Aplicación multiplataforma de compra y gestión de libros electrónicos

Andrés García Payá y Alexis Willy Andía Rosales

Contenido

[1. Resumen 2](#_Toc195210067)

[2. Introducción 2](#_Toc195210068)

[3. Objetivos 3](#_Toc195210069)

[4. Palabras clave 3](#_Toc195210070)

[5. Análisis de contexto 4](#_Toc195210071)

[5.1. Análisis del Contexto 4](#_Toc195210072)

[5.2. Innovación 5](#_Toc195210073)

[6. Modelo de datos 6](#_Toc195210074)

[6.1. Resumen 6](#_Toc195210075)

[6.2. Tablas 6](#_Toc195210076)

[6.2.1. Usuario 6](#_Toc195210077)

[6.2.2. Info\_usuario 7](#_Toc195210078)

[6.2.3. Biblioteca 7](#_Toc195210079)

[6.2.4. Libro 7](#_Toc195210080)

[6.2.5. Deseo 8](#_Toc195210081)

[6.2.6. Biblio\_libro 8](#_Toc195210082)

[6.2.7. Saga 8](#_Toc195210083)

[6.2.8. Idioma 8](#_Toc195210084)

[6.2.9. Genero 9](#_Toc195210085)

[6.2.10. Autor 9](#_Toc195210086)

[6.2.11. Libro\_idioma 9](#_Toc195210087)

[6.2.12. Libro\_autor 9](#_Toc195210088)

[6.2.13. Libro\_genero 9](#_Toc195210089)

[6.2.14. Editorial 10](#_Toc195210090)

[6.2.15. Libro\_edit 10](#_Toc195210091)

[6.3. Diagrama E.R 11](#_Toc195210092)

# 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 titulo de los libros, permitiendo buscar en la base de datos si ese libro esta 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 esta 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 catalogo 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

## 5.1. Análisis del Contexto

En los últimos años, el mundo de los libros electrónicos ha ganado mucha popularidad gracias a la facilidad que ofrecen para acceder, transportar y leer contenido desde prácticamente cualquier dispositivo. Sin embargo, este crecimiento no ha estado libre de problemas. Muchas plataformas conocidas, como Amazon Kindle o Google Books, aplican restricciones que afectan directamente al usuario, como el uso de DRM que limita el acceso a los libros solo desde ciertos dispositivos o cuentas. Incluso, en algunos casos, empresas como Amazon han eliminado la posibilidad de descargar libros comprados o los han modificado sin permiso del usuario.

Estas limitaciones han encendido la alarma entre muchos lectores digitales, que empiezan a cuestionarse si realmente "poseen" los libros que han comprado. Además, para aquellos que buscan una experiencia más libre o desean organizar sus libros como prefieran, las alternativas actuales son pocas o demasiado técnicas.

Inspirándonos en cómo otras industrias han resuelto problemas similares —como el mundo de los videojuegos en PC, donde existen plataformas como Steam o Epic Games que permiten comprar, gestionar y jugar desde una misma aplicación— nace este proyecto. Queremos trasladar ese mismo enfoque al mundo de los libros electrónicos: una única aplicación que no solo permita comprar libros, sino también gestionarlos, organizarlos y leerlos, todo desde un solo lugar y con total control por parte del usuario.

**Competencia**

Al analizar lo que ya existe en el mercado, identificamos varias plataformas populares:

Amazon Kindle: muy conocida, con un catálogo enorme, pero extremadamente cerrada y restrictiva. Los libros tienen DRM y están limitados a su propio ecosistema.

Google Play Books: más flexible, pero sin funciones avanzadas de organización local o acceso editorial.

Kobo: otra buena opción, pero también con restricciones técnicas y de formato.

Calibre: excelente para la gestión local de libros, sin tienda integrada, pero poco accesible para usuarios no técnicos.

Ninguna de ellas combina todas las funcionalidades que los usuarios modernos desean: compra, organización personalizada, gestión de libros externos, lectura, OCR, y además, herramientas para editoriales. Ese vacío es justamente el que nuestra aplicación busca llenar.

**DAFO del Proyecto**

Fortalezas:

Aplicación multiplataforma: misma experiencia desde PC y móvil.

Escaneo de portadas mediante OCR para encontrar libros fácilmente.

Conversión entre formatos de lectura (.pdf y .epub).

Posibilidad de añadir libros propios y organizarlos en bibliotecas locales.

Acceso exclusivo para editoriales para publicar directamente en la tienda.

Uso de tecnologías libres y sostenibles como PostgreSQL y Tesseract.

Debilidades:

Se requiere una inversión inicial para mantener servidores y servicios.

Al principio puede costar atraer usuarios y editoriales si ya usan otras plataformas.

Doble desarrollo y mantenimiento: versión móvil y versión de escritorio.

Oportunidades:

Muchos usuarios buscan alternativas más libres y completas.

Editoriales independientes necesitan espacios donde publicar sin intermediarios.

Hay un hueco en el mercado para una plataforma “todo en uno” para ebooks.

Amenazas:

Grandes empresas podrían adoptar ideas similares.

Obstáculos legales o cambios en normativas de protección de contenido.

Avances tecnológicos que exijan adaptación constante.

## 5.2. Innovación

Este proyecto nace con la intención de romper barreras y ofrecer a los lectores digitales una experiencia mucho más completa, intuitiva y libre. A diferencia de las plataformas tradicionales, esta aplicación no solo se centra en la venta de libros, sino que da poder al usuario para organizar, personalizar y conservar su contenido como prefiera.

Una de las funciones más innovadoras es el uso de OCR (Reconocimiento Óptico de Caracteres) integrado a través de Tesseract. Gracias a esta tecnología, los usuarios pueden escanear la portada de un libro o una nota donde esté escrito el título, y la aplicación buscará automáticamente si ese libro está disponible en la tienda. Esto convierte al móvil en una herramienta de descubrimiento literario práctica y potente.

Otra gran innovación es la posibilidad de convertir libros entre formatos .epub y .pdf, algo especialmente útil para quienes usan distintos dispositivos de lectura. Además, la app permite gestionar no solo los libros comprados, sino también aquellos que el usuario ya tenga en su ordenador o que haya adquirido por otras vías. Todo esto se podrá hacer desde una biblioteca personal organizada por el propio usuario, sin depender de ninguna nube externa o permisos especiales.

Y por supuesto, no nos olvidamos de las editoriales. La versión de escritorio incluye una sección especial para ellas, donde podrán subir sus libros, gestionar su catálogo y publicar directamente en la tienda. Esto es algo poco común en otras plataformas, y permite dar visibilidad a editoriales más pequeñas o independientes que buscan conectar con los lectores sin pasar por intermediarios.

En definitiva, nuestra aplicación apuesta por la libertad, el control y la centralización, ofreciendo una solución que no existe hoy en el mercado. Todo esto usando tecnologías abiertas, pensadas para durar y crecer con el tiempo.

# 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.
* **Relaciones:**
  + Relación 1:1 con info\_usuario ya que un usuario solo puede tener información de un perfil que es el suyo.
  + 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.

### Info\_usuario

* **Resumen:** Esta tabla contiene información que por lo que sea no se quiera gaurdar en la tabla de usuario como pueda ser el sal utilizado para hacer el salting. Tiene una relación 1:1 con la tabla usuario
* **Campos:**
  + **ID\_FK\_Usuario:** Es tanto la clave primaria de esta tabla como la clave foránea de usuario.
  + **Semilla:** Es el sal utilizado 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 1:1 con usuario ya que un usuario solo puede tener info de un perfil que es el suyo.

### 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 ultima 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 esta 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
  + **Titulo:** 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 numero de paginas que tiene el 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.

### 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 esta 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.

## Diagrama E.R

Interfaz de usuario gráfica

Descripción generada automáticamente

# 7. 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.

## 7.1. 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.

## 7.2. 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.

1. Bibliográfica

<https://github.com/stripe/stripe-java>

<https://github.com/tesseract-ocr/tessdoc>