Aplicación multiplataforma de compra y gestión de libros electrónicos

Andrés García Payá y Alexis Willy Andía Rosales

Contenido

[1. Resumen 3](#_Toc195283852)

[2. Introducción 3](#_Toc195283853)

[3. Objetivos 4](#_Toc195283854)

[4. Palabras clave 4](#_Toc195283855)

[5. Análisis de contexto 5](#_Toc195283856)

[6. Modelo de datos 5](#_Toc195283857)

[6.1. Resumen 5](#_Toc195283858)

[6.2. Tablas 5](#_Toc195283859)

[6.2.1. Usuario 5](#_Toc195283860)

[6.2.2. Biblioteca 5](#_Toc195283861)

[6.2.3. Libro 6](#_Toc195283862)

[6.2.4. Deseo 6](#_Toc195283863)

[6.2.5. Biblio\_libro 7](#_Toc195283864)

[6.2.6. Saga 7](#_Toc195283865)

[6.2.7. Idioma 7](#_Toc195283866)

[6.2.8. Genero 7](#_Toc195283867)

[6.2.9. Autor 7](#_Toc195283868)

[6.2.10. Libro\_idioma 8](#_Toc195283869)

[6.2.11. Libro\_autor 8](#_Toc195283870)

[6.2.12. Libro\_genero 8](#_Toc195283871)

[6.2.13. Editorial 8](#_Toc195283872)

[6.2.14. Libro\_edit 9](#_Toc195283873)

[6.3. Diagrama E.R 9](#_Toc195283874)

[6.4 Diagrama UML 10](#_Toc195283875)

[7. Diseño 12](#_Toc195283876)

[7.1. Elementos comunes 12](#_Toc195283877)

[7.1.1. Login, registro, recuperar contraseña 12](#_Toc195283878)

[7.2. UI de la aplicación de escritorio 14](#_Toc195283879)

[7.2.1. Vista usuario 14](#_Toc195283880)

[7.2.2. Vista Editorial 15](#_Toc195283881)

[7.3. UI de la aplicación móvil 16](#_Toc195283882)

[8. Casos de uso 17](#_Toc195283883)

[9. Diagrama de clases 18](#_Toc195283884)

[9.1. Versión de escritorio 18](#_Toc195283885)

[9.1.1. Login 18](#_Toc195283886)

[9.1.2. MandaCorreo 18](#_Toc195283887)

[9.1.3. ManejaImagen 18](#_Toc195283888)

[9.1.4. ManejaBibliotecaLocal 18](#_Toc195283889)

[9.1.5. ManejaBibliotecaLibro 18](#_Toc195283890)

[9.1.6. Conexión 19](#_Toc195283891)

[10. Planificación 19](#_Toc195283892)

[10.1. Diagrama de Gantt 19](#_Toc195283893)

[10.2. Definición de Recursos y Logística 19](#_Toc195283894)

[11. Bibliográfica 20](#_Toc195283895)

# Resumen

Esta aplicación constara de una versión de PC y una versión móvil. Ambas versiones tendrán funciones similares. Tanto la versión de PC como la versión de móvil están planteadas para que un usuario normal no esté obligado a usar las dos. Algunas de las funciones compartidas entre las versiones, son la capacidad de utilizar la tienda, descargar libros, leerlos, modificar ciertas características.

Las principales diferencias entre las dos versiones vienen de dos funciones concretas siendo estas la capacidad de la versión móvil de escanear una imagen que pueda contener el título de los libros, permitiendo buscar en la base de datos si ese libro está disponible en la tienda y la versión de escritorio, que tendrá un apartado exclusivo para las editoriales, lo que las dará la capacidad de subir libros a la tienda. La versión de escritorio también contara con la capacidad administrar la biblioteca interna de un libro electrónico.

# Introducción

La idea de este proyecto surge de una reciente noticia relacionada con Amazon, en la que comentan que la empresa elimina la capacidad de descargarse los libros que tenga uno ya comprados, así como también eliminarlos y o actualizarlos del propio dispositivo. Si bien esto es algo que solo afectaría a los usuarios de Kindle, estos mismos son gran parte del mercado, por lo que puede suponer un problema. Esto también imposibilita a los usuarios de Kindle a acceder a la parte de su biblioteca que pertenezca a Amazon si no disponen de una conexión a internet.

Otra de las razones por las que se escogió el proyecto fue que, como consumidores, nos parecía más cómodo tener todo lo relacionado con el tema en una sola aplicación, así como hacen otros sectores como pueda ser el de los videojuegos, que en PC cuenta con diversas aplicaciones desde donde se puede comprar el juego deseado, así como ejecutarlo, entre otras cosas. La idea es aplicar el enfoque usado en estos sectores al mundo de los libros electrónicos, ya que actualmente, no existe una aplicación que fusione la idea de la compra con el apartado de gestión de los libros.

De estos dos motivos principalmente nace la idea de este proyecto, una alternativa que le brinde al usuario un control mayor sobre el contenido digital que posee, así como también una mayor comodidad a la hora de tratar con él.

En resumen, este proyecto nace de la intención de mejorar la experiencia de los usuarios, dándoles una herramienta para adquirir y administrar sus libros de unas forma sencilla, eficiente, centralizada y menos restrictiva que el reste de opciones del mercado.

# Objetivos

El objetivo de esta aplicación es acercar lo más posible las ventajas que tienen los libros físicos sobre los digitales y viceversa. También se tiene la intención de centralizar todo el proceso relacionado a la compra de libros y el manejo de estos.

Para cumplir con estos objetivos, se ha decidido crear dos versiones de la aplicación, la de móvil y la de escritorio, estando la versión móvil enfocada en un tipo de usuario más inexperto por decirlo de alguna manera, ya que esta presenta unas funcionalidades más limitadas en comparación con la versión de escritorio que está diseñada con un usuario más avanzado o una editorial en mente.

La versión móvil brindara a los usuarios la opción de acceder a la tienda de la aplicación, añadir libros a su lista de deseados, descargar los libros que tenga comprados y leerlos. También podrán escanear la foto de una portada o una hoja donde tengan apuntado el título, para comprobar si se dispone de este en a la tienda. Para llevar a cabo esto se utilizará tesseract OCR, que estará integrado en un servidor y al que se le enviaran las fotos a través de una API. Ya que uno de los motivos de la aplicación es no depender en sobremedida de grandes se empresas se ha optado por este OCR sobre Google Cloud visión.

La versión de escritorio contara con las mismas funciones a excepción de la capacidad escanear imágenes. También contará con la capacidad de tener no solo la biblioteca digital, sino también con la opción de tener varias bibliotecas locales, en las que el usuario podrá almacenar también libros obtenidos de otras tiendas. Si lo desea, podrá leer los libros de su biblioteca desde la propia aplicación. Esta no es la única diferencia, también tendrá la capacidad convertir el formato de sus libros, entre .pdf y .epub, que es el utilizado en los libros electrónicos. Por último, también podrá manejar de estos, permitiéndole añadir o eliminar libros.

En lo que respecta a las editoriales, estas tendrán una sección oculta a la que solo podrán acceder ellas, que las permitirá subir libros al servidor, aumentando así su catálogo disponible en la tienda. Para ello, se les asignara a los usuarios un rol en función de si son una editorial o un consumidor. El rol de editorial se dará a mano mientras el que el rol de usuario común se le dará automáticamente al resto.

Para complementar las dos versiones, se dispondrá de un servidor de descarga, donde estará integrado el OCR, y desde donde se manejarán los pagos a través de la API de stripe.

# Palabras clave

API, versión de escritorio, versión móvil, libro electrónico, OCR, roles, aplicación multiplataforma, conversión de formatos

# Análisis de contexto

# Modelo de datos

## Resumen

Para hacer la base el apartado de base de datos se ha decidido utilizar PostgreSQL ya que es un proyecto open source por lo que se alinea con nuestra visión de marca. Otra de las razones por la que se decidió usar PostgreSQL fue que se trata de una basa relacional orientada a objetos, lo que la hace ideal cuando se usa en conjunto de un lenguaje vasado orientado a objetos como pueda ser java. Por último, se escogió PostgreSQL por su escalabilidad, ya que, en el contexto de este proyecto no es necesario manejar una gran carga de trabajo, pero deja la puerta abierta a esa capacidad.

## Tablas

### Usuario

* **Resumen**: Esta tabla contiene la información de las cuentas de los usuarios que hayan creado una cuenta. Se crea de forma automática cada vez que un usuario se registra en la aplicación, ya sea la versión de móvil o la de escritorio.
* **Campos:**
  + **ID\_usuario:** Es la clave principal del usuario, es incremental.
  + **Nombre**: Es el nombre real del usuario, puede ser NULL ya que la información se termina de añadir si se quiere en el apartado de perfil.
  + **Apellidos:** Son los apellidos reales del usuario, pueden ser NULL ya que la información se termina de añadir si se quiere en el apartado de perfil.
  + **Usuario:** Es el nombre con el que podrá iniciar sesión el usuario.
  + **Correo:** Es el correo asociado a la cuenta.
  + **Contraseña:** Es la contraseña haseada del usuario.
  + **Semilla:** Es la semilla utilizada para encriptar la contraseña del usuario.
  + **Cartera:** Es el saldo del que dispone el usuario.
  + **Fecha\_registro:** Es la fecha en la que el usuario creo su cuenta.
  + **Ultimo\_registro:** Es la fecha de la última conexión de la cuenta asociada
* **Relaciones:**
  + Relación 0:1 con editorial ya que una editorial solo puede tener un usuario, pero un usuario no tiene por qué tener editorial.
  + Relación n:m con usuario ya que un usuario puede desear n libros y un libro puede ser deseado por n usuarios.

### Biblioteca

* **Resumen:** Es la tabla que almacena los libros que se hayan comprado en la aplicación, por lo que no tiene en cuenta los libros de las posibles bibliotecas locales del usuario.
* **Campos:**
  + **ID\_biblioteca:** Es la clave primaria de la biblioteca, es incremental.
  + **ID\_FK\_usuario:** Es la clave foránea de usuario.
  + **Ultimo\_registro:** Es la fecha de la última vez que se añadió un libro a esta biblioteca.
* **Relaciones:**
  + Relación 1:1 con la tabla usuario ya que un usuario solo puede tener una biblioteca.
  + Relación n:m con libro ya que un libro puede está en varias bibliotecas y barias bibliotecas pueden tener un libro

### Libro

* **Resumen:** Es la tabla que almacena la información de los libros que están a la venta al público desde la aplicación. Tiene una relación n:1 con saga.
* **Campos:**
  + **ID\_libro:** Es la clave primaria de la tabla, es incremental. En este caso es el ISBN
  + **Título:** Es el título del libro.
  + **Fecha\_publi:** Es la fecha en la que se publicó el libro.
  + **Precio:** Es el precio actual del libro.
  + **Descuento:** Es el descuento que pueda tener el libro en el momento. Puede ser NULL ya que no tiene por qué tener descuento alguno.
  + **DRM:** Indica si el libro tiene DRM.
  + **N\_paginas:** Indica el número de páginas que tiene el libro.
  + **N\_votos:** Es el número de veces que este libro se ha calificado, se utilizara para sacar la puntuación medio del libro
  + **Sinopsis:** Es una breve descripción de la trama del libro
  + **Valoración:** Es la nota media del libro.
  + **URLibro:** Es la dirección desde la que se puede descargar el libro.
  + **URLportada:** Es la dirección desde la que se puede descargar la portada del libro.
  + **ID\_FK\_saga:** Es la clave foránea de la tabla saga. Puede ser NULL ya que un libro no tiene por qué pertenecer a una saga.
* **Relaciones** 
  + Relación n:0 con saga ya que un libro puede o no estar en una saga, pero una saga puede tener n libros.
  + Relación n:m con autor ya que un libro puede tener n autores y un autor escribir n libros.
  + Relación n:m con idioma ya que un libro puede estar en n idiomas y un idioma puede estar en n libros.
  + Relación n:m con genero ya que un libro puede tener n géneros y un género puede estar en n libros.
  + Relación n:m con usuario ya que un usuario puede desear n libros y un libro puede ser deseado por n usuarios.

### Deseo

* **Resumen:** Tabla intermedia resultante de la relación n:m de usuario y libro.
* **Campos:**
  + **ID\_FK\_usuario:** Es la clave tanto primaria como foránea relacionada con la tabla usuario.
  + **ID\_FK\_libro:** Es la clave tanto primaria como foránea relacionada con la tabla libro.
  + **Fecha\_registro:** Fecha en la que se creó el registro

### Biblio\_libro

* **Resumen:** Es la tabla intermedia resultado de la relación n:m entre las tablas libro y biblioteca.
* **Campos:**
  + **ID\_FK\_biblioteca:** Es la clave tanto primaria como foránea relacionada con la tabla biblioteca.
  + **ID\_FK\_libro:** Es la clave tanto primaria como foránea relacionada con la tabla libro.
  + **Fecha\_compra:** Es la fecha en la que se compró el libro.
  + **Precio:** Precio al que se compró el libro.

### Saga

* **Resumen:** Es la tabla que almacena información de las distintas sagas de libros
* **Campos:**
  + **ID\_saga:** Es la clave principal, es incremental.
  + **Saga:** Es el nombre de la saga a la que pertenece el libro.
* **Relaciones:**
  + Relación 0:n con libro ya que un libro puede o no pertenecer a una saga pero una saga tiene n libros.

### Idioma

* **Resumen:** Es la tabla que contiene los distintos idiomas en los que puede estar un libro.
* **Campos:**
  + **ID\_idioma:** Es la clave primaria, es incremental.
  + **Idioma:** El nombre del idioma.
* **Relaciones:**
  + Relación n:m con libro ya que un libro puede estar en n idiomas y un idioma puede estar en n libros.

### Genero

* **Resumen:** Es la tabla que contiene los distintos géneros que puede tener un libro.
* **Campos:**
  + **ID\_genero:** Es la clave primaria, es incremental.
  + **Genero:** Es el genero al que puede pertenecer el libro.
* **Relaciones:**
  + Relación n:m con genero ya que un libro puede tener n géneros y un género puede estar en n libros.

### Autor

* **Resumen:** Esta tabla contiene información básica del autor, ya que son las editoriales quienes tratan con ellos.
* **Campso:**
  + **ID\_autor:** Es la clave primaria, es incremental.
  + **Nombres:** Es el nombre del autor.
  + **Apellidos:** Son los apellidos del autor.
* **Relaciones:**
  + Relación n:m con idioma ya que un libro puede tener n autores y un autor puede tener en n libros.

### Libro\_idioma

* **Resumen:** Es la tabla intermedia creada a partir de la relación n:m entre la tabla libro y la tabla idioma.
* **Campos:**
  + **FK\_ID\_libro:** Es la clave tanto primaria como foránea relacionada con la tabla libro.
  + **FK\_ID\_idioma:** Es la clave tanto primaria como foránea relacionada con la tabla idioma.

### Libro\_autor

* **Resumen:** Es la tabla intermedia creada a partir de la relación n:m entre la tabla libro y la tabla autor.
* **Campos:**
  + **FK\_ID\_libro:** Es la clave tanto primaria como foránea relacionada con la tabla libro.
* **FK\_ID\_autor:** Es la clave tanto primaria como foránea relacionada con la tabla autor.

### Libro\_genero

* **Resumen:** Es la tabla intermedia creada a partir de la relación n:m entre la tabla libro y la tabla género.
* **Campso:**
  + **FK\_ID\_libro:** Es la clave tanto primaria como foránea relacionada con la tabla libro.
  + **FK\_ID\_genero:** Es la clave tanto primaria como foránea relacionada con la tabla género.

### Editorial

* **Resumen:** Es la tabla que contiene información referente a las editoriales con las que está asociada la aplicación. Desde dentro de la aplicación no se puede crear un registro de esta tabla ni el usuario al que va asociado.
* **Campos:**
  + **ID\_editorial:** Es la clave primaria. En este caso es el NIF de la empresa.
  + **Nombre:** Nombre asociado a esa empresa.
  + **dirección:** Dirección física de la empresa.
  + **Correo:** Es el correo de la empresa.
  + **Teléfono:** Es el teléfono de la empresa.
  + **Contacto:** Es el nombre de persona de contacto. Puede ser NULL.
  + **Web:** Es la dirección de la web de la editorial en caso de que tengan. Puede ser NULL.
  + **ID\_FK\_usuario:** Es la clave foránea de la tabla usuario.
* **Relaciones:**
  + Relación n:m con la tabla libros ya que una editorial puede tener varios libros publicados y un libro puede ser publicado en varias editoriales a la vez.
  + Relación 0:1 con la tabla usuario ya que una empresa siempre tendrá un usuario en la aplicación, pero un usuario puede no tener asociado una editorial.

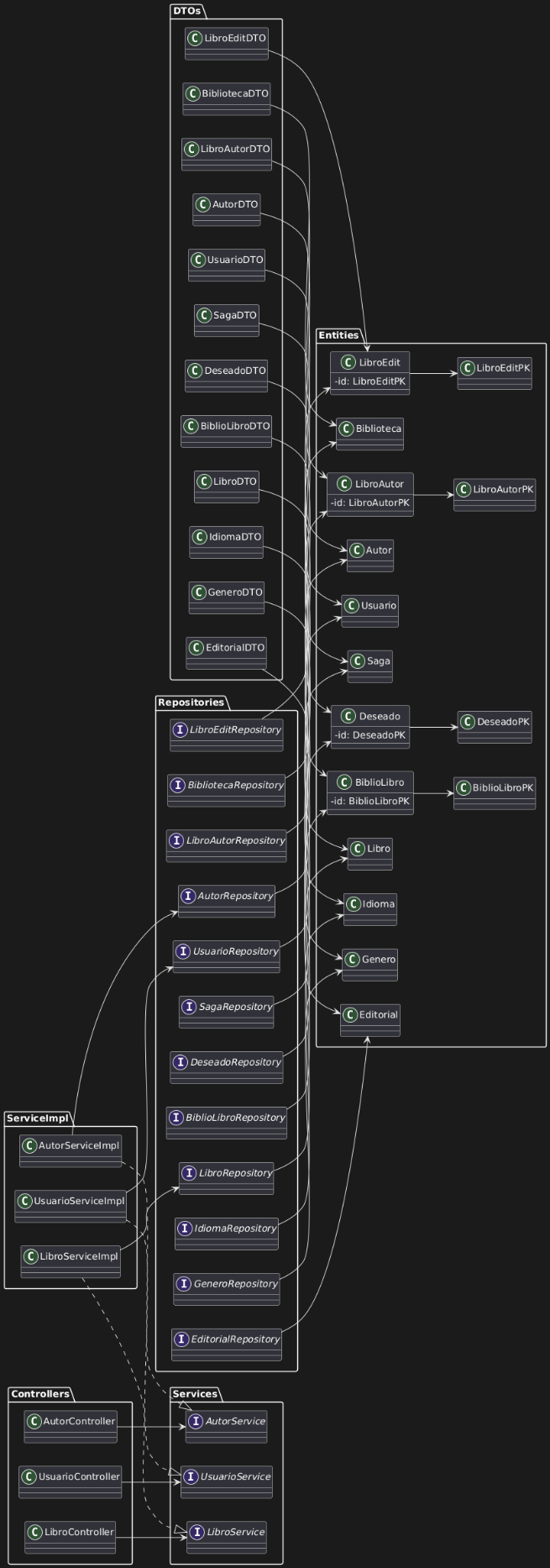
### Libro\_edit

* **Resumen:** Es la tabla intermedia creada a partir de la relación n:m de la tabla libro y la tabla editorial.
* **Campos:**
  + **ID\_FK\_editorial:** Es la clave tanto primaria como foránea relacionada con la tabla editorial.
  + **ID\_FK\_libro:** Es la clave tanto primaria como foránea relacionada con la tabla libro.
  + **Fecha\_alta:** Es la fecha en la que el libro fue publicado por esa editorial.

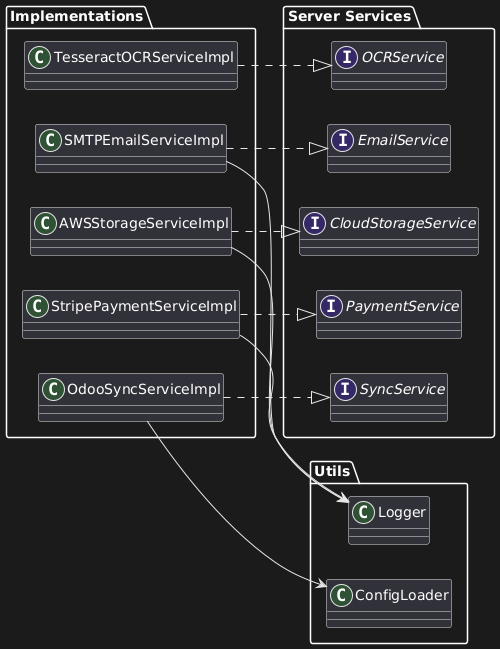
## Interfaz de usuario gráfica Descripción generada automáticamenteDiagrama E.R

## 6.4 Diagrama UML

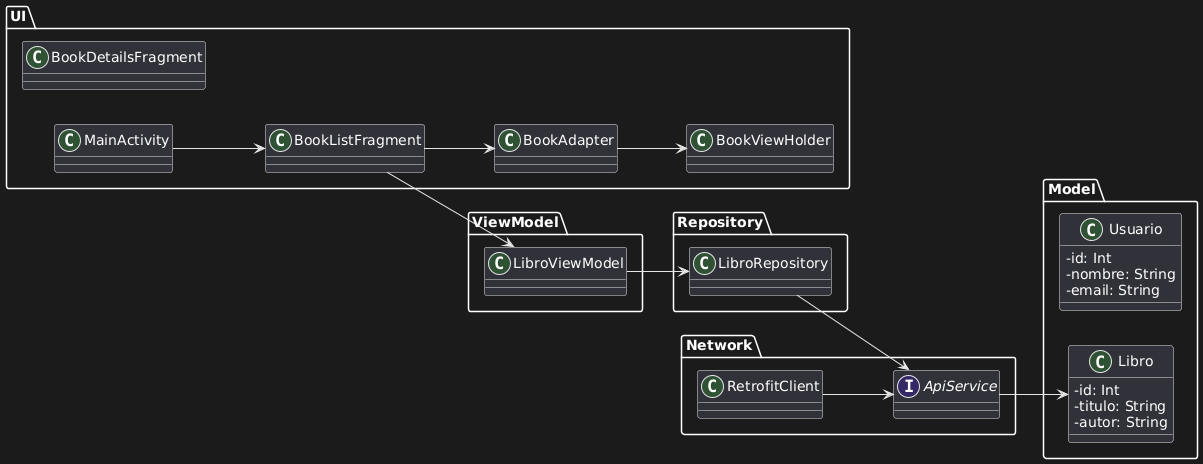
UML del API REST



UML del Servidor



UML de Android



# Diseño

Ya que la aplicación cuenta con dos versiones, una para escritorio y una móvil, cada aplicación contará con su respectiva interfaz gráfica. El objetivo es que sea lo más natural e intuitivo pasar de una versión a la otra, para ello se seguirá un diseño parecido en ambas, teniendo en cuenta las limitaciones del entorno con respecto a su contraparte. A su vez, también tendrán elementos comunes como puede ser el login, el registro, la recuperación de contraseña, y la pagina de ayuda, ya que esta última simplemente redirigirá a una url de que estará colgada en la página web de la aplicación. Para la interfaz gráfica de la versión de escritorio se utilizará java swing y para la versión móvil se utilizará kotlin. Como nota a tener en cuenta, los diseños que se mostraran son prototipos, por lo que no se verá reflejado cosas como colores, iconos, etc. También se contará con un servidor, pero no se hablará de el en esta sección.

## Elementos comunes

### Login, registro, recuperar contraseña

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

El login será lo primero que vea el usuario nada más abrir la aplicación en ambas versiones. Desde aquí se podrá navegar entre las distintas pestañas de registro y recuperar contraseña. Una vez el usuario haya insertado sus datos en la pestaña de login y suponiendo que todo sea correcto, pasara a la ventana principal.

Para reutilizar lo más posible el código, todo estará en un solo frame, y la navegación se hará con paneles. En el caso de la pestaña de recuperar contraseña, lo único visible al principio serán los campos de usuario, correo, código de verificación, y los botones de enviar código y confirmar código. En el momento que el usuario le dé al botón de confirmar código, se hagan todas las comprobaciones pertinentes y estas resulten correctas, revelara el botón de cambiar contraseña.

## UI de la aplicación de escritorio

Debido a que la versión de escritorio está pensada también para ser usada por las editoriales, lo que vea el usuario nada más entrar cambiará entre la ventana de usuario y la de editorial. Este hecho de esta forma para que un usuario promedio no pueda acceder a la ventana de subir libros y ya que a una editorial no le interesa las funciones del usuario promedio, esta es redirigida a la ventana de editorial directamente. En principio no habrá navegación entre las dos y las editoriales tampoco podrán acceder a la tienda.

### Vista usuario

Diagrama

Descripción generada automáticamente

En la imagen se puede ver el flujo de las ventanas que de un usuario. La primera ventana que vera siempre nada mas logearse será la de biblioteca, desde donde podrá acceder al resto de ventanas. Se ha diseñado pensando en utilizar un frame principal e ir cambiando entre la ventana de biblioteca y la de tienda, que serán paneles. Ambas ventanas tendrán una barra de herramientas, desde donde podrán acceder a las opciones de sus respectivas ventanas.

La ventana de biblioteca estará dividida en tres secciones, filtros de búsqueda, tabla de contenido e información básica. En la sección de filtros se le permitirá al usuario buscar por género, autor, saga, y titulo. En la tabla de contenido se mostrarán todos los libros de los que dispone el usuario, así como los que tenga en la biblioteca local actualmente. El contenido de esta tabla se verá afectado por los filtros que se le apliquen y se podrá escoger que campos del libro se mostrarán. En el panel de información se mostrará el autor, la saga, el género, la portada y la sinopsis del libro seleccionado.

La ventana de tienda estará distribuida de forma similar a la ventana de biblioteca, contando esta también con tres secciones, filtros de búsqueda, tabla de contenido e información básica. La sección de búsqueda cuenta con los mismos filtros que la biblioteca, con el añadido de poder buscar por fecha de publicación, ISBN, Precio y editorial. La tabla de contenido también será prácticamente idéntica, excepto por el añadido de que se mostrará el precio y si hay un descuento. Por último, en el panel de información se mostrará la portada del libro seleccionado y la sinopsis. También contara con el botón de añadir al carrito.

En la ventana de perfil se podrá cambiar directamente los datos como el nombre, los apellidos, la contraseña y el correo, aunque estos últimos abrirán un cuadro de dialogo para hacer comprobaciones antes de llevarse a cabo. También se podrá cerrar la sesión actual, abrir la ventana para añadir fondos que pedirá la información de la tarjeta a la que cobrar, y eliminar la cuenta por completo.

Interfaz de usuario gráfica

Descripción generada automáticamente

En la barra herramientas se podrá acceder a distintas opciones, como la navegación por la aplicación, añadir o eliminar un libro, crear una biblioteca o eliminarla, etc. Exceptuando tienda, biblioteca, perfil, ayuda y descargar libro abrirán su respectivo pop-up como aparece en la imagen. El botón de ayuda abrirá en el navegador la sección de ayuda de página web. El botón de “descargar libro” descargará todos los libros comprados que no tengamos ya descargados en la biblioteca local.

### Vista Editorial

Captura de pantalla de un celular

Descripción generada automáticamente

Ya que la editorial no es realmente un usuario que vaya a consumir la aplicación para otra cosa que no sean negocios, se le proporcionara una vista diferente a la del usuario común, dándole acceso a la capacidad de añadir libros con un formulario simple, ver una métrica rápida de cuanto ha generado cada libro, y ver los libros publicados. Como se ha mencionado antes, las editoriales solo tendrán acceso a esta vista.

A diferencia de la vista de usuario, la vista de editorial está pensada para ser lo más concisa y rápida de usar ya que es lo que más les suele interesar a las empresas. En el apartado de las métricas, las editoriales podrán acceder dinero total ganado con la venta de libros, cuanto han ganado por libro, cuanto han ganado por mes, etc.

## UI de la aplicación móvil

Para la versión móvil se utilizarán fragments y framelayout, ya que nos permitirá crear una interfaz bastante mas flexible a la hora de hacer la navegación entre pantallas y nos permitirá aprovechar gran parte del código que tengamos.

Diagrama

Descripción generada automáticamente

Al igual que con la versión de escritorio, lo primero que vera el usuario nada mas logearse será su biblioteca, desde donde tendrá acceso a la tienda, el perfil, y el OCR. Tanto la tienda como la biblioteca serán fragments y se crearán en la misma activity, teniendo acceso ambos a la barra de navegación. Los botones de OCR como filtros abrirán una ventana modal que llevará a cabo sus respectivas tareas.

Desde el fragment de biblioteca, el usuario verá todos los libros que tiene comprados en la tienda. Tendrá una vista previa con la portada, el título, el autor y el género. Cuando pulse en esa vista previa, esta se expandirá, mostrando la información del libro en más detalle y el botón de descarga. También se dispondrá de un buscador, en el que se podrá escribir el título del libro deseado. Si se ha activado algún filtro en su ventana correspondiente también se verá reflejado en la lista de libros.

El fragment de tienda reutiliza la idea de la biblioteca y contara con la misma lista con información básica del libro que se expandirá en el momento que el usuario pulse sobre él. También contara con un buscador por título y la ventana de filtros. Una vez escogido el libro deseado se procederá a la compra.

En la ventana de perfil, al igual que en la versión de escritorio, se podrá cambiar la información de la cuenta, añadir fondos y eliminarla, abriendo cada uno su respectiva ventana.

# Casos de uso

La aplicación estará enfocada en dos tipos de usuarios distintos, las empresas o editoriales y los usuarios comunes, siendo estos últimos los usuarios a los que está enfocada principalmente y que más uso le darán. Ya que el proyecto consta de dos aplicaciones, se procederá a explicar que usuario tiene acceso a que y desde donde.

* **Usuario común:** Este usuario tendrá acceso tanto a la aplicación de escritorio como a la aplicación móvil.
  + **Escritorio:** Desde esta aplicación podrá administrar su cuenta, administrar tanto su biblioteca local como la virtual, podrá comprar libros y leer sus libros.
  + **Movil:** En esta versión podrá administrar su cuenta, comprar libros, descargar los libros comprados, escanear imágenes de portadas para saber si el libro se encuentra en la tienda.
* **Editoriales:** Las editoriales solo cuenta con la versión de escritorio de la aplicación. Desde ella podrán subir a la tienda, eliminarlos, administrar sus precios, y consultar estadísticas de ventas.

# Diagrama de clases

A continuación, se mostrará los diagramas UML de las versiones de escritorio, de móvil, y el servidor.

### Versión de escritorio

Pantalla de computadora

Descripción generada automáticamente con confianza media

### Login

Esta clase será la encargada de comprobar las credenciales del usuario cuando intente iniciar sesión, también será la encargada del registro cuando este no tenga cuenta. Por último, contará con métodos para confirmar que el correo que se mete es un correo valido y que la contraseña tenga unos caracteres minimos, una mayuscula, numero, etc. También se encargará de autocompletar el campo de usuario si este lo desea, a la hora de iniciar sesión a través del archivo .conf.

### MandaCorreo

Esta clase será la encargada de gestionar el envío de correos electrónicos desde la aplicación. Se utilizará para funcionalidades como la recuperación de cuenta, cambio de correo electrónico, envío de códigos de verificación y otras comunicaciones importantes.

### ManejaImagen

Sera la clase encargada de gestionar las imágenes que utilice la aplicación. Mas específicamente, se encargará de redimensionar imágenes y cambiar el formato según sea necesario.

### ManejaBibliotecaLocal

Esta clase se ocupará de la gestión local de las bibliotecas del usuario. Permitirá crear nuevas bibliotecas, eliminar las existentes y cambiar entre ellas. Las rutas y configuraciones se almacenarán en un archivo de configuración, donde también se indicará cuál de ellas es la biblioteca predeterminada. Además, esta clase se encargará de vincular cada libro con su respectiva portada, asegurando así una correcta presentación visual en la interfaz.

### ManejaBibliotecaLibro

Clase centrada en la gestión del contenido de las bibliotecas de libros electrónicos. Utilizará la librería epulib para acceder al contenido de los archivos EPUB y extraer información relevante como título, autor, número de páginas, metadatos o la portada, de los libros que agregue el usuario. Permitirá gestionar su lectura, y mostrar sus datos de manera estructurada.

### Conexión

Clase responsable de establecer y gestionar la comunicación entre la aplicación de escritorio y el servidor. Esta clase será la encargada de consumir la API que conecta con la base de datos, enviando y recibiendo datos de forma segura. Implementará métodos para realizar operaciones como login, registro, sincronización de bibliotecas, descarga de libros o cualquier otra interacción entre cliente y servidor.

# Planificación

La planificación es clave para llevar el proyecto a buen puerto, ya que nos permite organizar las tareas y tener un camino claro hacia los objetivos. Al dividir el trabajo en fases, con un enfoque flexible que permita trabajar en paralelo en algunos casos, buscamos optimizar el tiempo y la colaboración.

## Diagrama de Gantt

En este cronograma se agrupan las actividades principales del proyecto, estableciendo su secuencia y considerando tareas que podemos ejecutar simultáneamente. Cada fase es importante, y la idea es avanzar de manera iterativa para adaptarnos a cualquier cambio.

**Fases del proyecto:**

1. **Montaje y prueba de la base de datos en entorno local**: Asegurarnos de que la base de datos funcione correctamente en un entorno controlado antes de pasar a la siguiente fase.
2. **Desarrollo de la API RESTful con Spring Boot y PostgreSQL**: Aquí empezamos con el backend, utilizando tecnologías que nos permitan manejar solicitudes y respuestas de forma eficiente.
3. **División de tareas**:
   * Un integrante se encargará del **servidor** (backend).
   * Otro se centrará en el **desarrollo de la aplicación de escritorio**.
4. **Implementación del OCR con Tesseract y la pasarela de pagos Stripe**: Incluir el reconocimiento de texto y las opciones de pago en el sistema.
5. **Desarrollo de la aplicación móvil con Kotlin para Android**: Desarrollar la versión móvil, donde Kotlin será clave para la interfaz y el comportamiento de la app.
6. **Integración con Odoo para la gestión empresarial**: Conectar nuestro sistema con Odoo para optimizar procesos internos.
7. **Migración y despliegue en un entorno cloud (Render, AWS, etc.)**: Llevar todo a la nube para un mejor rendimiento y escalabilidad.
8. **Pruebas de calidad, documentación final y despliegue**: Finalmente, realizar pruebas, documentar el trabajo y poner todo en producción.

## Definición de Recursos y Logística

Para cada fase, hemos identificado los recursos necesarios, asegurándonos de que todo esté listo para llevar a cabo el desarrollo de manera efectiva.

**Recursos materiales:**

* **Ordenadores personales**: Para trabajar en el desarrollo de backend y frontend. Con un entorno adecuado de desarrollo Java (NetBeans o IntelliJ).
* **Dispositivos móviles Android**: Para realizar pruebas durante el desarrollo de la app móvil.
* **Conexión a internet**: Esencial para pruebas, descargas de dependencias y despliegues.
* **Hosting en la nube (Render, AWS)**: Para la fase final de despliegue en producción.

**Recursos tecnológicos:**

* **Spring Boot, PostgreSQL y Lombok**: Para construir la API de backend.
* **Tesseract OCR**: Para la funcionalidad de reconocimiento de texto.
* **Stripe API**: Para las transacciones de pago seguras.
* **Java Swing/JavaFX**: Para el desarrollo de la aplicación de escritorio.
* **Kotlin + Android Studio**: Para la aplicación móvil en Android.
* **Odoo**: Para la integración con el sistema de gestión empresarial.
* **Servicios cloud (Render, AWS)**: Para el despliegue final.

**Recursos humanos:**

* **Desarrollo colaborativo**: Un integrante se dedicará al backend, mientras que el otro se encargará de las interfaces (escritorio y móvil).
* Ambos trabajarán juntos en pruebas, despliegue y documentación final.

**Logística:**

* **Control de versiones (GitHub/GitLab)**: Para mantener el código sincronizado.
* **Reuniones periódicas**: Para revisar el progreso, identificar obstáculos y ajustar el plan.
* **Herramientas colaborativas (Trello, Notion)**: Para gestionar las tareas y asegurarnos de que todo esté en su lugar.

**Documentación centralizada**: Para evitar errores de integración y asegurar que todo el equipo esté alineado.

# Bibliográfica

<https://github.com/stripe/stripe-java> - API para manejar pagos

<https://github.com/tesseract-ocr/tessdoc> - OCR usado para el reconocimiento de imágenes

<https://github.com/nguyenq/tess4j> - Wrapper para tesseract

<https://github.com/psiegman/epublib> - Libreria para manejar ebooks