicación. Permitirá agregar datos a su cuenta personal, se le deberá mostrar de forma clara un calendario con los días diferenciados por colores mostrando las tareas que debe realizar para ese día y distintos apartados como un apartado donde podrá visualizar las tareas que ya ha terminado como "logro personal" entre otros añadidos como un panel con informes y gráficos que aportaran al usuario información adicional y una forma distinta de consultar su progreso en la aplicación

2.2. Para qué sirve

Manage My Life sirve para facilitar la vida a las personas, tratando de agilizar las distintas tareas que una persona tiene a lo largo de su vida, con recordatorios programados y un control total sobre las tareas, añadiendo y eliminando pequeños pasos en cada una de sus tareas para cumplir con cada una de ellas.

Sirve para mostrar al usuario de forma sencilla todas sus tareas pendientes en distintas categorías, indicando las prioridades, indicando su tipo (Estudios, ocio, trabajo..) En que productos y servicios gasta mas su dinero, programar una lista de compra, programar listas de deudas, cobros pendientes y todo lo que alcance a la imaginación del cliente.

Permite al usuario manejar de forma eficaz recordatorios, tareas, cobros, deudas y proyectos permitiendo al usuario tener mas tiempo libre y despreocupándose de recordar todos los trabajos pendientes.

3. REQUISITOS

3.1. Requisitos Funcionales

- **Función de Inicio de Sesión**

La aplicación permitirá al usuario gestionar los inicios de sesión a la aplicación, conociendo que usuario se ha conectando y mostrando en los distintos componentes la información pertinente de cada usuario.

Se gestionará este requisito en las distintas aplicaciones de formas similares, un formulario donde se insertará el usuario/contraseña y se comprobará a través del servidor si el usuario existe o no, informando al usuario con alertas.

Se comprobará el inicio de sesión cifrando en SHA-256 y transformando a hexadecimal y se comprobará en la base de datos si existe un cifrado igual.

- **Registro de cuentas**

El programa permite al usuario registrar una cuenta con el usuario, contraseña, nombre del usuario y un balance.

Las aplicaciones (Escritorio/Android) gestionará el registro de usuarios de formas similares, a través de una pantalla que se podrá acceder desde el inicio de sesión, rellenando un pequeño formulario donde se pedirán distintos datos (Usuario, nombre, contraseña, balance) al rellenar todos los datos se realizará el registro del usuario a través de un botón de registro, la gestión se realizará a través de una petición al servidor, y este una orden a la base de datos.

Se implementará un cifrado de contraseñas con un algoritmo SHA-256 y una conversión a hexadecimal, este hexadecimal se registrará en la base de datos cifrando así la contraseña.

- **Configurar conexiones**

Se le permitirá al usuario configurar conexiones tanto a la base de datos como a servidores.

Desde la aplicación servidor podrá configurar desde el directorio raíz un archivo config.txt donde podrá indicar parámetros de conexión en el cual podrá indicar los parámetros de conexión a una base de datos MySQL (nombre de la base de datos, usuario, contraseña).

Desde la aplicación de escritorio podemos conectarnos desde un archivo de configuración y distintos parámetros a un servidor u otro, en este archivo como primer parámetro se indicará la dirección IP del servidor (el cual será la dirección IPV4 del ordenador en el que esta instalado) al igual que el puerto

Desde la aplicación móvil podrá (y debe, al instalarlo por primera vez) registrar desde un botón de ajustes en la pantalla principal una dirección IP y un puerto por el que se va a conectar (en el caso del proyecto, se conectará la dirección IP donde este instalado el servidor y el puerto escogido por el usuario, el cual debe coincidir con el puerto que esta utilizando ese servidor), internamente registrará estos datos en una tabla de una base de datos implementada con la librería *Room*.

Se ha implementado funcionalidad en *Room* para que sustituya el primer registro de la tabla interna la dirección IP y el puerto cada vez que el usuario registra un nuevo servidor al que conectarse.

Tras la primera instalación (si no se desinstala) se registrará en una base de datos interna estos parámetros y el usuario no tendrá que indicar mas veces a que dirección y puerto conectarse.

- **Función de Modificar Perfil**

La aplicación debe permitir al usuario visualizar sus datos y a su vez, modificar sus datos, se gestionará esta funcionalidad de formas distintas en las distintas aplicaciones pero trabajará de forma centralizada a través del servidor

Escritorio: A través de la pantalla principal y un botón de ajustes permitirá al usuario acceder a una ventana modal donde podrá visualizar sus datos y poder actualizarlos (campos modificables: usuario y nombre).

Android: A través de un menú desplegable en la barra de la aplicación se podrá modificar el perfil, accediendo a una pantalla donde se mostrará toda la información y se permitirá modificarla a través de un botón.

- **Agregar imágenes**

Manage My Life permitirá al usuario agregar una imagen a su perfil el cual se mostrará en las distintas pantallas de las aplicaciones, el servidor hará una copia de la imagen y la guardará en una carpeta interna 'media' junto con una ruta en la base de datos con la relación del usuario.

Escritorio: A través de la pantalla del perfil se mostrará un botón de ajustes donde podrá visualizar distintos datos para modificar, aquí se le mostrará al usuario un mensaje que la imagen debe ser de formato 'jpg' y un botón que abrirá un explorador de búsqueda, este permitirá cargar una imagen y el usuario antes de aplicar los cambios, visualizará una previsualización de la imagen para saber como quedará en la aplicación.

Android: A través de la pantalla principal del perfil, se mostrará un botón Modificar Perfil, en este tendrá 2 opciones:

- Cámara: Se encenderá la cámara donde podrá hacer una foto y podrá aceptar para cargar la imagen capturada en el servidor.
- Galería: Se abrirá la aplicación de galería predeterminada del terminal donde el usuario podrá escoger una imagen

Estas imágenes se les aplicará un tamaño por defecto para encajar bien con los distintos componentes de la aplicación.

- **Seguimiento de tareas**

La aplicación mostrará una lista/tabla con las tareas simulando una agenda donde se podrá visualizar información como el nombre de la tarea, su estado y su prioridad el cual se mostrará de distintas formas dependiendo de la aplicación que se este utilizando.

Escritorio: En la aplicación de escritorio se mostrará un campo en tabla informando si el estado es "*Pendiente*" o "*Terminado*" y una columna para indicar si es prioritaria o no

Android: En la aplicación de Android lo mostrará con un checkbox que cambia automáticamente con la interacción del usuario al igual que un icono indicando su prioridad.

- **Agregar Tareas**

Permite al usuario desde la lista/tabla de tareas registrar distintas tareas, indicando para este distintos campos e informando al usuario con alertas si hay algún campo no válido o vacío.

Alguno de estos datos que se deberán de rellenar son: Nombre de la tarea, descripción, categoría, fecha a realizar e indicar si es prioritaria, la gestión interna del servidor determinará otros campos como el estado inicial de la tarea (0: pendiente) al igual que el identificador del usuario al que pertenece la tarea recién registrada y la fecha en la que se registró

Escritorio: Se gestionará el registro de tareas a través de un botón en la tabla de tareas donde se mostrará una ventana modal con los distintos campos a rellenar.

Android: Se gestionará el registro de tareas a través de un botón flotante en la lista de tareas donde se mostrará una ventana modal con los distintos campos a rellenar.

Se informará al usuario con una ventana de alerta si algún campo esta vacío o contiene algún dato erróneo.

- **Eliminar Tareas**

Permite al usuario desde la lista/tabla de tareas la opción de eliminar distintas tareas y con el, las subtareas asociadas a esta. La eliminación de tareas se gestionará de distintas formas en las distintas aplicaciones pero trabajará internamente de la misma forma a través del servidor multihilo.

Escritorio: Se eliminará las tareas accediendo a la deseada y a través de un botón se eliminará la tarea de la base de datos a través de una petición al servidor.

Android: Se eliminará la tarea desde la lista de tareas deslizando **hacia la derecha**, este avisará al usuario que la tarea se eliminará, al confirmar el usuario podrá ver como la tarea desaparece de la lista, eliminándose de la base de datos a través de una petición al servidor.

- **Modificar Tareas**

Permitirá al usuario el poder modificar los valores de las distintas tareas, obteniendo la tarea seleccionada, mostrando los datos actuales de la tarea y permitiendo cambiar los valores de sus campos, a través de una petición al servidor se realizará la modificación a la base de datos y actualizando la vista del usuario de distintas formas en las distintas aplicaciones

Escritorio: Se modificará la tarea accediendo a una de ellas y a través de un botón se notificará al usuario que la información que se esta visualizando ya es editable, habilitando un botón para aplicar cambios la cual modificará los valores de la base de datos

Android: Se modificará la tarea accediendo a la lista de las tareas, eligiendo la deseada se mostrará la distinta información de la tarea y a través de un botón se permitirá al usuario modificar los distintos valores de la tarea y actualizando la base de datos.

- **Cambiar estado de la tarea**

Se gestionará un cambio de estado de la tarea fácil y rápido, permitiendo al usuario cambiar de estado "Pendiente" a "Terminado" o viceversa de forma sencilla sin la necesidad implícita de modificar la tarea y actualizando las subtareas asociadas a la tarea con el estado de la tarea "padre".

Escritorio: Se gestionará la actualización del estado de la tarea accediendo a la información de la tarea e indicando a través de un check 'Terminado' si la tarea esta terminada o no, actualizando a su vez la tabla de subtareas asociada.

Android: Se gestionará la actualización del estado de la tarea deslizando hacia la izquierda la tarea deseada, actualizando el estado de sus sus subtareas asociadas.

- **Mostrar información de las tareas**

Se mostrará en una ventana diferenciada a la lista/tabla de tareas la información pertinente de la tarea seleccionada, esa información sería los mismos campos que introdujo el usuario a la hora de registrar la tarea a su vez que podrá visualizar la fecha en la que se registró la tarea y el estado de la tarea a través de un *check*, se mostrará de forma distinta en cada una de las aplicaciones información sobre su prioridad, nombre, descripción, categoría y su lista de subtareas.

- **Mostrar Subtareas**

Se mostrará las subtareas a través de la información de las tareas, gestionándose la muestra de este tipo de datos en distintas tablas o listas al igual que las tareas.

- **Agregar Subtareas**

El programa debe permitir al usuario agregar subtareas a la tarea seleccionada, esta gestión se mostrará de distintas formas en cada una de las aplicaciones pero se trabajará a través del servidor de la misma forma, obteniendo la tarea que el usuario esta gestionando y agregándole distintas subtareas (pequeñas tareas que pertenecen a esa tarea) también dando la posibilidad de no tener alguna subtask, siendo así una tarea de un solo objetivo.

Escritorio: Podrá agregar subtareas a través de una tabla dentro de la información de la tarea seleccionada y un botón que mostrará una ventana modal con el nombre de la subtask que se agregará al sistema, configurando el estado inicial internamente en Pendiente.

Android: Podrá agregarse subtareas en una lista distinta a la de las tareas pero que se accederá a ella a través de una tarea de la dicha lista de tareas, mostrando de forma clara las subtareas que pertenecen a la tarea seleccionada y un botón flotante para agregar el nombre de esa subtask.

- **Eliminar Subtareas**

Al igual que agregar subtareas, también permitirá eliminar subtareas, esta eliminación de subtareas se gestionará de la misma forma que las tareas, esta interacción con el usuario se trabajará igual que agregando subtareas, pero la forma en como se interacciona será distinta en cada aplicación.

Escritorio: Podrá eliminar subtareas a través de la tabla de subtareas y un botón para eliminar cada registro de subtareas en la tabla.

Android: Podrá eliminar subtareas deslizando a la derecha la subtask deseada.

- **Visualizar calendario de Tareas**

Escritorio: El programa mostrará un calendario en las distintas aplicaciones donde informará al usuario con un calendario, este calendario mostrará distintos días con distintos colores creando de esa forma un calendario interactivo donde el usuario al hacer clic en cada uno de los días se le mostrará con un alerta en pantalla mostrando el día y las tareas pendientes para ese día.

Android: El programa mostrará un calendario de tareas catalogado en distintas fechas y usando colores para mostrar los días que haya tareas y los días que no haya tareas, se mostrará la cantidad de tareas que hay por día a través de un alertar al interactuar con los distintos días

- **Visualizar Gastos**

Manage My Life permitirá visualizar de distintas formas los gastos que acumula el usuario permitiendo ver en distintas listas personalizadas distintos productos y con ello generar distintos informes y gráficos que informarán al usuario de sus gastos y que tipo de gastos genera.

Escritorio: Se mostrará a través de una tabla, informando la cantidad de gastos y el balance actual

Android: Se mostrará a través de una lista, informando la cantidad de gastos y el balance actual.

- **Añadir Fondos**

La aplicación permitirá al usuario agregar fondos al balance actual desde un botón en pantalla, apareciendo una ventana emergente para indicar la cantidad que queremos ingresar.

Escritorio: Se mostrará a través de un botón en la pantalla de finanzas, aparecerá una ventana modal donde el usuario podrá indicar la cantidad a ingresar actualizando el balance en pantalla.

Android: Se mostrará a través de un botón flotante en la pantalla de finanzas una ventana emergente donde el usuario podrá indicar la cantidad a ingresar, actualizando el balance en pantalla

- **Agregar Gastos**

La aplicación permitirá al usuario agregar gastos al sistema, estos 'gastos' son listas de productos los cuales contendrán distintos registros con distintos precios que registrará el usuario, estos gastos se sumarán restando al balance actual del usuario informándole de su balance actual para controlar de esta forma, los distintos gastos que tiene.

Escritorio: Desde la pantalla de finanzas, se podrá visualizar el balance y la cantidad de gastos junto con la foto de perfil del usuario, aquí podrá insertar distintos gastos indicando el nombre del gasto y el tipo de gasto, el tipo de gasto se utilizará mas adelante para la generación de distintos informes y gráficos

Android: Desde la pantalla de finanzas, se podrá visualizar una lista de registros 'gastos', habrá un botón flotante desde donde se visualizará una pequeña ventana pop donde podrá insertar el nombre del gasto y el tipo de gasto

- **Eliminar Gastos**

Las aplicaciones permitirán a su vez, eliminar los distintos gastos que se agregan al sistema.

Escritorio: Desde la pantalla de finanzas, se podrá acceder a cada uno de los gastos accediendo a estos desde la tabla, al acceder a la pantalla de los productos de estos gastos podrá visualizar el usuario un botón de eliminar gasto, permitiendo eliminar todos los productos del gasto y a su vez el gasto, esto sumará al balance actual todos los productos eliminados del gasto eliminado.

Android: Desde la pantalla de finanzas, se podrá acceder a los gastos, el usuario tendrá la opción de deslizar a la derecha para eliminar el gasto, el sistema avisará que el gasto se eliminará y con ello, todos los productos dentro, dándole la opción al usuario de permitir o cancelar la petición.

- **Agregar Productos**

Las aplicaciones permitirán agregar productos a estos distintos gastos que habrá creado previamente el usuario.

Estos productos tendrán un nombre y un precio registrado por el usuario, al agregar un producto actualizará el balance actual del usuario a través del servidor.

Escritorio: Se mostrará los productos a través de la tabla de gastos donde se abrirá un modal donde podrá visualizar los productos y a través de un botón accederá a una ventana donde podrá insertar el nombre y el precio del producto.

Android: Se mostrará los productos a través de una pantalla que se accederá al acceder a uno de los gastos, en esta pantalla verá otra lista de productos con el nombre y el precio, se podrá agregar a través de un botón flotante y una ventana donde podrá insertar nombre y precio del producto, la lista se actualizará al agregar un producto y se restará el precio del balance actual del usuario.

- **Eliminar productos**

A su vez *Manage My Life* permitirá al usuario eliminar productos tanto en la aplicación *Swing* como en la aplicación Android, al eliminar cada uno de los productos se sumará el precio de ese producto al balance, cada eliminación o agregación de un producto actualizará el registro del balance actual del usuario permitiendo visualizar el balance en cada momento si hay algún cambio en alguna de sus listas.

Escritorio: Se eliminará los productos escogiendo de la tabla de productos el producto que se desea eliminar y a través de un botón 'eliminar'.

Android: Se eliminará los productos deslizando de la lista de productos hacia la derecha cada uno de los registros.

- **Visualizar informes y gráficos**

El proyecto *Manage My Life* constará de una aplicación de escritorio que permitirá al usuario poder visualizar en distintos informes su evolución durante su transcurso en la aplicación al igual que poder visualizar información de sus tareas en forma de informe y gráficos

También podrá ver otros tipos de informes que pueden resultar útiles o interesantes al usuario como poder ver una lista de usuarios registrados en el sistema junto con sus logros (en este caso, sus tareas terminadas) como si de un *ranking* se tratase

- **Gestionar Notificaciones**

Manage My Life permitirá al usuario gestionar desde un menú de opciones en la aplicación Android, establecer una hora donde la aplicación notificará al usuario si desea registrar alguna tarea, realizando recordatorios para motivar al usuario a usar la aplicación y permitir recordatorios diarios para incitar al usuario a utilizar la aplicación.

3.2. Casos de Uso

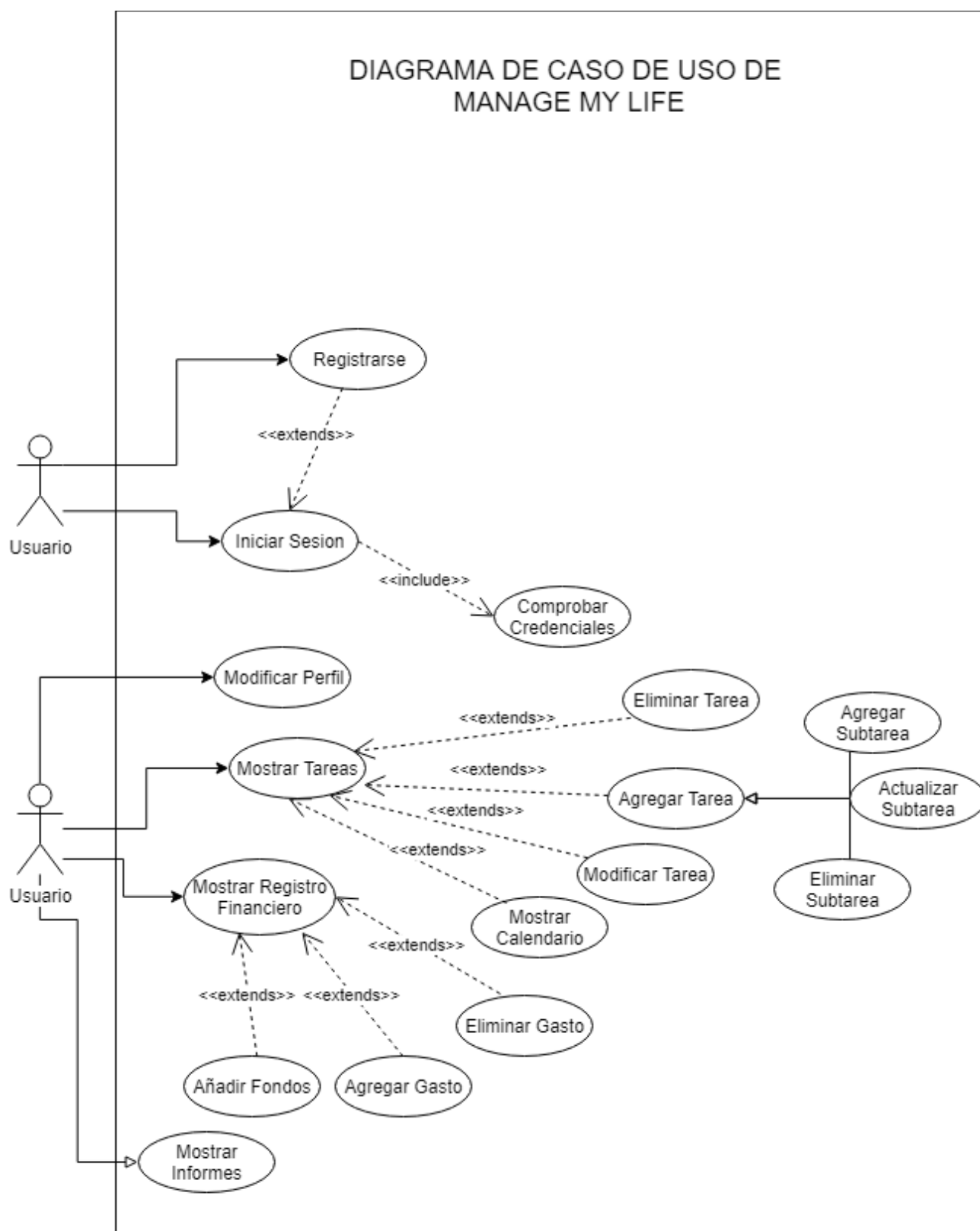


Ilustración 11: Diagrama de Caso de Uso representando los requisitos funcionales

4. DESCRIPCIÓN GRÁFICA

4.1. Aplicación Móvil

El diseño de la aplicación constará de distintas pantallas Pantalla de inicio de sesión, Pantalla de registro, Panel de navegación, Perfil, Calendario, Finanzas, Tareas y Tareas Terminadas.

Cada pantalla trabaja independiente aunque entre ellas se van comunicando desde un Activity principal donde se recoge un objeto usuarios con los datos del usuario conectado actualmente en el sistema.

Se utilizará un componente *Drawer Layout* (menú lateral) para la navegación de las ventanas y se ha utilizado una paleta de colores con temática en azul tratando de trabajar con distintas tonalidades azules utilizando la herramienta coolours.co, con tamaño de textos y márgenes utilizando referencia de *Material Design*.

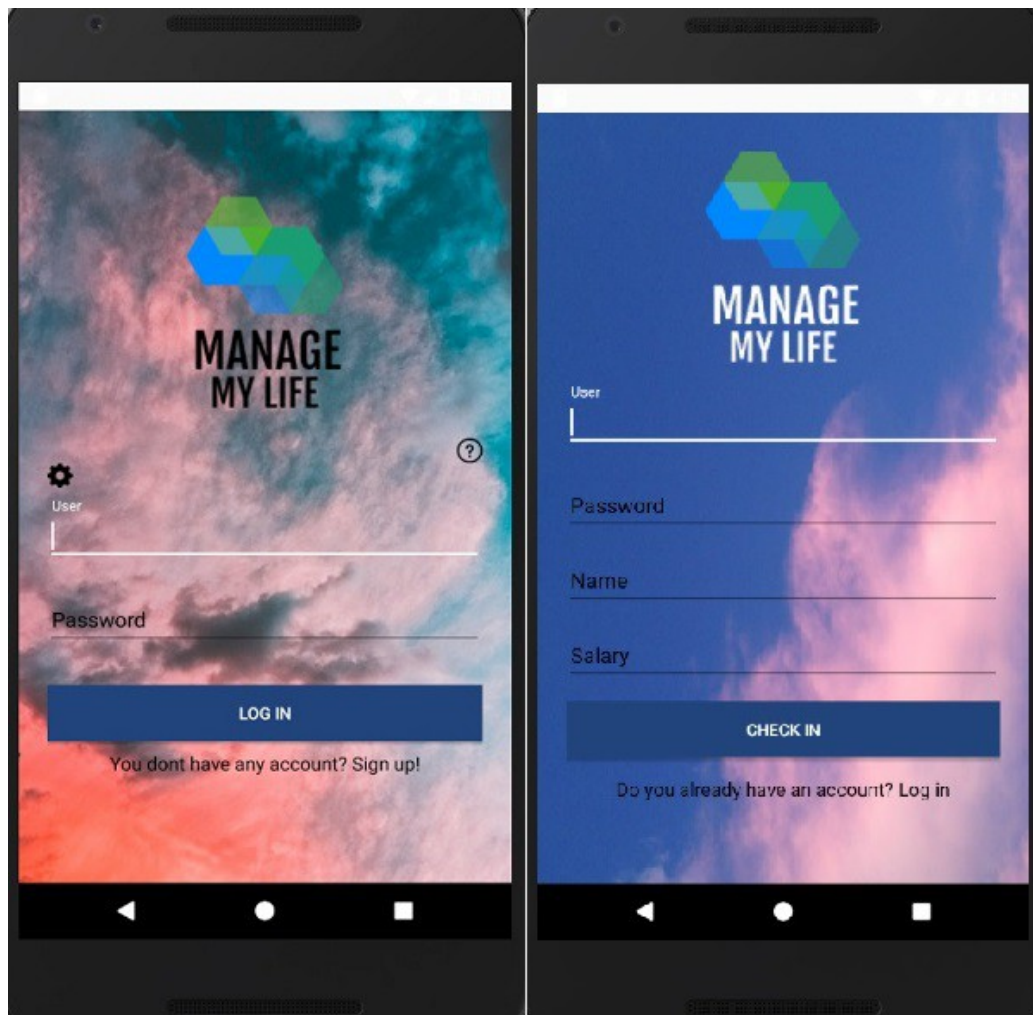


Ilustración 12: Diseño de la pantalla de Inicio de Sesión y Registro

La pantalla de inicio de sesión y la pantalla de registro son 2 tipos de formularios donde trabajaran de formas similares, en el registro cifraremos la contraseña con SHA-256, almacenando el cifrado en la base de datos, en la pantalla de Inicio de Sesión se cifrara la contraseña introducida y se comparará con la del usuario registrado en el sistema, comprobando si cada uno de los hash coinciden.

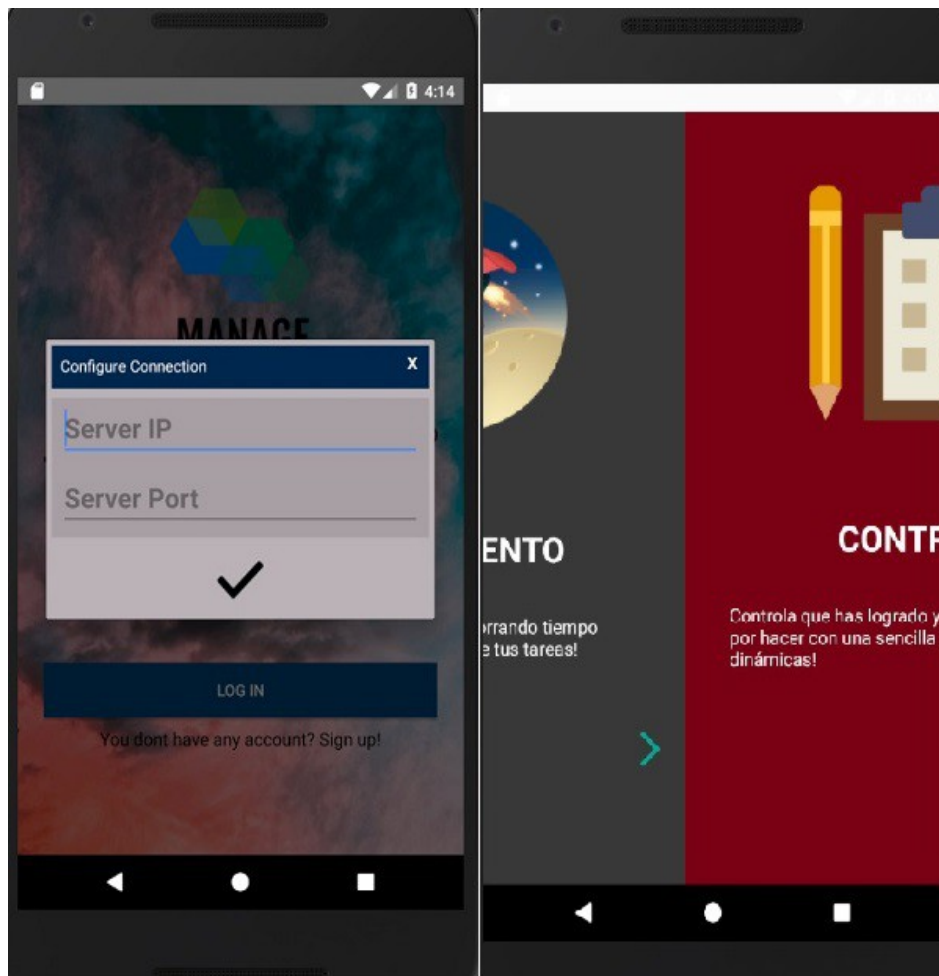


Ilustración 13: Diseño de configuración de conexión y navegación de información

En la pantalla de inicio de sesión podremos observar 2 imágenes distintas indicando ajustes y un botón de información.

Ajustes: Permite configurar una conexión con el servidor indicando la dirección IP del servidor y su Puerto al que se va a conectar para obtener la información

Información: Envía al usuario a una pantalla *slider* con 3 pantallas distintas informando al usuario que podrá realizar en la aplicación con distintos colores e imágenes, al final de todas las pantallas habrá un botón "Empezar" que enviará al usuario de nuevo a la pantalla de *login* para iniciar sesión en el sistema o registrarse en el.

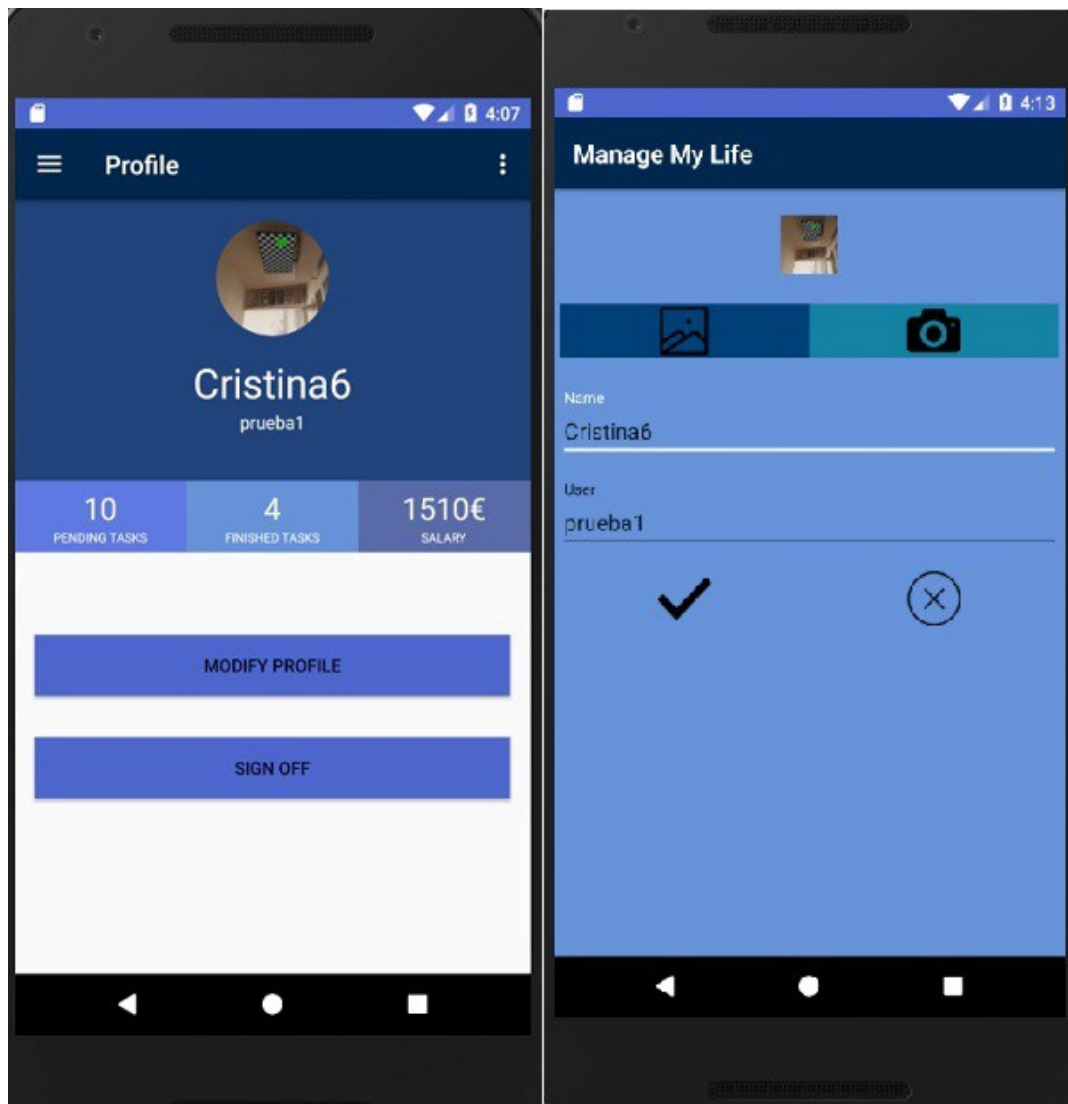


Ilustración 14: Diseño de la pantalla de perfil y modificar perfil

Al acceder a la aplicación podremos encontrar distinta información.

La primera pantalla es la pantalla principal la cual representa el panel del usuario con distinta información como la cantidad de tareas pendientes que tiene, la cantidad de tareas finalizadas y el salario actual.

En esta primera pantalla hay 2 botones distintos los cuales indican “Modificar Perfil” y “Cerrar Sesión”.

Modificar Perfil nos envía a una pantalla donde podemos editar el nombre o el usuario del perfil, en esta pantalla cabe destacar la posibilidad de agregar o eliminar imágenes desde 2 opciones:

Galería: Permite acceder a la galería del dispositivo y agregar una imagen.

Cámara: Permite acceder a la funcionalidad de la cámara del dispositivo y agregar una imagen.

Al guardar los cambios informará al usuario de que se cerrará la sesión automáticamente para guardar los datos y deberá iniciar sesión de nuevo.

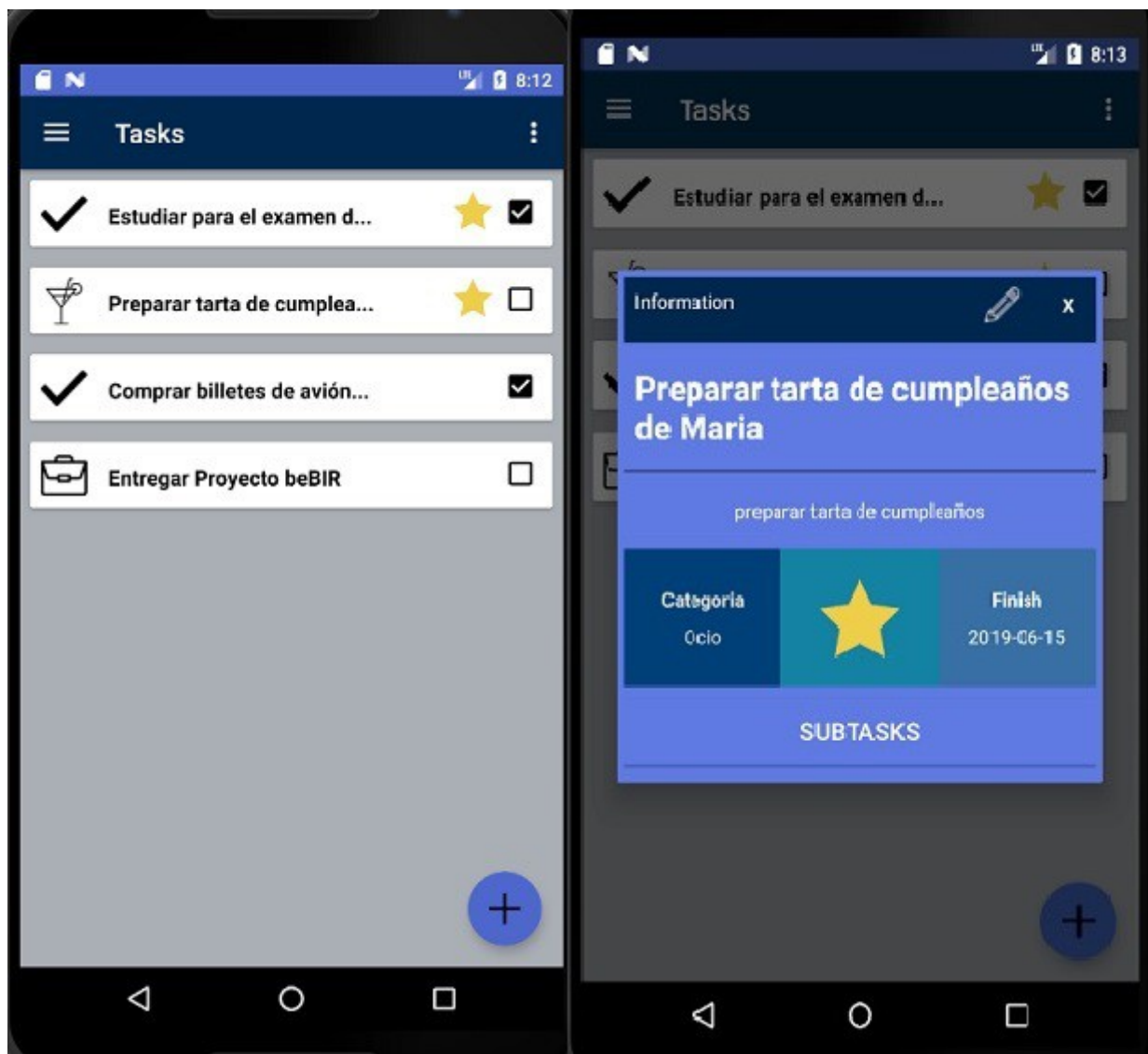


Ilustración 15: Diseño de la pantalla de tareas e información de las tareas

La pantalla de tareas podemos ver una lista *recycler* (RecyclerView) el cual tan solo cargará la información en memoria que el usuario visualiza en pantalla.

En esta lista podemos ver distinta información, el nombre de distintas tareas, un icono representando el tipo de tarea (Estudios, trabajo, ocio, otros), un check indicando el estado de la tarea y una estrella indicando si la tarea es prioritaria, el usuario podrá realizar distintos eventos en los distintos elementos y un check indicando si la tarea se ha terminado o no.

Eliminar tarea: El usuario podrá eliminar la tarea deslizando a la derecha una tarea, el sistema le notificará que la tarea se eliminará permanentemente, si el usuario acepta, este se eliminará.

Actualizar estado de la tarea: Deslizando a la izquierda cada registro el usuario podrá actualizar la tarea indicando que la tarea se ha terminado apareciendo un tick en el check representando que la tarea se ha finalizado con éxito, el usuario puede volver a deslizar para actualizarlo de nuevo y dejarlo en estado Pendiente.

El usuario podrá acceder al panel de información de cada tarea pinchando en cada una de ellas, en este aparecerá una ventana con los distintos parámetros de la tarea.

En esta ventana podrá ver si la tarea es prioritaria (la imagen de la estrella), la fecha de cuando se debería terminar la tarea, su descripción, categoría y nombre.

En el panel de información de la tarea hay un icono de edición en donde todos los campos se podrán editar permitiendo modificar la tarea.

También podrá ver un botón de Subtareas donde se podrá visualizar las distintas subtareas de la tarea (pequeños objetivos para terminar la tarea principal).

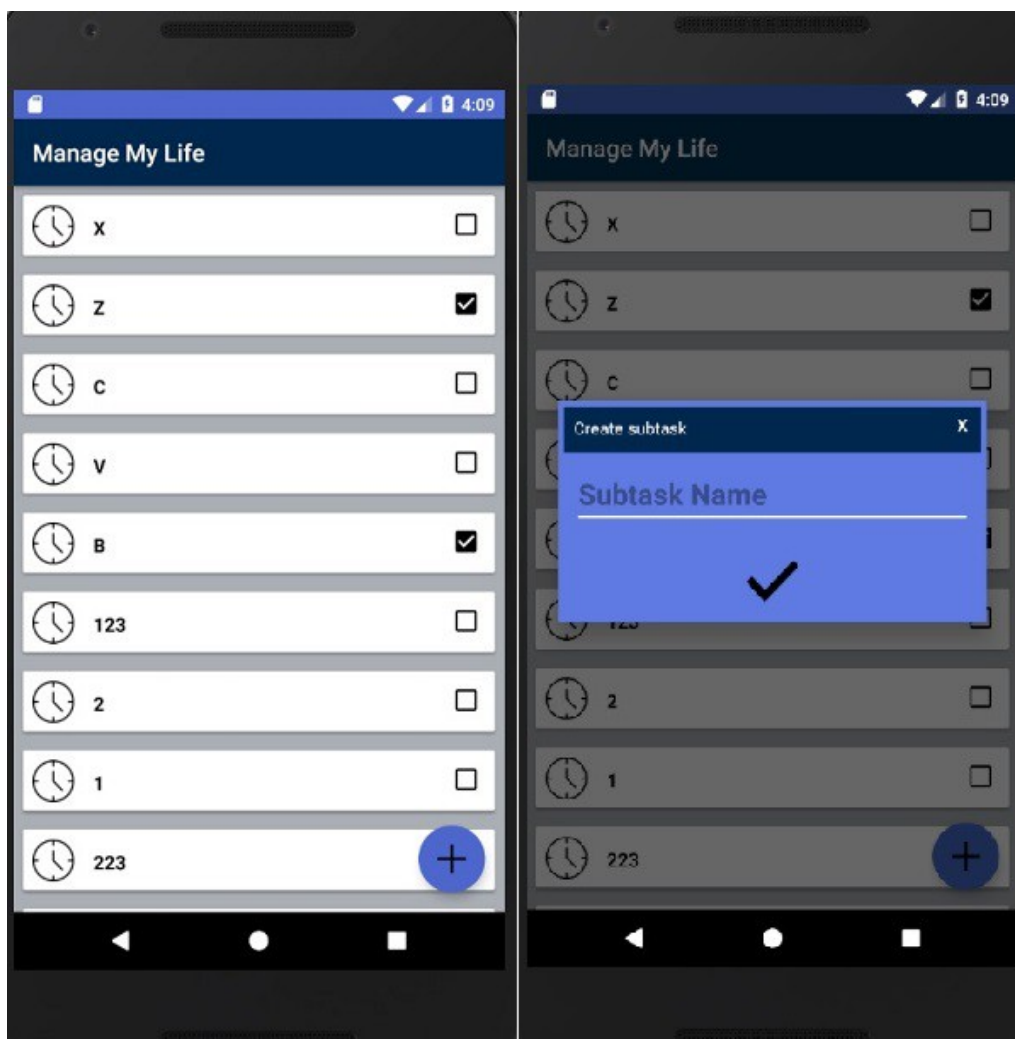


Ilustración 16: Diseño de la pantalla de Subtareas y Agregar Subtareas

Al acceder al panel de subtareas se podrá ver una lista de subtareas, similar a la anterior, la cual representa pequeños objetivos a lograra para completar con éxito la tarea. Se puede agregar subtareas a través de un *Dialog* en el cual insertaremos el nombre de la subtarea y se agregará automáticamente a la lista.

Estos registros tienen los mismos eventos.

Eliminar subtarea: Deslizamos hacia la derecha, se nos preguntará si queremos eliminar la subtarea.

Actualizar subtarea: Podemos actualizar la subtarea de Pendiente a Terminado o viceversa deslizando el registro hacia la izquierda.

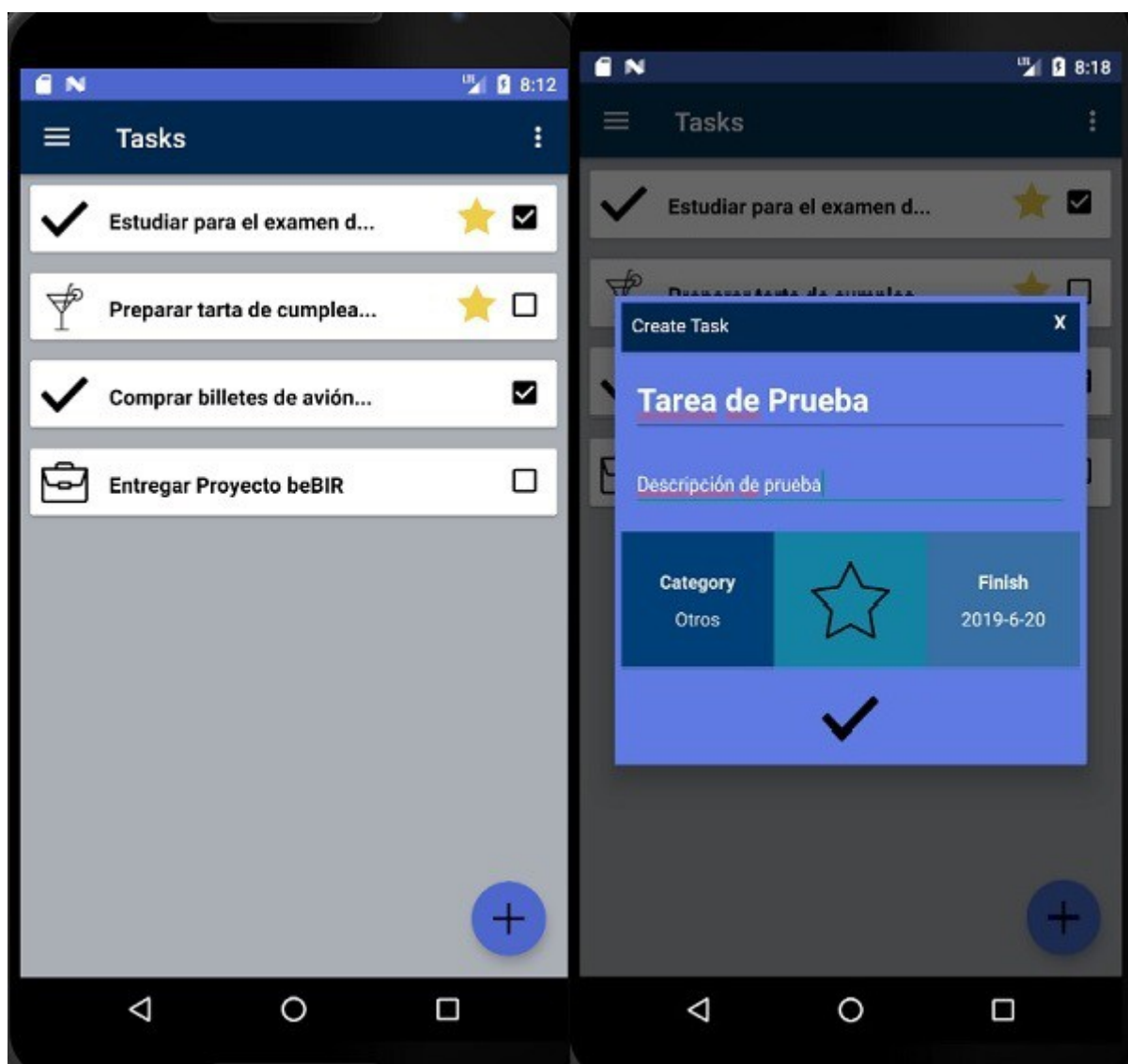


Ilustración 17: Diseño de la pantalla de Creación de Tareas

Volviendo a la lista de tareas, si pulsamos en el botón flotante se podrá agregar una tarea indicando distintos parámetros, estos parámetros son los siguientes:

- **Nombre de la tarea**
- **Descripción de la tarea**
- **Categoría de la tarea:** Aparecerá un *popup* con las distintas opciones
- **Prioridad:** Si es prioritaria, podemos pulsar en la estrella y se iluminará.
- **Fecha a realizar:** Componente calendario donde escogeremos cuando terminar la tarea.
- **Aceptar:** Confirmamos.

Al confirmar se analizará que todos los campos tiene algún valor, si algún campo importante no tiene escrito algún valor la aplicación notificará al usuario permitiendo agregar información en los campos restantes.



Ilustración 18: Diseño de la pantalla de Tareas Terminadas

En esta pantalla el usuario podrá visualizar en una lista las distintas tareas que se han terminado como si de un sistema de logros se tratase, permitiendo ver toda la cantidad de tareas terminadas que ha finalizado desde que comenzó a gestionar sus tareas con *Manage My Life*.

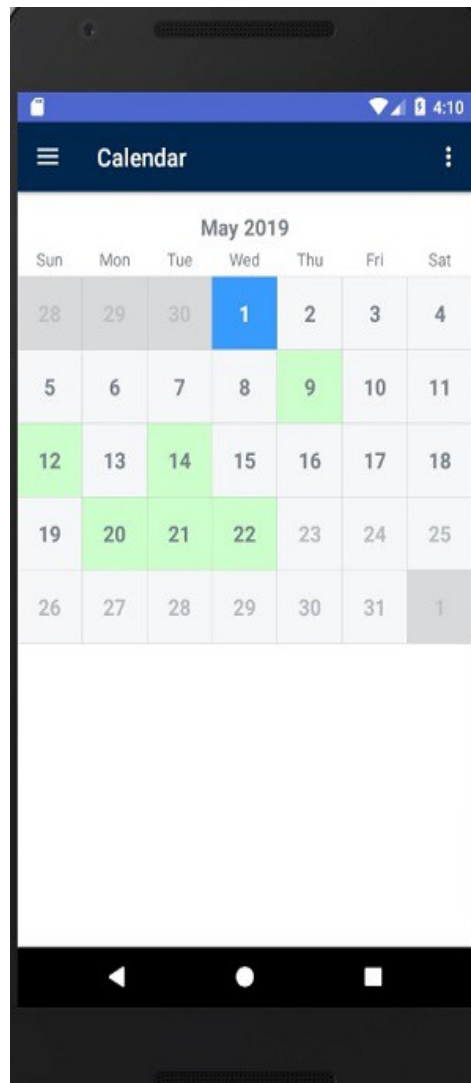


Ilustración 19: Diseño de la pantalla de Calendario

La pantalla del Calendario permite al usuario ver en distintos colores los días del mes.

Es una forma práctica para el usuario ver los distintos días del mes en los que tiene tareas registradas en estado Pendiente. Al pinchar en cada uno de estos días el usuario verá a través de una alerta temporal en la parte superior de la pantalla información sobre ese día y la cantidad de tareas que tiene que realizar para ese día en concreto.

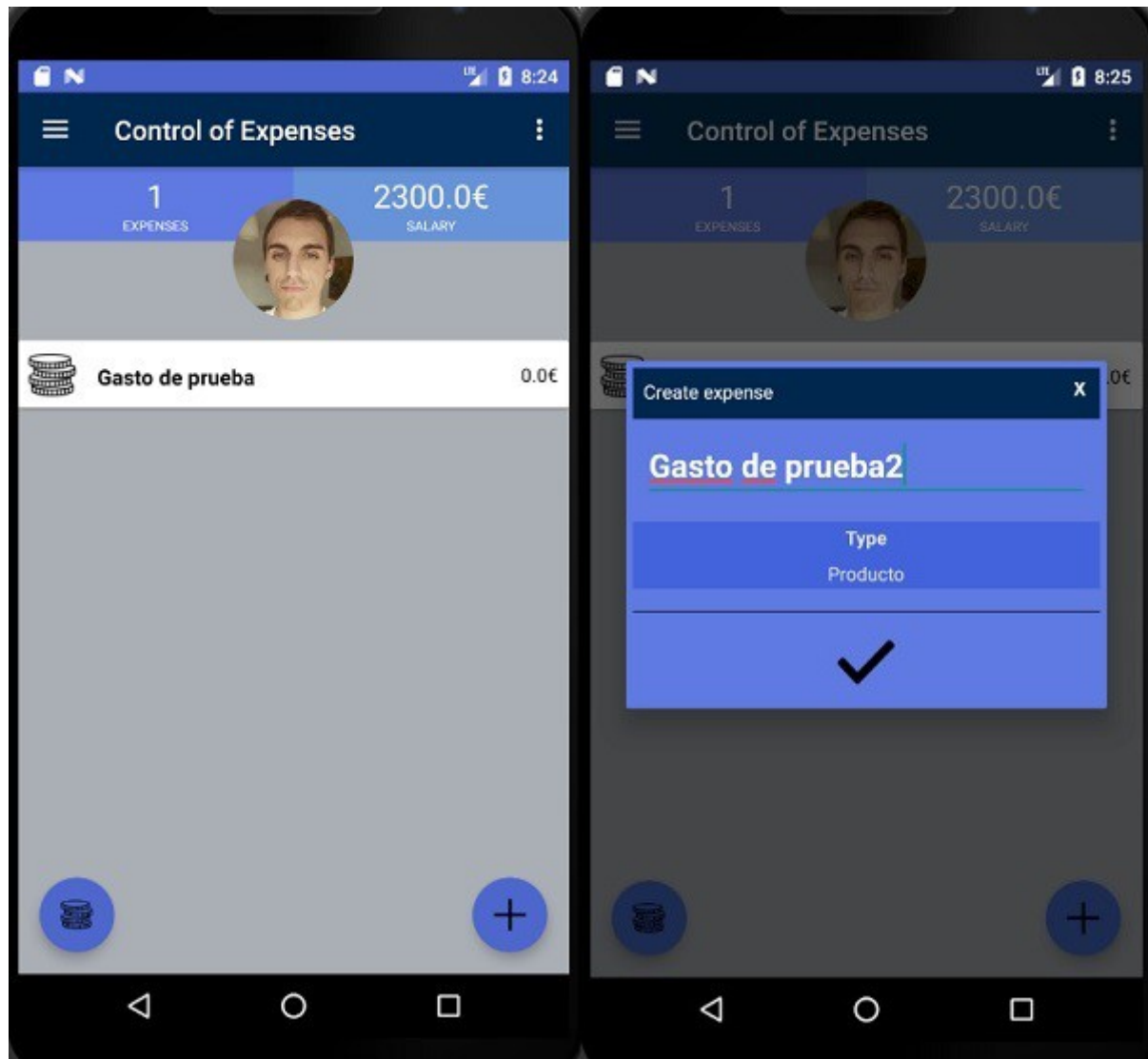


Ilustración 20: Diseño de la pantalla de Gastos y Crear Gastos

Pantalla de Gastos, en esta pantalla permite gestionar distintos gastos, en el podemos ver una lista *RecyclerView* similar a la de las tareas, en el que podremos ver el nombre del gasto, al elegir uno de los gastos, accederemos a una pantalla de productos, productos relacionados con ese gasto y donde podremos añadir distintos gastos incrementando el precio o disminuyendo el precio de ese gasto.

Para agregar una tarea tenemos un botón flotante el cual mostrará un modal donde se podrá insertar el nombre del gasto y al pinchar en la casilla de Tipo nos permitirá elegir entre 2 tipos de gastos (Servicio o Producto) el cual se utilizará internamente para la generación de Informes y Gráficas en la aplicación de escritorio.

Para eliminar un gasto se deslizará a la derecha y se actualizará en pantalla el numero de gastos.

Si el usuario decide eliminar un gasto con varios productos dentro, se sumará todos los productos de ese gasto al salario devolviendo el importe al balance del usuario.

Si el usuario decide añadir fondos al balance actual podrá hacerlo desde el botón flotante de la izquierda

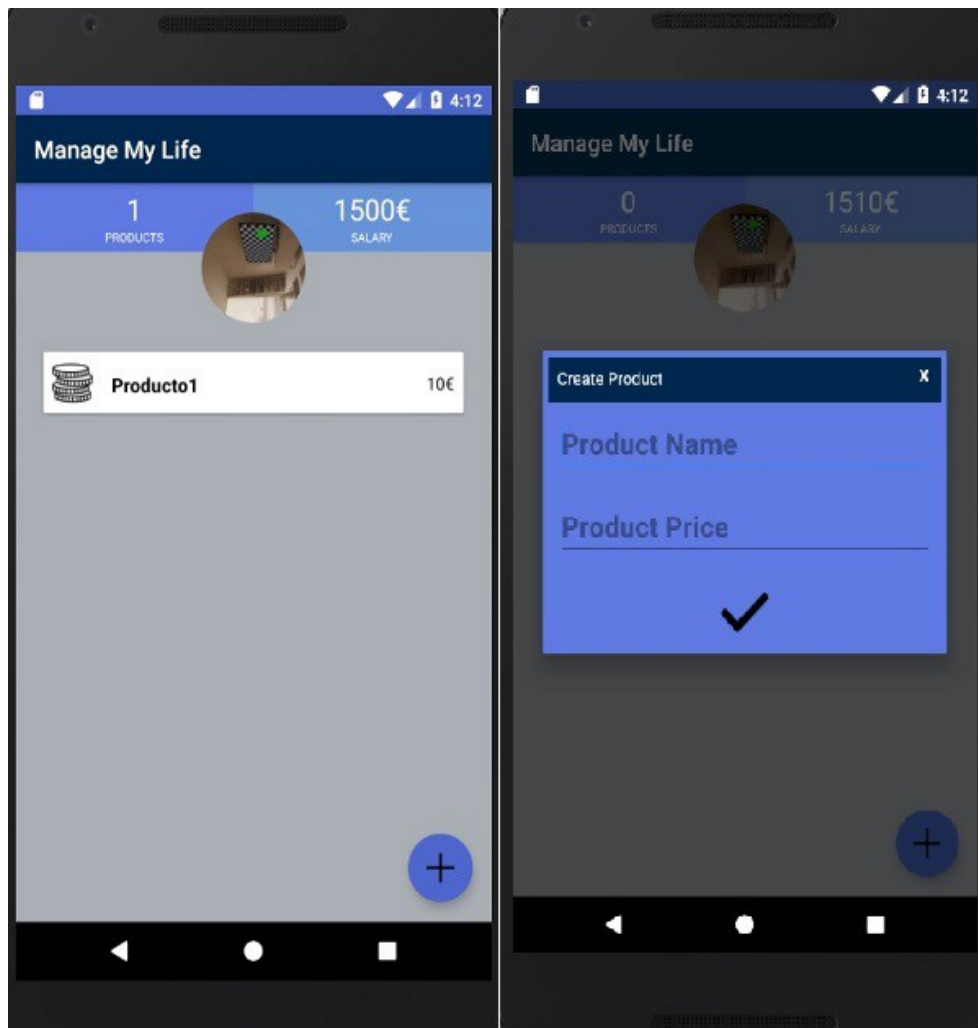


Ilustración 21: Diseño de las pantallas de Productos y Crear Productos

La pantalla de productos permite visualizar al igual que la pantalla de gastos distintos productos los cuales cada registro mostrará el nombre del producto y el precio de cada producto, sumando y actualizando en pantalla la información de cantidad de productos y el balance actual (balance actual – suma del precio de los productos).

Para ingresar un producto tendremos un botón flotante donde podrá indicar el nombre de producto y el precio.

Podrá eliminar los productos deslizando a la derecha actualizando el numero de productos y el balance.

Se hace uso de distintas librerías para la implementación de distintos componentes como *Dialogs* personalizados para escoger fechas, componentes de calendarios para visualizar tareas pendientes entre otros componentes que se ha trabajado a través de investigación propia e implementación propia por el desarrollador.

El diseño de la interfaz se ha realizado a través del propio *IDE Android Studio* modificando a mano los distintos *layouts* (**XML**) implementando otras librerías de investigación propia para la inserción de imágenes circulares y trabajando con los 2 principales tipos de *layouts* (*LinearLayouts*, *RelativeLayouts*).

Se ha trabajado con los distintos componentes dados durante el curso

Se ha trabajado con **Activities** para manejar distintas pantallas, habrá un *MainActivity* encargado de manejar distintos **Fragments**, cada pantalla del menú lateral es un *fragment* para trabajar de forma mas ligera para el usuario y cambiar el contenido de la pantalla por un *fragment* distinto sin la necesidad de cargas de *Activity* distintos.

Se ha trabajado con distintas listas **RecyclerView** y **Adapter** para la muestra de información, implementado librería **Room** como base de datos para almacenar configuración de conexión al servidor.

Se ha implementado **ViewPager** para realizar una página informativa con distintas páginas informando al usuario de las distintas funcionalidades de la aplicación deslizando entre pantallas.

Se ha diseñado la comunicación con el servidor a través de la clase **AsyncTask** para trabajarlo desde el *background* sin bloquear la interfaz del usuario y mejorar la experiencia

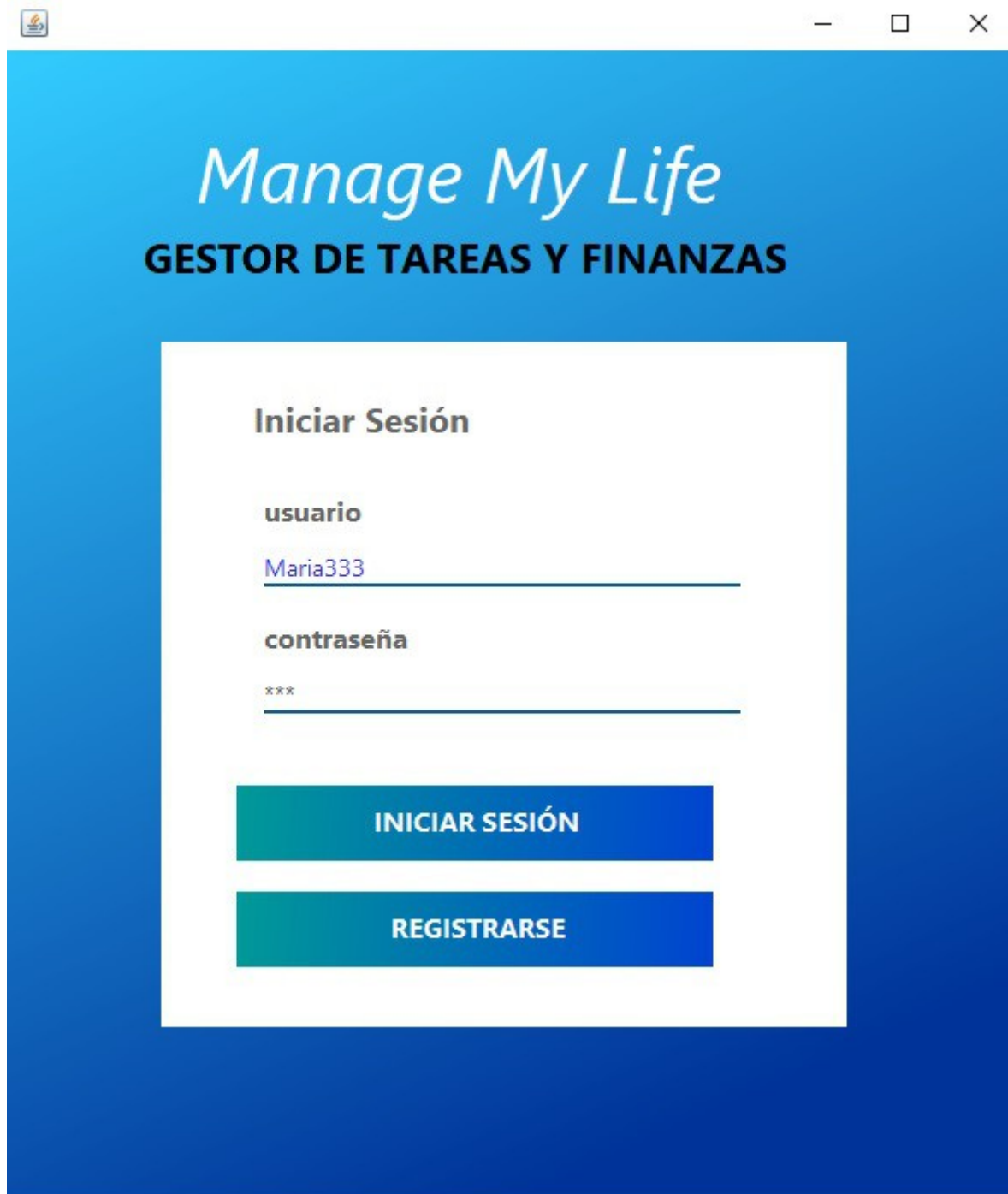
La navegación entre *activities* se ha utilizado **Intents** y pasos de datos (Datos del usuario conectado en el sistema) y los *fragments* han obtenido los datos del usuario a través del *Activity* padre (*MainActivity*)

Se ha implementado **librerías de investigación propia** como *AlarmReceiver*, *SwipeableRecyclerView* (permitir animación de swipe en los distintos registros), *CircleImageView* entre otros.

Implementación de **distintos componentes** para la implementación de distintas funcionalidades dadas durante el curso: *FloatingActionButton*, *Popups*, *ClickListeners*, *Callbacks*, *Views*..

La aplicación ha sido traducida también al inglés, si ejecuta la aplicación desde un terminal en español, el usuario podrá visualizar los distintos parámetros de la aplicación en español y si lo tiene en inglés, los podrá visualizar en inglés.

4.2. Aplicación de Escritorio



Manage My Life
GESTOR DE TAREAS Y FINANZAS

Iniciar Sesión

usuario
Maria333

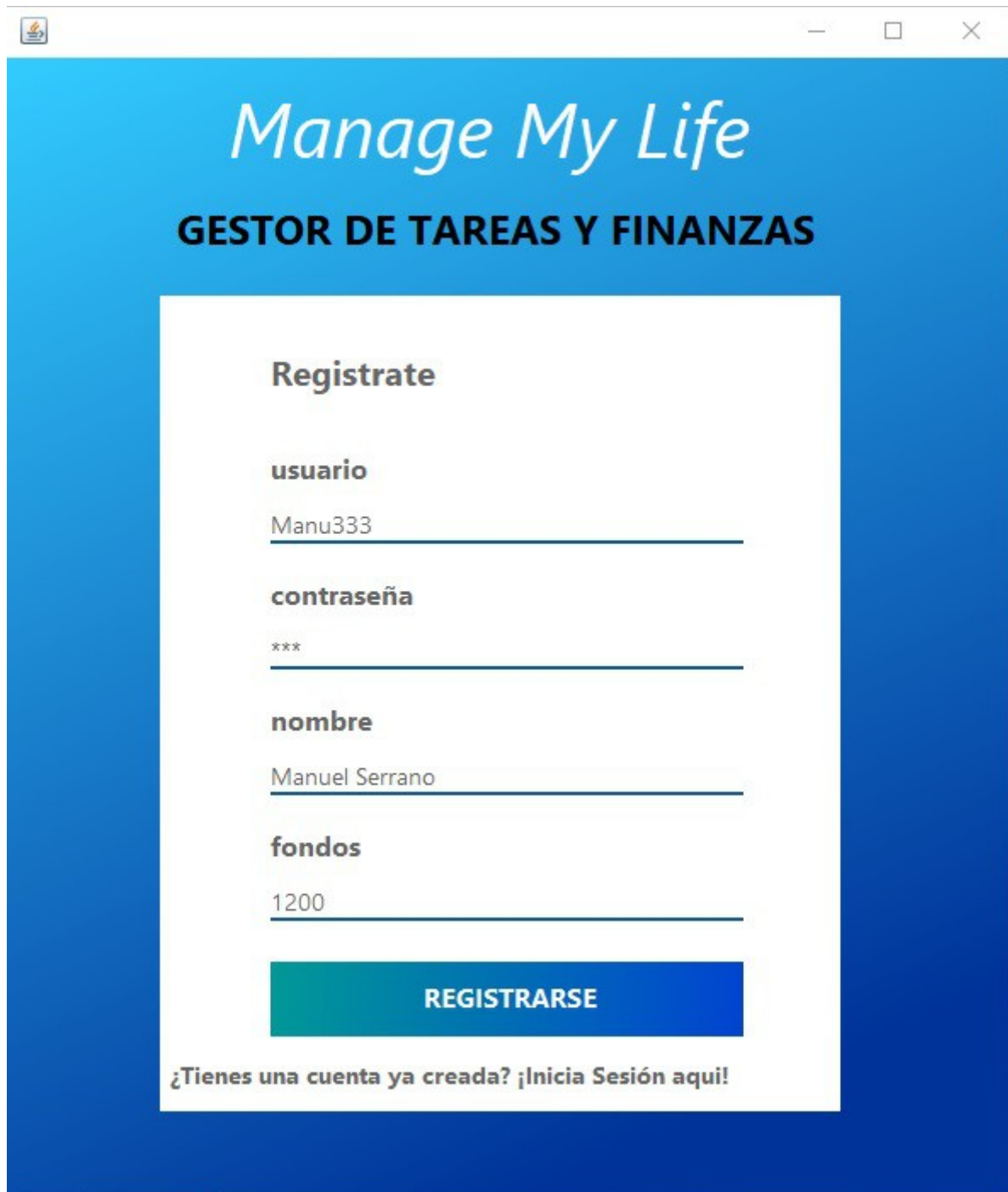
contraseña

INICIAR SESIÓN

REGISTRARSE

Ilustración 22: Swing - Diseño de la pantalla de Inicio de Sesión

La pantalla de inicio de sesión es donde el usuario podrá insertar sus credenciales para acceder al sistema, permitirá acceder a la pantalla de registro con el botón Registrarse.



Manage My Life

GESTOR DE TAREAS Y FINANZAS

Registrate

usuario

Manu333

contraseña

nombre

Manuel Serrano

fondos

1200

REGISTRARSE

¿Tienes una cuenta ya creada? ¡Inicia Sesión aquí!

Ilustración 23: Swing - Diseño de la pantalla de Registro

En esta pantalla el usuario podrá rellenar todos los campos (los cuales todos son obligatorios) y registrarse en el sistema, al registrarse enviará al usuario a la pantalla de Login



Ilustración 24: Swing - Diseño de la pantalla de Perfil

La pantalla de perfil podremos ver una pantalla con distintos valores como las tareas pendientes, tareas terminadas y el balance actual al igual que distintas opciones como un botón de ajustes donde se podrá actualizar distintos campos en una ventana modal al igual que añadir una imagen.

También podrá cerrar sesión para acceder con otra cuenta con el botón 'candado' el cual lanzará la pantalla de inicio de sesión.

Este panel al igual que los demás tendrá un panel lateral trabajando como un menú lateral tratando de parecerse al máximo a la aplicación Android, a partir de este menú lateral el usuario podrá navegar entre las distintas pantallas



Ilustración 25: Swing - Diseño de la pantalla de Tareas

Pantalla de tareas, la cual mostrará distintas tareas en un componente `Jtable` con el nombre de la tarea, su estado (Pendiente, Terminado) y su prioridad, también tenemos un botón de añadir tareas donde podremos ver un `Jdialog` para insertar tareas

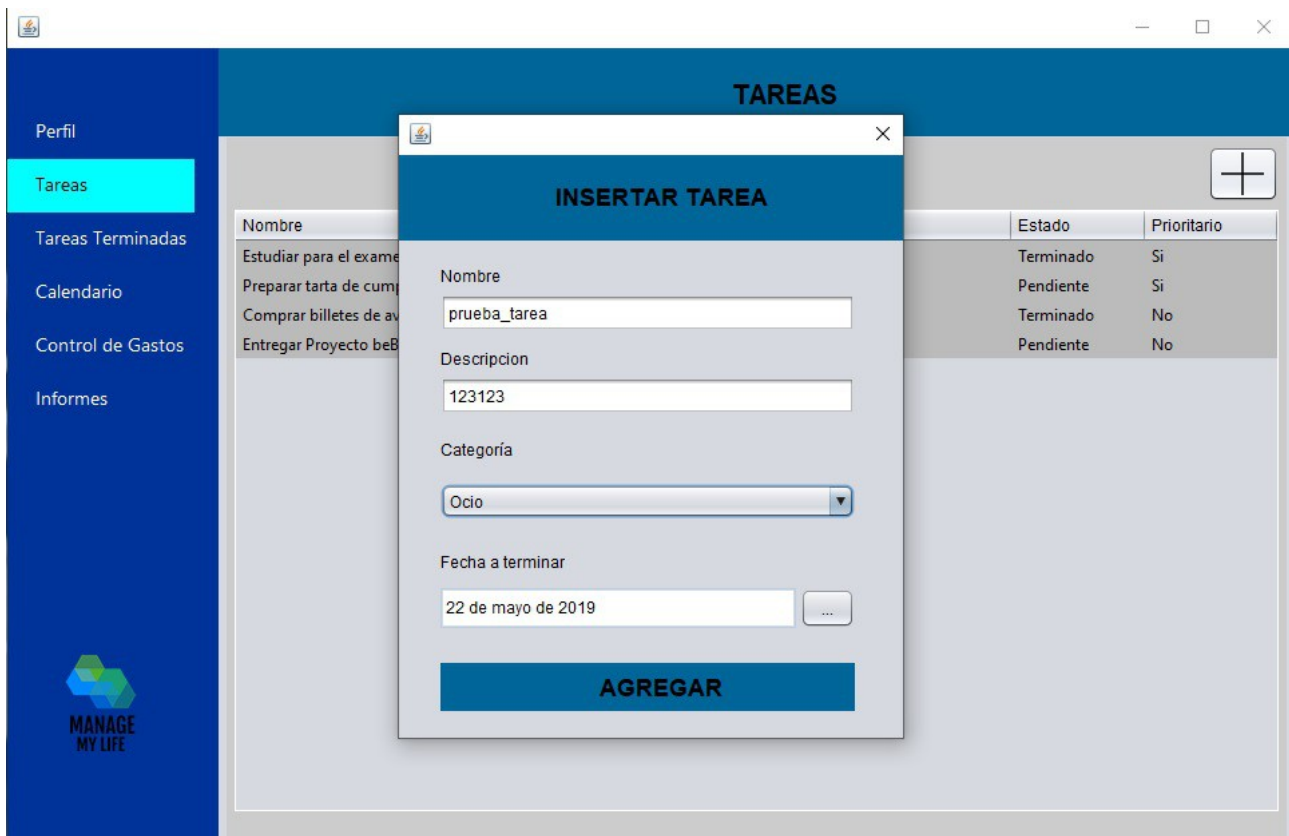


Ilustración 26: Swing – Diseño de la pantalla de insertar tareas

Aquí podremos ver el *JDialog* para insertar tareas donde podremos escribir el nombre de la tarea, descripción, categoría y la fecha a terminar (la cual es un componente de una librería externa explicada mas adelante).

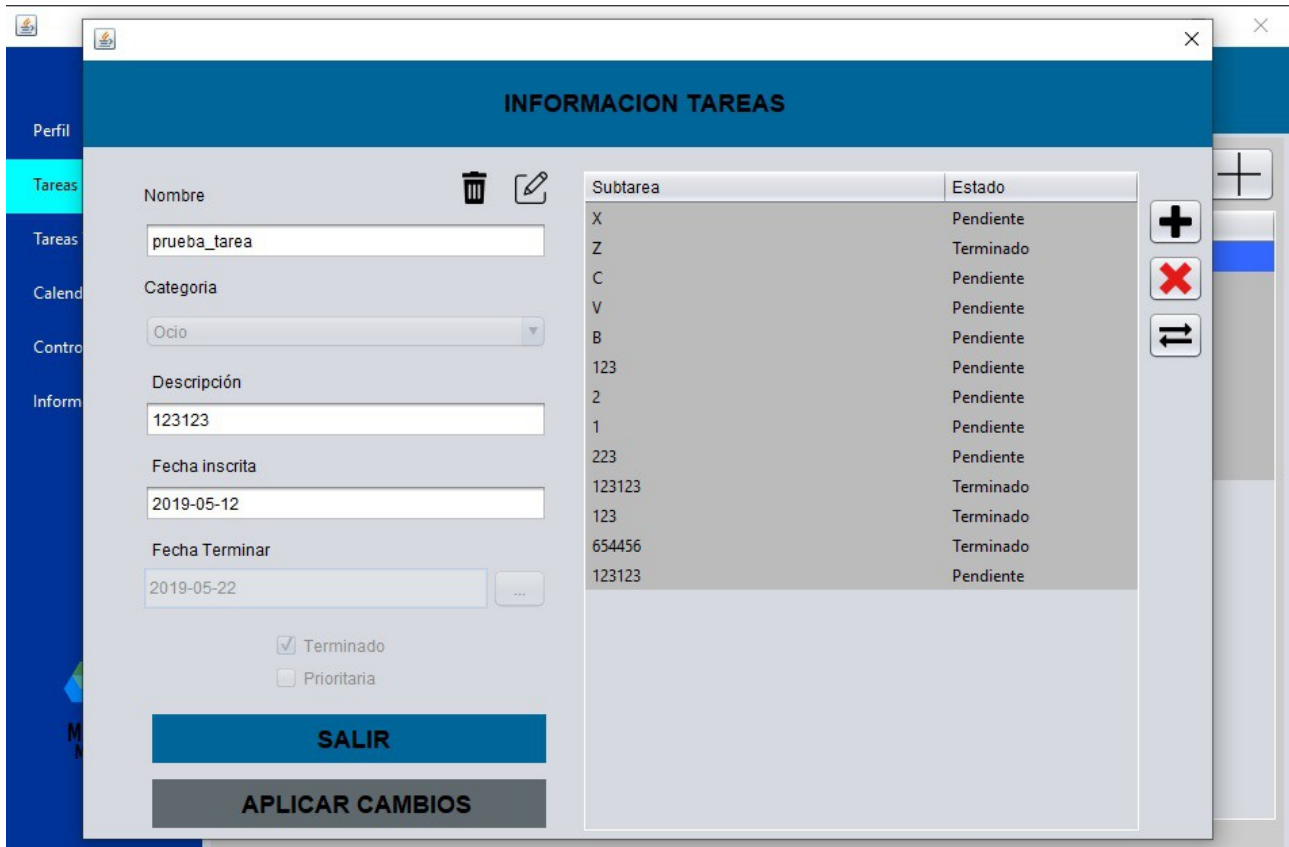


Ilustración 27: Swing – Diseño de la pantalla de información de tareas

Al pinchar en una tarea podemos visualizar un Jdialog con un Jtable con distinta información.

Nombre de la tarea, categoría, descripción, una fecha inscrita (fecha en la que se escribió la tarea), Fecha Terminar en la que se registró al insertar en la tarea, varios checkbox (terminado y prioritaria).

También podrá visualizar una lista de subtareas en la que podremos añadir, eliminar subtareas y cambiar el estado de las subtareas.

También podremos ver un botón de modificar tarea, el cual habilitará los distintos campos de la pantalla y habilitará el botón de 'Aplicar Cambios'.



Ilustración 28: Swing - Diseño de la pantalla de tareas terminadas

Podremos ver una lista parecida a la lista de tareas, esta mostrará el nombre, la descripción de la tarea y el tipo de tarea, tan solo mostrará las tareas que estén terminadas.

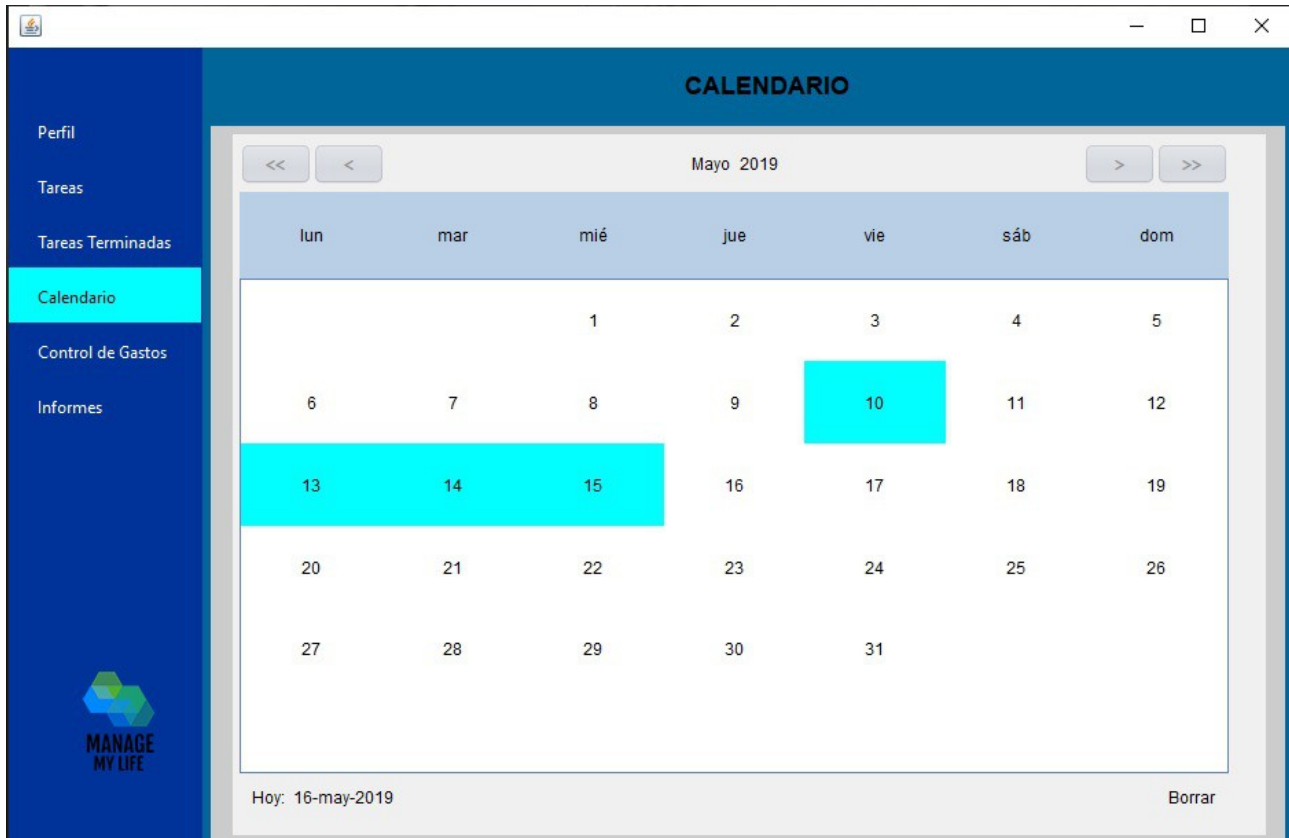


Ilustración 29: Swing – Diseño de la pantalla del Calendario

En esta pantalla podremos ver un calendario (componente de librería externa explicado mas adelante) con los distintos días donde se iluminarán los distintos días del calendario (aquellos días que hay tareas pendientes) e informando al usuario al interactuar con cada uno de ellos mostrando el día y la cantidad de tareas pendientes para ese día.



Ilustración 30: Swing - Diseño de la pantalla de Gastos

La pantalla de gastos se mostrará en un componente *JTable*, con distinta información como el nombre del gasto y la cantidad en total del gasto. Botón para agregar gasto donde se podrá agregar distintas listas de gastos indicando el nombre del gasto y el tipo de gasto y botón para añadir fondos al balance actual

También podemos ver distinta información como el numero de gastos y el balance actual que se actualizan automáticamente tras cualquier cambio en el balance o en la inserción de cualquier gasto.

The screenshot shows a web application window titled "PRODUCTOS". The main content area displays "1 PRODUCTOS" on the left, a central image of a man's face, and "2289.5 € BALANCE" on the right. Below the image are two buttons: a blue "+" button and a red "X" button. A table below these buttons lists products and their prices:

Producto	Precio
Producto de Prueba	10.5 €

At the bottom of the window are two buttons: "GUARDAR CAMBIOS" and "ELIMINAR GASTO".

Ilustración 31: Swing – Diseño de la pantalla de Productos

Pantalla de productos, este se mostrará al pinchar en un gasto, se mostrará una lista de productos, el nombre de productos, el precio del producto y un botón donde podrá eliminar los productos o agregar los productos.

También habrán 2 botones, uno para guardar los cambios y otro para eliminar el gasto, al eliminar el gasto se eliminarán todos los productos actualizando el salario contando todos los productos de ese gasto.

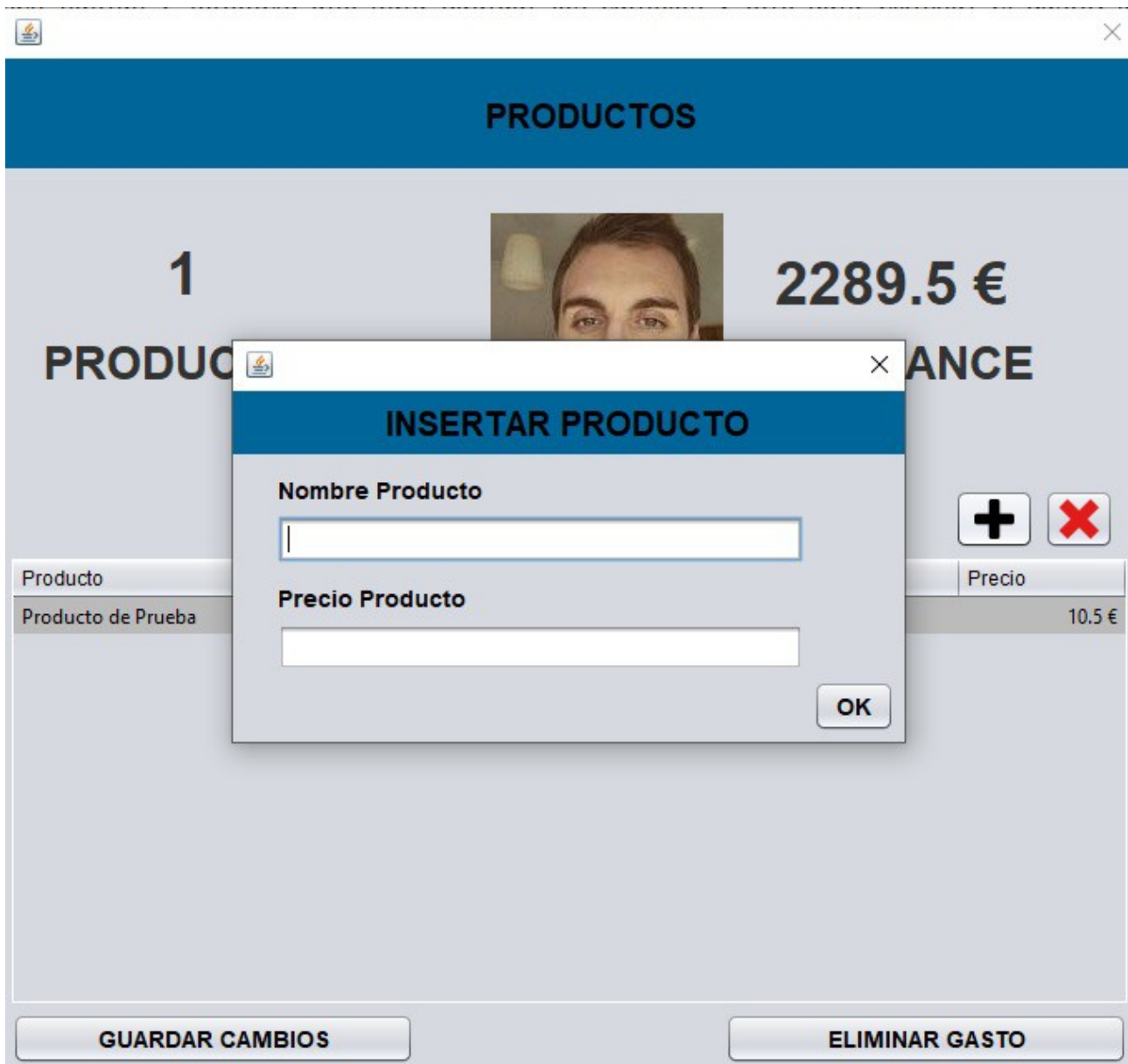


Ilustración 32: Swing – Diseño de la pantalla de insertar productos

Jdialog que aparece para insertar los distintos parámetros del producto a insertar, este esta compuesto por el nombre del producto y el precio del producto, al darle a Ok se insertará el nuevo producto en la lista, actualizándose la cantidad de productos y el balance actual.



Ilustración 33: Swing – Diseño de la pantalla de Informes

Panel con distintos informes, se generan los distintos informes clicando en cada uno de los distintos botones del panel. Cada diagrama se genera siguiendo el flujo de funcionamiento de informes a través del servidor explicado en el apartado **1.4. Informes**

Estos informes se cargarán a través de una petición indicando el tipo de informe que quiere recibir pinchando en el botón, el servidor recibirá la petición, creando el informe, generando un archivo PDF, realizando una conversión y un cifrado de esta conversión y enviándolo al cliente a través de flujos.

El cliente a través de la librería *awt* abrirá el lector de PDF que tenga en su sistema por defecto para poder leer el archivo PDF que se generará a partir de los datos recibidos por el servidor.

Las pantallas de inicio de sesión y registro se ha implementado una librería para mejorar el diseño *KGradientPanel* que permite crear paneles (en este caso un *background*) con efectos gradientes.

Las distintas pantallas de tareas, tareas terminadas y finanzas son tablas dinámicas que van mostrando la distinta información y son susceptibles a eventos, si el usuario hace clic en alguna de estas tablas se mostrarán distintas pantallas (*modal*) con la información solicitada y mostrando mas opciones como agregar algún registro, eliminar registros, modificar registros.

Al realizar alguna acción con alguna de estas opciones (agregar, eliminar, modificar cualquier tipo de registro (tareas, gastos, subtareas, productos, balance...)) se actualizará las tablas automáticamente mostrando al usuario la información actualizada.

5. Prototipos de Diseño

5.1. Prototipo de Android

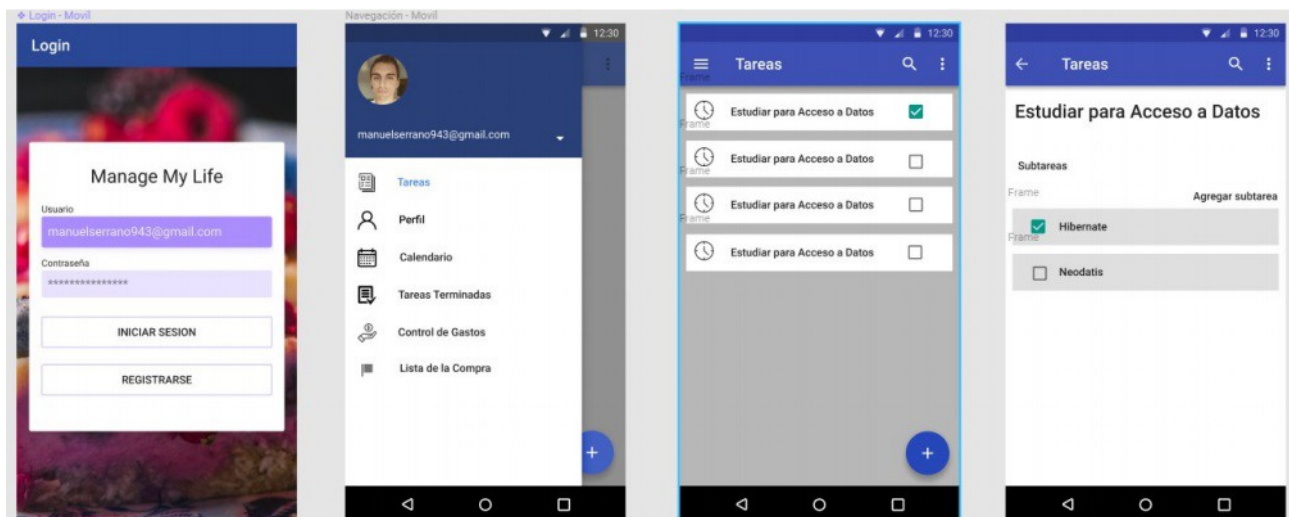


Ilustración 34: Diseño Prototipo de Android

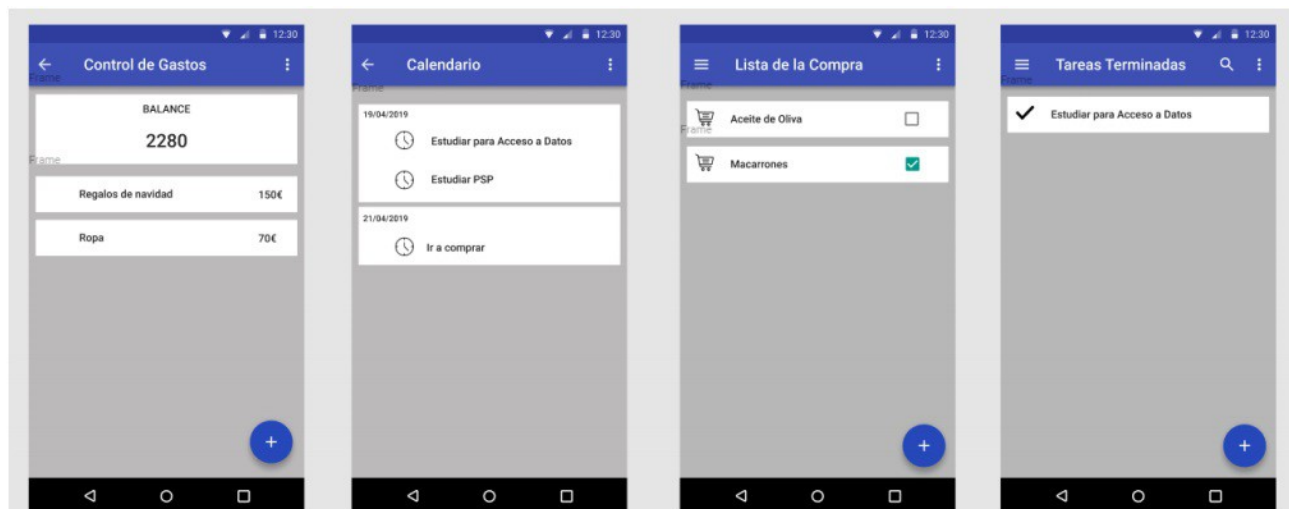


Ilustración 35: Diseño Prototipo de Android 2

5.2. Prototipo de Swing

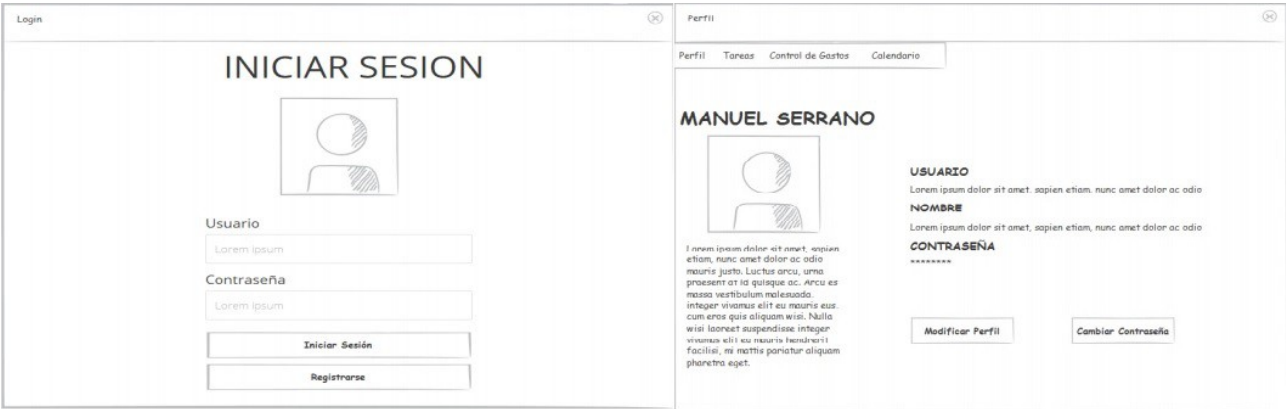


Ilustración 36: Diseño Prototipo de Swing

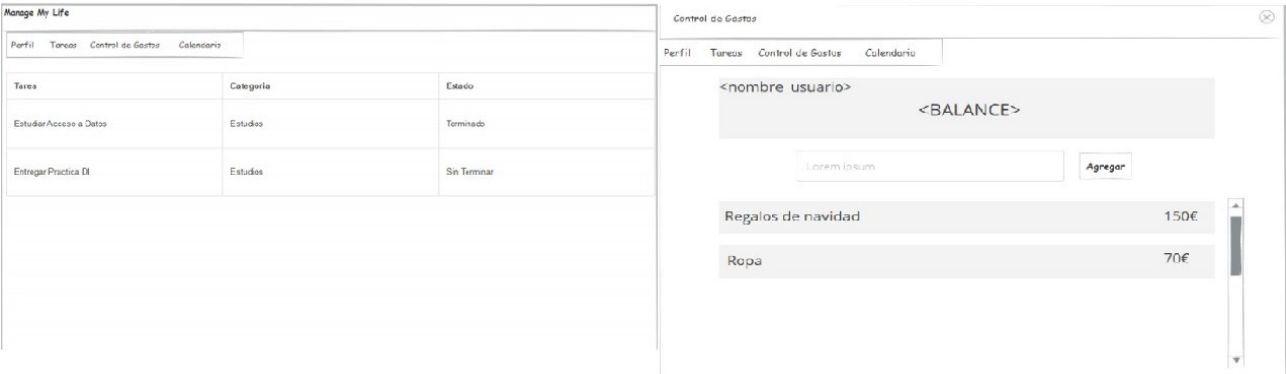


Ilustración 37: Diseño Prototipo de Swing 2

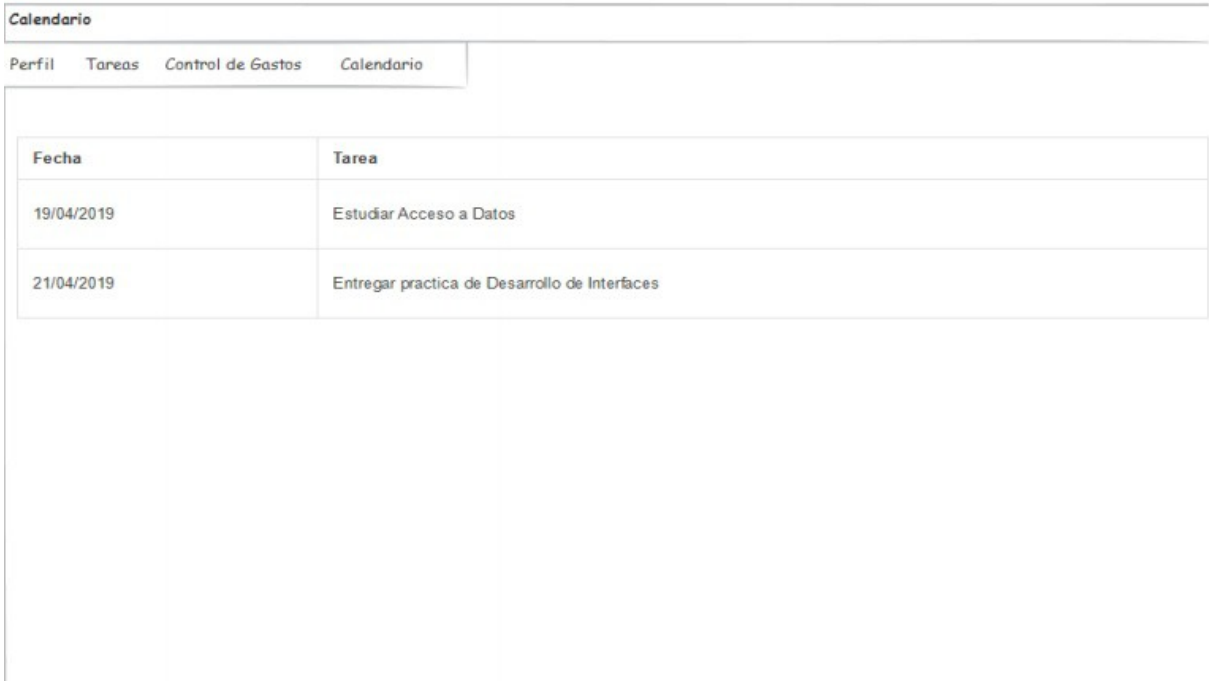


Ilustración 38: Diseño Prototipo de Swing 3

6. Medios Requeridos

6.1. Requerimientos Hardware

Sistema Operativo	Windows 10
CPU	I5-6300HQ 2.3GHz
Memoria RAM	8GB
Tarjeta Gráfica	Nvidia Geforce GTX 960M

Tabla 1: Requerimientos Hardware

Será necesario otros requisitos *hardware* como la posibilidad de un cable USB para instalar la APK desde el ordenador, hará falta un dispositivo disponible con capacidad suficiente para instalar la aplicación.

6.2. Requerimientos Software

Aplicaciones necesarias	Software
Aplicación web para diseño de casos de uso	DrawIO
Software para diseño del prototipo de la aplicación de escritorio	JustInMind
Software para diseño de diagrama de Gantt	GanttProject
Paquete software para base de datos	XAMPP
Cliente MySQL	HeidiSQL
Entorno de desarrollo para la aplicación de escritorio	NetBeans IDE
Entorno de desarrollo para la aplicación móvil	Android Studio
Aplicación web para diseño del prototipo de la aplicación móvil	Figma
Gestor de Proyecto	GIT
Procesador de textos para documentación	OpenOffice
Aplicación para diseño de informes para Swing	iReport
Aplicación web para obtener colores usados en la aplicación	Coolors.co
Aplicación web para diseñar otros tipos de diagramas	Creately

Tabla 2: Requerimientos Software para el desarrollo del proyecto

6.3. Recursos Humanos

La implementación del diseño, lógica, diseño e implementación de la base de datos junto con registros de pruebas, creación de servidor multihilo y conexión entre este y distintos clientes mas documentación junto manual de usuario adjunto ha sido implementado por el alumno Manuel Serrano Scholz.

La implementación de distintos componentes y lógica ha sido implementada con ayuda y colaboración del profesorado, tanto para las distintas mejoras del diseño, implementación de nuevas funcionalidades, corrección de errores, ideas de implementación y análisis de datos.

Se destaca el trabajo realizado durante la fase de practicas en la empresa *Numenti S.L* trabajando en el proyecto *BeBir* y *ScootCity*, durante esta fase de aprendizaje se ha investigado la posibilidad de implementar distintos componentes usados en estos proyectos adaptándolos al lenguaje JAVA en **Manage My Life**. Algunos de estos componentes son: *Sliders, refreshers, checkbox*.

6.4. Requerimientos mínimos

Para poder trabajar con todas las funcionalidades de la aplicación móvil y la aplicación de escritorio será necesario distintos requisitos.

Aplicación Servidor

- Equipo con los **Requerimientos Hardware** necesarios
- Sistema Operativo instalado junto con drivers y JDK instalado
- Base de datos funcional MySQL local con un esquema de tablas idéntico al esquema de base de datos.
- Configuración del **config.txt** con credenciales de base de datos

Aplicación de Escritorio

- Equipo con los **Requerimientos Hardware** necesarios.
- Sistema operativo instalado junto con drivers y JDK instalado
- Base de datos funcional MySQL local (generación de informes)
- Servidor arrancado
- Configuración del **config.txt** con credenciales del servidor y base de datos

Aplicación móvil

- Emulador Android o dispositivo móvil para ejecutar aplicación móvil
- Servidor arrancado
- Configurar en pantalla Login a través de ajustes una **dirección IP y Puerto del servidor**

7. Recogida de Información

La idea principal del proyecto, distribución de distintos componentes y la forma en como se mostrará los datos en cada uno de las distintas funciones del proyecto se tenía clara desde un principio, trabajar con componentes ya practicados en anteriores proyectos dados durante el curso.

Pero hacía falta mas información, el cual se estableció un tiempo de recogida de información, investigación de librerías necesarias, muestra de datos en distintos componentes y utilizar un diseño respetando las normas "*Material Design*" e intuitivo para facilitar la gestión de la aplicación orientado a usuario.

Se le incluyo tiempo de investigación y preparación para la base de datos entre otros datos importantes para la implementación de los distintos requisitos funcionales

1) ¿Que información se almacena en las tareas?

Necesitamos conocer el nombre de la tarea, la descripción, categoría a la que pertenece, conocer si es prioritaria, su estado, fecha en la que se escribió la tarea, fecha en la que se debe de realizar la tarea gestionada por el usuario y conocer a que usuario pertenece la tarea.

2) ¿Como se mostrará las tareas y los gastos en las distintas aplicaciones?

Respetando normas de *material design* y para cumplir con el objetivo de diseñar una aplicación fácil de usar se utilizará componentes de navegación como listas y tablas, tratando de permitir al usuario de interaccionar directamente con estas, buscando distintos tipos de interacción y eventos (Métodos `onClick`, listas deslizables, botones...) para los distintos requisitos funcionales de cada apartado del proyecto: Perfil, tareas, gastos, informes.

3) ¿Como se generarán los distintos informes y como se visualizará en cada aplicación?

El usuario podrá generar automáticamente estos informes en la aplicación de escritorio usando la librería JasperReport utilizando distintos botones diferenciados por temática para cada tipo de informe. Estos informes los gestionará el servidor y se le mandará al usuario los informes que el usuario pida a través de flujos.

4) ¿Que datos/información obtendrá y podrá consultar el usuario?

El usuario podrá generar información directamente en la aplicación con la generación de tareas, gastos e incluso con la generación de datos en su pantalla de perfil.

Internamente se generará información indirectamente gestionado por el servidor a través de una base de datos MySQL, algunos datos de este tipo serían gráficos y estadísticas que informan sobre la evolución del usuario respecto al uso de la aplicación, visualizando las tareas que ha completado, consultado así sus logros personales

También se publicarán informes con información pública sobre otros usuarios permitiendo al usuario visualizar datos relativos a otros usuarios como por ejemplo la cantidad de usuarios que utilizan la aplicación, y poder consultar la fecha de su ultima actividad al igual que otros datos que pueden ser de interés.

4) ¿Que información será susceptible de cambio para el usuario?

Manage My Life es una aplicación totalmente interactiva y gestionada por el usuario, datos de su pantalla de perfil, tablas, listas de los datos de las tareas, listas de los productos y gastos incluso los datos de los informes personales dependerán de los datos e información que registre el usuario

5) ¿Como se configura la conexión al servidor y a la base de datos?

Manage My Life es una aplicación interactiva con el usuario, y esto implica también permitir al usuario escoger la dirección IP y puerto desde el que se va a conectar al servidor.

La aplicación servidor encargada de recoger las consultas y objetos del cliente y obtener los datos de la base de datos se configurará a través de un archivo config.txt en su directorio raíz. En este se indicará según un orden establecido ciertos parámetros (Nombre de la base de datos, usuario de la base de datos, contraseña de la base de datos).

La aplicación escritorio se conectará a través de un archivo config.txt en su directorio raíz, este también tendrá ciertas directivas que se indicarán en un archivo de manual. Este archivo se indicará la dirección IP del servidor al que se va a conectar y el puerto del servidor.

La aplicación móvil podrá visualizar el usuario un botón de ajustes en la pantalla principal desde donde podrá escribir la dirección IP y el puerto al que se va a conectar, esto será necesario incluso antes de iniciar sesión en la aplicación para obtener una conexión establecida, al configurar los 2 parámetros la aplicación informará a través de una alerta temporal en la pantalla superior de la aplicación a que dirección IP se ha conectado y desde que puerto se ha conectado.

8. Preparación de Entorno de Ejecución e Instalación

Creación de archivo ejecutable para la instalación de la aplicación y dependencias

Para una instalación y ejecución cómoda para el usuario se ha preparado un archivo (script) ejecutable para la instalación de las aplicaciones desde donde el usuario dando permisos previos al archivo, podrá ejecutarlo e instalarse de forma automática en el sistema permitiendo a su vez arrancar automáticamente la aplicación.

Se proporcionará otro archivo para la ejecución automática de la aplicación sin necesidad de instalar todo nuevamente.

Para la construcción del script se ha trabajado los distintos comandos para la extracción de carpetas y archivos al directorio `/opt/`, proporcionar permisos necesarios a los archivos de servicios y carpetas de informes, a la conexión con *MySQL* y creación de base de datos, tablas e inserción de datos y configuración del servicio para arrancar el servidor desde el background, siendo totalmente transparente al Usuario.

El script esta conformado por los siguientes comandos:

```
sudo cp -r * /opt
```

Comando que permite mover todo el contenido de la instalación a `/opt/` desde donde se gestionará los siguientes comandos

```
sudo chmod 755 /opt/Servidor/server.sh
```

Comando que aporta de permisos totales al creador del contenido y permisos de ejecución y lectura a los demás usuarios al fichero **server.sh** encargado de cargar en el sistema a través de la ejecución de un servicio el servidor.

```
sudo chmod -R 777 /opt/Escritorio/informes_generados
```

Comando que aporta permisos totales a la carpeta y subcarpeta/archivos referente a los informes que el usuario puede abrir desde la aplicación, necesario para permitir acceso al software nativo del sistema para la apertura de archivos de extensión PDF.

```
sudo chmod -R 777 /opt/Servidor/reports
```

Comando que aporta permisos totales a la carpeta y subcarpetas/archivos referente a los informes que genera el servidor para el correcto acceso a las distintas funcionalidades comprometidas a la creación de PDF desde la librería JasperReport.

```
sudo chmod -R 777 /opt/ManageMyLife/Servidor/media
```

Comando que aporta permisos totales a la carpeta media para almacenar las imágenes cargadas por el usuario.

```
sudo chmod +x /opt/ManageMyLife/Escritorio/ejecutar.sh
```

Comando que aporta permisos de ejecución al fichero ejecutar.sh para formar el acceso directo en el escritorio.

```
mysql -u root -p < bd.sql
```

Comando que permite acceder al servicio mysql con usuario root y ejecutar el fichero **bd.sql** el cual contiene las distintas sentencias para crear la base de datos, crear las distintas tablas necesarias e inserción de distintos datos para poblar la base de datos con información.

```
sudo mv /opt/Servidor/server.service /etc/systemd/system/server.service
```

Comando que permite mover el fichero **server.service** (encargado de gestionar el Servidor en el background siendo transparente para el usuario) a */etc/systemd/system/*

```
sudo mv /opt/ManageMyLife/Escritorio/escritorio.desktop /usr/share/applications/escritorio.desktop
cp /usr/share/applications/escritorio.desktop ~/Escritorio/escritorio.desktop
sudo chown "$USER" ~/Escritorio/escritorio.desktop
sudo chmod +x ~/Escritorio/escritorio.desktop
```

Conjunto de comandos que mueve el fichero (creado previamente) **escritorio.desktop** que hace referencia al acceso directo que se creará en el Escritorio para la aplicación Swing, se moverá y copiará el fichero a las distintas rutas requeridas, se le dará al usuario actual del sistema la propiedad del archivo para poder ejecutarlo y se le aportará permisos de ejecución para poder ejecutarlo.

```
systemctl daemon-reload
```

Comando que permite recargar la lista de servicios y regenerar el árbol de dependencias permitiendo detectar el nuevo servicio cargado en el sistema


```
systemctl enable server.service
```

Comando que permite habilitar el servicio Servidor para poder arrancarlo en el sistema

```
systemctl start server.service
```

Comando para arrancar el servicio Servidor, de esta forma el servidor permanecerá arrancado en el background siendo totalmente oculto al usuario.

```
cd /opt/Escritorio/  
java -jar Escritorio.jar
```

Comando para acceder al directorio donde se encuentra la aplicación de escritorio y lo ejecuta, esto ejecutará automáticamente la aplicación al instalarse.

Creación de archivo ejecutable para la preparación del entorno de ejecución

Se prepara un archivo ejecutable que permite instalar las distintas dependencias en el sistema, estas dependencias están compuestas por Java y un servicio MySQL con un usuario configurado (por defecto configuraremos el usuario root) para poder acceder a el desde una aplicación externa, ya que el Servidor como servicio deberá acceder al servicio MySQL desde el usuario configurado para poder acceder a los distintos datos de las distintas tablas de la base de datos.

Este script ejecutará los siguientes comandos:

```
sudo apt-get install openjdk-8-jdk
```

Comando para instalar Java 8, necesario para ejecutar la aplicación la cual esta escrita en Java y es necesario el JDK para ejecutarlo.

```
sudo apt-get install mysql-server
```

Comando para instalar MySQL server, gestor de base de datos necesario que utilizará la aplicación para las distintas funcionalidades.

```
sudo mysql -u root -p < config_mysql.sql
```

Accederemos a mysql con el usuario root con permisos privilegiados (la cual no tiene configurado aún ninguna contraseña de acceso por lo cual, no se nos permitirá acceder desde una aplicación externa) y le indicaremos que ejecute el archivo **config_mysql.sql** el cual contiene distintas acciones:

```
use mysql;  
update user set authentication_string=password('123456') where user='root';
```

Actualizará el usuario root aplicandole una contraseña en este caso le indicaremos la contraseña '123456' el cual será por defecto en el archivo **config.txt** del servidor (Parámetros de conexión por defecto entre servidor-bdd).

Seguido de ello actualizaremos a través del siguiente comando

```
update user set plugin="mysql_native_password" where User='root';  
flush privileges;
```

Nos permitirá actualizar correctamente la contraseña, *este comando es necesario tras la versión 5.7 de MySQL para la actualización correcta de la contraseña del usuario root.*

Finalmente con flush privileges actualizaremos los cambios realizados en el usuario root permitiendo trabajar correctamente con el archivo **instalación.sh** para la instalación de las distintas aplicaciones, creación del servicio e inserción de base de datos, tablas y datos en MySQL

9. Uso de Librerías

Aplicación Móvil	
Nombre	Enlace
CircleImageView	https://github.com/hdodenhof/CircleImageView
Square Time Up	https://github.com/square/android-times-square
Custom Alerter	https://github.com/Tapadoo/Alerter

Tabla 3: Librerías Externas en Android

CircleImageView

Librería para Android que permite cargar imágenes con un efecto redondeado, permitiendo utilizar imágenes (*ImageView*) con una forma circular sin necesidad de aplicar técnicas CSS.

Uso: He usado esta librería en mi aplicación móvil para colocar una imagen en la pantalla de perfil y en el menú lateral

Square Time Up

Librería que trabaja de forma similar que un *CalendarView* pero permitiendo funcionalidades extras como añadir *listener* mas sencillos y completos pintando los distintos días y sin las limitaciones del uso de un calendario nativo.

Uso: He usado esta librería en mi aplicación para cargar la pantalla del calendario, trabajando con las funcionalidades *listener* para cargar un *alerter* personalizado (**Custom Alerter**) y remarcando con color los días que contienen tareas pendientes.

Librería que permite cargar con un *timer* interno una notificación de tipo alerta en la parte superior de la pantalla permitiendo muchas funcionalidades y eventos permitiendo informar al usuario de cualquier suceso en la aplicación.

Uso: He usado esta librería en mi aplicación para cargar una alerta a la hora de elegir una fecha del calendario informando al usuario del día que ha seleccionado y la cantidad de tareas pendientes que tiene ese día, si el usuario lo decide, al pinchar en la alerta le llevará directamente a la lista de tareas.

Aplicación Escritorio	
Nombre	Enlace
KGradientPanel	https://github.com/k33ptoo/KGradientPanel
LGoodDatePicker	https://github.com/LGoodDatePicker/LGoodDatePic ker
Absolute Layout	Librería Interna

Tabla 4: Librerías Externas en Swing

KGradientPanel

Librería que nos carga varios componentes 'Panel' con efectos *Gradient* permitiendo utilizar una interfaz visual mucho mas elegante.

Uso: He usado esta librería en mi aplicación Swing para el diseño de la pantalla de Inicio de sesión y la pantalla de Registro.

También he aplicado estos panel en forma de botón en la aplicación.

LGoodDatePicker

Librería que nos permite cargar distintos componentes relacionados con fechas, nos permite cargar un componente para seleccionar fechas con distintas funcionalidades como aplicar colores, fuentes personalizadas, tamaño, habilitar edición por teclado..

También permite cargar un calendario independiente al componente de fechas el cual también se puede personalizar con distintos eventos y colores.

Uso: He usado esta librería en mi aplicación para realizar la pantalla de Calendario y el uso del componente de fechas para seleccionar fechas en las funcionalidades de Agregar Tarea y Modificar Tarea.

Absolute Layout

Librería que permite al usuario diseñar las distintas interfaces y componentes sin tener en cuenta el tamaño o donde estén colocados los demás componentes permitiendo facilitar el diseño de la interfaz sin que colisione y desordene toda la interfaz al intentar colocar un componente en el campo reservado de otro.

Uso: He usado esta librería para ayudarme con el diseño de toda la interfaz de la aplicación swing

