

## INSTITUTO TECNOLÓGICO DE OAXACA

Desarrollar una **Réplica** de la **Aplicación** de **YouTube** para **Dispositivos Android**  
con un **Enfoque** de **Geolocalización** dentro del estado de **Oaxaca de Juárez**

### Integración De Procesos De Desarrollo De Software

Prof. Espinosa Pérez Jacob

Ángel Arturo Pérez García

<b>Introducción</b>	<b>3</b>
<b>Planteamiento del problema</b>	<b>3</b>
<b>Objetivo general</b>	<b>3</b>
<b>Objetivos específicos</b>	<b>4</b>
<b>Justificación</b>	<b>4</b>
<b>Metodología</b>	<b>5</b>
Diseño	5
Métodos para la Recolección de la Información	5
Modelo ER	5
Diagrama ER	6
Diagramas UML	6
Prototipos de Pantallas	6
Selección de las herramientas	6
Sistema de Control de Versiones	6
Sistema Gestor de Tareas	7
Sistema Operativo Móvil	8
Entorno de Desarrollo Integrado	8
Kit de Herramientas	8
Selección de la Metodología de Desarrollo de Software	9
<b>Proceso de Desarrollo</b>	<b>9</b>
Requerimientos del Proyecto	9
Historias de Usuario	9

## **Introducción**

Las plataformas de video en línea se han convertido en uno de los principales medios de comunicación, entretenimiento y difusión de información a nivel mundial. Sin embargo, la mayoría de estas plataformas priorizan algoritmos de recomendación basados en intereses personales, historial de reproducción y popularidad global, dejando de lado un enfoque geográfico que permita a los usuarios descubrir contenido relevante de su entorno inmediato.

El proyecto propone el desarrollo de una aplicación móvil similar a YouTube, pero con un enfoque en la geolocalización de los videos, de modo que los usuarios puedan descubrir y consumir contenido generado en su misma zona geográfica, fomentando así una conexión más cercana con su comunidad, eventos locales, negocios, cultura y noticias de interés regional.

## **Planteamiento del problema**

Las plataformas actuales de distribución de videos carecen de un mecanismo que permita filtrar o priorizar el contenido en función de la ubicación donde fue creado o publicado. Esto representa una limitante para:

- Usuarios locales, que tienen dificultades para encontrar información relevante de su zona a riesgo posible de perder su identidad cultural.
- Creadores de contenido regional, que pierden visibilidad frente a videos de alcance global, reduciendo su capacidad de llegar a su audiencia inmediata.
- Comunidades y negocios, que podrían beneficiarse de un medio que difunda actividades, noticias o productos de interés estrictamente local.

La falta de una plataforma que organice y muestre videos en función de su procedencia geográfica deja un vacío para quienes buscan una experiencia de descubrimiento más centralizada.

## **Objetivo general**

Desarrollar una réplica móvil del sistema de YouTube para dispositivos Android (oreo 8.0+), que permita visualizar, compartir y acceder a contenido audiovisual

utilizando la API de YouTube integrando un sistema de recomendación basado en la geolocalización, automática o manual, del usuario.

### **Objetivos específicos**

- Implementar un sistema de autenticación para acceder a la API de YouTube de forma segura.
- Diseñar una interfaz intuitiva que permita la navegación fluida entre videos y canales.
- Desarrollar un módulo de detección automático y manual para la geolocalización del usuario en tiempo real.
- Desarrollar un sistema de recomendación que filtre y sugiera videos populares o relevantes con base en la ubicación del usuario.
- Desarrollar un sistema de historial de búsqueda con filtros integrados.
- Realizar pruebas de usabilidad y optimización del sistema.

### **Justificación**

El consumo de información local es cada vez más relevante, por eso surge la necesidad de diseñar e implementar una aplicación tipo “YouTube geolocalizado”, que permita:

- Asociar cada video subido a un punto geográfico (usando metadatos de GPS o ubicación manual).
- Filtrar y recomendar contenido con base en la posición actual del usuario.
- Proveer herramientas de búsqueda por regiones específicas.

Esta solución no solo responderá a las necesidades de información local, sino que también fomentará la interacción entre creadores y audiencias de un mismo lugar, impulsando la identidad y cultura de las comunidades.

## Metodología

### Diseño

#### *Métodos para la Recolección de la Información*

Los instrumentos de recolección de datos estarán diseñados para captar los requerimientos generales y específicos del cliente para realizar la réplica del sistema de YouTube, junto con el módulo de geolocalización para visualizar los videos generados aledañamente a la ubicación.

- Guías de uso de la API de YouTube (Data API): Se hará una revisión que ofrece YouTube junto con su API, de esta forma se podrá seleccionar la mejor ruta para la implementación de la localización como funcionalidad para la recomendación de videos locales.
- Entrevistas: Se realizarán entrevistas para comprender los requisitos y necesidades del cliente para con la aplicación móvil.

#### *Modelo ER*

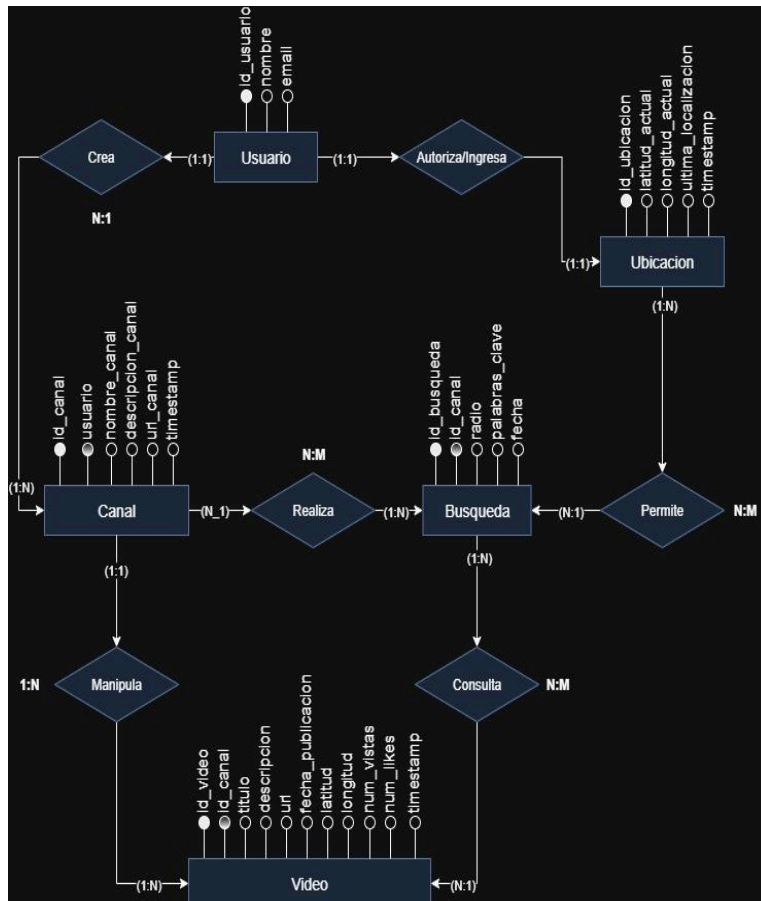
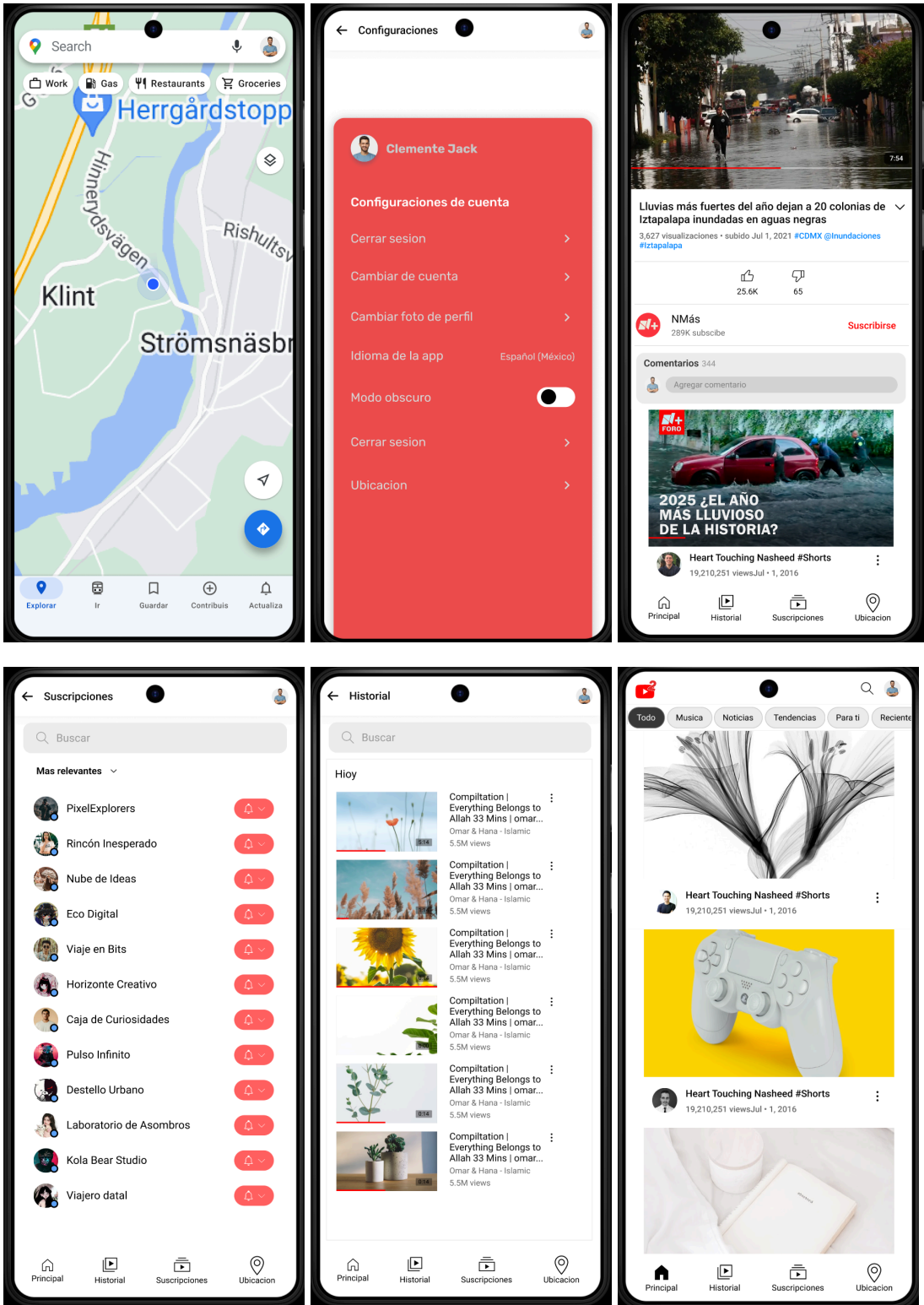
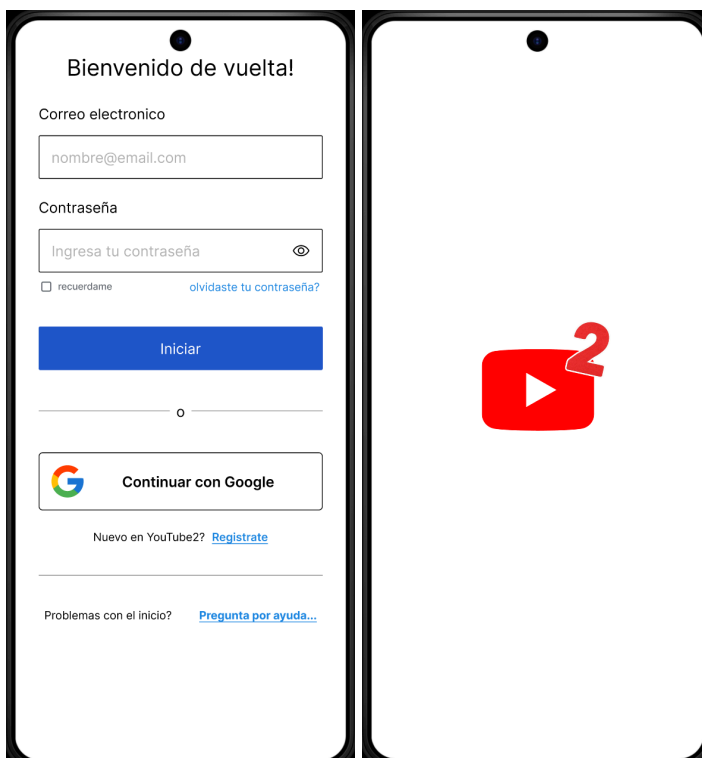
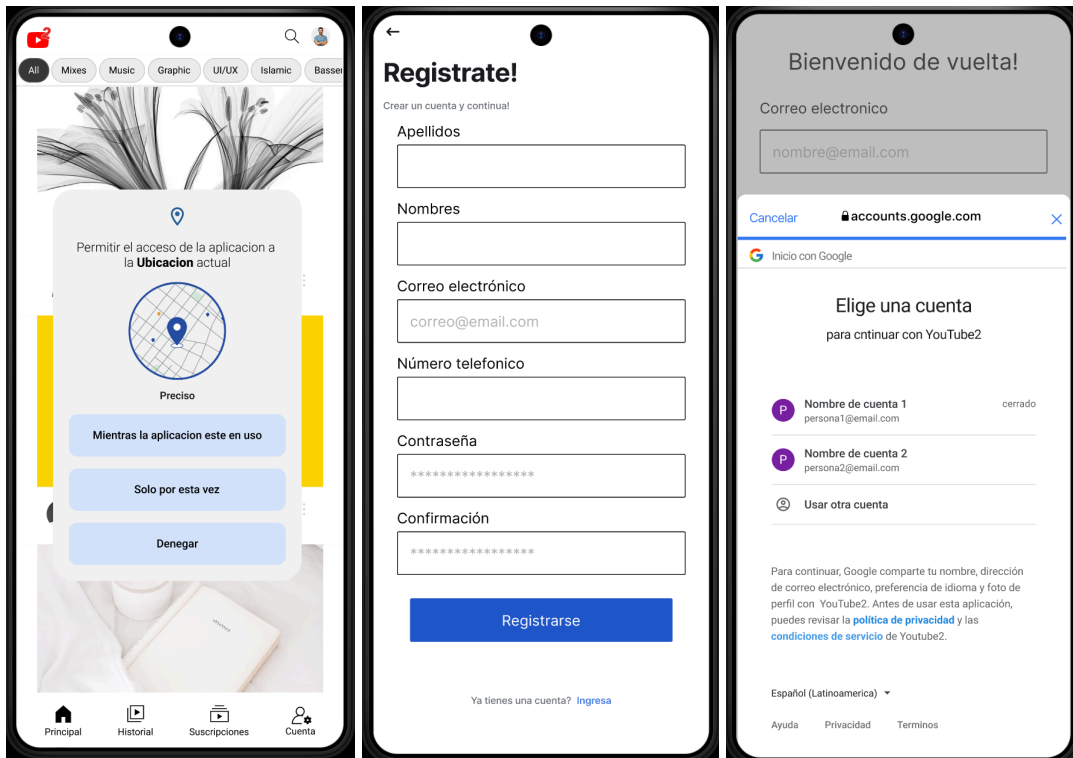


Diagrama ER

Diagramas UML

Prototipos de Pantallas





## Selección de las herramientas

Android Studio

Flutter

Expo

Node.js

cloudflare

Visual Studio

Postman

GitHub

Jira

### ***Sistema de Control de Versiones***

Se eligió GitHub como gestor de versiones, dado que proporciona una plataforma completa sobre Git, integración con el gestor de tareas que se eligió, repositorios públicos/privados sin complicaciones y una interfaz amigable y ampliamente documentada, ideal para proyectos que requieren un desarrollo rápido.

Características	Git	GitHub	GitLab
Tipo	Sistema de control de versiones distribuido (DVCS)	Plataforma de hospedaje para repositorios Git	Plataforma de hospedaje y CI/CD para repos Git
Instalación	Local en cualquier sistema	Basado en la nube (GitHub.com) o Enterprise Server	Nube (gitlab.com) o instalación on-premise (self-hosted)
Licencia	Software libre (GPL v2)	Freemium	Community Edition: MIT / Enterprise: comercial
Control de versiones	Sí, completo	Basado en Git	Basado en Git
Uso	Control de versiones en proyectos locales o en cualquier servidor	Open source, proyectos colaborativos públicos y privados	Empresas que requieren self-hosting o CI/CD integrado
Popularidad	Base de todo el	Muy alta	Alta en empresas



	ecosistema		
Costo	Gratis	Plan gratuito y planes de pago	Edición gratuita y de pago

### **Sistema Gestor de Tareas**

Se eligió el sistema gestor de tareas Jira dado que cuenta con un plan gratuito sin restricciones relevantes en cuando a condiciones de trabajo, pero si en número de integrantes en la participación del proyecto, más esto, sin embargo, no es relevante para la ocasión, además de contar con una metodología ágil como modelo principal para el desarrollo de proyectos.

Producto	Tipo de licencia	Metodologías	Tipo de plataforma/herramienta	Control de Versiones	Clientes	Max. de participantes
Jira	Modelo comercial basado en suscripción / modelo gratuito limitado	Scrum, Kanban	Gestión de proyectos y seguimiento de incidencias	Historial de cambios, historial de issues, versionado de productos	Fedora Commons, Hibernate, Apache Software	Estandar y Plan gratuito (10)
Monday.com	Modelo comercial basado en suscripción	Gantt, Kanban, Cronograma, etc	Gestión de trabajo automatizado	Historial de actividad y versiones de documentos	Herdez, Coca Cola, Coppel, Uber	Ilimitados
ClicUp	Modelo comercial basado en suscripción	Kanban, Scrum, etc	Gestión de trabajo (all in one)	Historial de actividad, version de tareas	Logitech, AT&T, Paramount, Sephora	Ilimitados

## ***Sistema Operativo Móvil***

El sistema operativo seleccionado para el desarrollo de la aplicación es **Android**, el cual destaca por ser de código abierto y contar con la mayor cuota de mercado a nivel mundial en dispositivos móviles. Además de ser el sistema operativo del móvil del desarrollador, mismo que se utilizara para realizar todas las pruebas del proyecto.

Razones de la elección:

- Amplia disponibilidad de dispositivos: Android está presente en una gran variedad de marcas y gamas de precio, lo que permite que la aplicación llegue a un público más amplio.
- Ecosistema abierto: al ser un proyecto de código abierto (basado en el kernel de Linux), ofrece mayor flexibilidad para personalización y despliegue.

## ***Entorno de Desarrollo Integrado***

Para el desarrollo se utiliza Android Studio, el IDE oficial para Android mantenido por Google.

Razones de la elección:

- Soporte oficial y actualizaciones constantes: garantiza compatibilidad con las últimas versiones de Android.
- Herramientas integradas: emuladores de dispositivos, depurador en tiempo real, editor de interfaces y asistentes para empaquetado de APK.
- Compatibilidad multiplataforma: funciona en Windows, macOS y Linux, facilitando el trabajo en distintos entornos.
- Integración con Flutter y Kotlin: facilita tanto el desarrollo nativo como multiplataforma, con configuraciones preestablecidas y plantillas.

## ***Kit de Herramientas***

El kit de herramientas elegido es Flutter, un framework de desarrollo de aplicaciones multiplataforma creado por Google.

Razones de la elección:

- Código único para varias plataformas: permite generar aplicaciones para Android, iOS, Web, optimizando tiempo y recursos.
- Rendimiento nativo: ofrece interfaces fluidas y de alta calidad.
- Gran comunidad y soporte oficial: cuenta con abundante documentación, paquetes y librerías de terceros para el desarrollo.
- Integración con Android Studio: permite desarrollar y depurar desde un mismo entorno sin configuraciones complejas.

### **Selección de la Metodología de Desarrollo de Software**

Para la gestión del proceso de desarrollo se adoptó la metodología ágil SCRUM.

Razones de la elección:

- Adaptabilidad a cambios
- Trabajo incremental
- Control de calidad continuo
- Principal metodología ágil utilizada dentro del gestor de tareas elegido

### **Proceso de Desarrollo**

#### **Requerimientos del Proyecto**

Funcionales	No Funcionales
Los usuarios podrán iniciar sesión de manera segura.	El sistema debe filtrar videos en función de la ubicación actual o seleccionada por el usuario.
El usuario podrá inicializar la aplicación logueándose por medio de las cuentas de Google.	El sistema contará con la integración de multiples API'S, como son Google Maps, Google Auth, YouTube, Google Translate, etc.
El sistema debe permitir configurar manualmente una ubicación alternativa.	El sistema debe conectarse a la API de YouTube mediante autenticación
El usuario podrá visualizar los videos desde la aplicación.	El sistema debe priorizar la visualización de videos cercanos al usuario.
El usuario podrá visualizar los metadatos básicos de los videos.	El sistema debe detectar automáticamente la ubicación del usuario en tiempo real.

El sistema debe mostrar una lista de videos disponibles desde la API de YouTube.	El sistema debe mostrar un historial de videos vistos.
El usuario podrá navegar por la aplicación principalmente mediante un componente navigator.	El sistema debe registrar búsquedas y reproducciones de cada usuario.
El usuario podrá cambiar los modos de la aplicación, oscuro o claro, y el lenguaje del mismo.	El sistema debe contar con una interfaz intuitiva y responsiva.
El usuario podrá ver diferida sus historiales en función a la ubicación en la que se produjeron estas	El sistema contará con dos sistemas de historiales, local y cambiante.

### ***Historias de Usuario***