

UNIVERSIDAD AUTÓNOMA GABRIEL RENÉ MORENO

FACULTAD DE INGENIERÍA EN CIENCIAS DE LA COMPUTACIÓN Y TELECOMUNICACIONES



INVESTIGACIÓN 1

MATERIA: Tópicos Avanzados de Programación

ESTUDIANTE : Dariana Soliz Felias

SANTA CRUZ

Contenido

1. INVESTIGACIÓN DE APIs	4
Facebook API (Meta).....	4
Instagram API (parte de Meta)	4
LinkedIn API	5
TikTok API.....	6
WhatsApp API (WhatsApp Business API)	7
2.TABLA DE CARACTERISTICAS.....	7
3. SELECCIÓN DE LLM	9
4. PROPUESTA DE ARQUITECTURA	13

1. INVESTIGACIÓN DE APIs

Facebook API (Meta)

Permisos Requeridos:

1. **user_posts**: Permite leer las publicaciones de un usuario en su perfil o en las páginas que administra.
 - o **Uso**: Acceder a las publicaciones realizadas por el usuario.
 - o **Restricciones**: Sólo funciona para cuentas de usuario con permisos específicos. Es necesario que el usuario otorgue este permiso durante el proceso de autenticación.
2. **pages_manage_posts**: Permite crear, editar y borrar publicaciones en una página de Facebook.
 - o **Uso**: Gestionar contenido publicado en las páginas que un usuario administra.
 - o **Restricciones**: Solo funciona para páginas que el usuario tiene permisos de administrador.
3. **public_profile**: Permite acceder a la información básica pública del perfil de un usuario (nombre, foto de perfil, etc.).
 - o **Uso**: Obtener información pública del usuario.
 - o **Restricciones**: Este es un permiso básico y debe ser concedido al autenticar al usuario, pero es limitado a la información accesible públicamente.
4. **email**: Permite acceder al correo electrónico del usuario.
 - o **Uso**: Obtener la dirección de correo electrónico del usuario para propósitos de contacto o autenticación.
 - o **Restricciones**: No todos los usuarios proporcionan esta información.

Requisitos:

- **Registrar aplicación** en el Facebook Developer Portal.
- **OAuth 2.0** para autenticación: Necesitarás implementar un flujo de autenticación de OAuth 2.0 para obtener un token de acceso.
- **Token de acceso**: Necesitas un token de acceso a la API, que puede ser de corta o larga duración.

Limitaciones:

- **Límites de tasa**: Facebook establece límites de tasa para las API, lo que puede restringir la cantidad de solicitudes que puedes hacer.
- **Revisión de permisos**: Algunos permisos, como user_posts, requieren que la aplicación sea revisada y aprobada por Facebook antes de que puedas usarlos en producción.

Instagram API (parte de Meta)

Permisos Requeridos:

1. **instagram_basic**: Permite acceder a la información básica del perfil de Instagram del usuario, como su nombre, foto de perfil y biografía.
 - o **Uso**: Obtener información básica del perfil de Instagram del usuario.
 - o **Restricciones**: Este es el permiso más básico y es necesario para interactuar con la API de Instagram.
2. **user_media**: Permite acceder a las publicaciones del usuario, incluidas fotos y videos.
 - o **Uso**: Obtener las publicaciones del usuario en su cuenta de Instagram.
 - o **Restricciones**: El usuario debe dar su consentimiento explícito para acceder a sus medios.
3. **pages_show_list**: Permite obtener una lista de las páginas de Facebook asociadas al perfil de Instagram.
 - o **Uso**: Para usuarios que vinculan su cuenta de Instagram a una página de Facebook, se puede acceder a esta lista.
 - o **Restricciones**: Este permiso es útil solo para interacciones entre Facebook e Instagram.

Requisitos:

- **Registrar aplicación** en el Facebook Developer Portal.
- **OAuth 2.0** para autenticación: Similar a la API de Facebook, se usa OAuth 2.0 para autenticar a los usuarios y obtener un token de acceso.
- **Token de acceso**: Se necesita un token de acceso a la API para interactuar con los datos de Instagram.

Limitaciones:

- **Límites de uso**: Los límites de uso dependen de los permisos y la cantidad de interacciones con las publicaciones. Instagram también impone límites a la cantidad de publicaciones que puedes acceder por solicitud.
- **Revisión de permisos**: Algunas funcionalidades avanzadas requieren la revisión de la aplicación por parte de Instagram.

LinkedIn API

Permisos Requeridos:

1. **r_liteprofile**: Permite acceder a la información básica del perfil del usuario, como nombre, foto, título profesional y ubicación.
 - o **Uso**: Obtener la información básica de un perfil de LinkedIn.
 - o **Restricciones**: Este es el permiso mínimo requerido para interactuar con la API de LinkedIn.

2. **r_emailaddress**: Permite acceder a la dirección de correo electrónico del usuario.
 - **Uso**: Obtener la dirección de correo electrónico para contactar al usuario.
 - **Restricciones**: No todos los usuarios compartirán su dirección de correo electrónico.
3. **w_member_social**: Permite publicar contenido en nombre del usuario (crear publicaciones).
 - **Uso**: Permite interactuar en el feed del usuario, creando publicaciones en LinkedIn.
 - **Restricciones**: Este permiso solo se puede usar para publicar contenido en el perfil del usuario.

Requisitos:

- **Registrar aplicación** en el LinkedIn Developer Portal.
- **OAuth 2.0** para autenticación: Debes implementar OAuth 2.0 para autenticar usuarios.
- **Token de acceso**: Necesitarás un token de acceso para acceder a los datos del usuario.

Limitaciones:

- **Límites de uso**: LinkedIn tiene límites de tasa para sus API, lo que puede restringir la cantidad de llamadas a la API.
- **Revisión de permisos**: Algunos permisos requieren la aprobación de LinkedIn.

TikTok API

Permisos Requeridos:

1. **user.info.basic**: Permite obtener la información básica del perfil del usuario, como su nombre de usuario y su foto de perfil.
 - **Uso**: Obtener los detalles básicos de un perfil de TikTok.
 - **Restricciones**: Requiere la autenticación del usuario.
2. **video.list**: Permite acceder a la lista de videos del usuario.
 - **Uso**: Obtener los videos del usuario para análisis o interactuar con ellos.
 - **Restricciones**: Solo se pueden acceder a los videos del usuario si se ha dado permiso.
3. **video.upload**: Permite cargar un video a la cuenta del usuario.
 - **Uso**: Subir contenido de video a TikTok.
 - **Restricciones**: Los videos deben cumplir con las políticas de contenido de TikTok.

Requisitos:

- **Registrar aplicación** en el TikTok Developer Portal.
- **OAuth 2.0** para autenticación: Implementar OAuth 2.0 para autenticar a los usuarios y obtener su token de acceso.
-

Limitaciones:

- **Límites de uso:** TikTok impone límites de uso sobre la cantidad de videos que puedes acceder y la frecuencia de las solicitudes.
- **Revisión de la aplicación:** Si tu aplicación utiliza ciertos permisos, puede ser necesario pasar por una revisión.

WhatsApp API (WhatsApp Business API)

Permisos Requeridos:

1. **messages:** Permite enviar y recibir mensajes de texto, imágenes, videos, documentos, etc.
 - **Uso:** Enviar y recibir mensajes a través de WhatsApp Business.
 - **Restricciones:** Solo está disponible para cuentas de WhatsApp Business verificadas.
2. **contacts:** Permite acceder a la lista de contactos de la cuenta de WhatsApp Business.
 - **Uso:** Verificar los números de teléfono de los contactos con los que interactúas.
 - **Restricciones:** Solo disponible para empresas con una cuenta de WhatsApp Business verificada.

Requisitos:

- **Registrar número de teléfono:** Necesitarás registrar un número de teléfono a través de la WhatsApp Business API.
- **API Key:** Debes obtener una clave de API que te permita autenticar las solicitudes.
- **Autenticación:** Generalmente se usa un token de autenticación en lugar de un flujo OAuth estándar.

Limitaciones:

- **Límites de uso:** Hay límites en la cantidad de mensajes que puedes enviar por día.
- **Verificación de empresa:** Solo las empresas verificadas pueden acceder a la API de WhatsApp Business.

2.TABLA DE CARACTERISTICAS

Red Social	Permisos Requeridos	Tipo de Autenticación	Acciones Disponibles	Requisitos	Limitaciones	Documentación
Facebook (Meta)	- user_posts - pages_manage_posts - public_profile - email	OAuth 2.0	<ul style="list-style-type: none"> - Leer publicaciones del usuario - Gestionar publicaciones en páginas - Obtener información básica de perfil - Obtener correo electrónico del usuario - Obtener información básica del perfil de Instagram - Obtener y gestionar publicaciones de usuario - Obtener lista de páginas vinculadas de Facebook - Obtener perfil básico (nombre, foto, título profesional) - Obtener correo electrónico del usuario - Publicar contenido en el feed del usuario - Obtener información básica de perfil - Obtener lista de videos del usuario - Subir videos a la cuenta del usuario 	<ul style="list-style-type: none"> - Registro en Facebook Developer Portal - Token de acceso - Registro en Instagram Graph API - Token de acceso - Registro en LinkedIn Developer Portal - Token de acceso - Registro en TikTok Developer Portal - Token de acceso 	<ul style="list-style-type: none"> - Límite de solicitudes diarias - Requiere revisión de permisos avanzados (ej. user_posts) - Límite de solicitudes diarias - Requiere revisión de permisos avanzados - Límite de solicitudes diarias - Necesita aprobación para publicar contenido - Límite de solicitudes diarias - Requiere revisión de la aplicación para algunos permisos 	Facebook Graph API
Instagram (Meta)	- instagram_basic - user_media - pages_show_list	OAuth 2.0				Instagram Graph API
LinkedIn	- r_liteprofile - r_emailaddress - w_member_social	OAuth 2.0				LinkedIn API
TikTok	- user.info.basic - video.list - video.upload	OAuth 2.0				TikTok API

WhatsApp - messages - contacts API Key	- Enviar y recibir mensajes - Acceder a los contactos de la cuenta	- Registro de número en <u>WhatsApp Business API</u> - API Key	- Límite de mensajes por día - Solo disponible para cuentas <u>WhatsApp Business API</u> verificadas
---	--	--	--

3. SELECCIÓN DE LLM

El objetivo es evaluar y seleccionar el **mejor modelo de lenguaje grande (LLM)** para el proyecto. Los **LLMs** (como GPT, PaLM, Claude, etc.) son herramientas poderosas para tareas como generación de texto, comprensión, traducción y análisis semántico.

1. ¿Qué es un LLM?

Un **modelo de lenguaje grande (LLM)** es un tipo de modelo de inteligencia artificial entrenado con enormes cantidades de datos textuales para realizar tareas relacionadas con el lenguaje natural. Estos modelos son capaces de generar texto coherente y relevante, entender preguntas, realizar resúmenes, tradiciones, análisis de sentimientos y mucho más.

2. ¿Por qué elegir un LLM?

El **LLM** es fundamental en tu arquitectura porque será el componente encargado de **procesar y generar respuestas inteligentes** a partir de los datos recibidos de las **APIs sociales** (Facebook, Instagram, LinkedIn, etc.). De acuerdo a tu flujo de trabajo, un **LLM** puede ayudarte a:

- **Generación de texto:** Respuestas automáticas, interacciones de chatbot, descripciones generadas de manera automática.
- **Análisis de sentimientos:** Analizar la emoción o el sentimiento detrás de un mensaje o publicación.
- **Respuestas inteligentes:** Generación de respuestas a consultas basadas en datos de redes sociales y otras fuentes.
- **Traducción:** Si es necesario manejar múltiples idiomas.
- **Clasificación:** Agrupar datos o generar categorías a partir de información no estructurada.

El LLM elegido debe tener **capacidad para integrarse bien con tus APIs** y responder a las necesidades de tu sistema de manera eficiente.

3. Factores clave para elegir un LLM.

a) Precisión y Rendimiento

- **¿Qué tan bien puede generar texto relevante y preciso?**
- **¿Cómo maneja las tareas específicas (como análisis de sentimientos, traducción, etc.)?**

Importancia: El rendimiento del modelo es crucial para garantizar que el sistema responda correctamente a las solicitudes del usuario y que los resultados generados sean útiles.

b) Costo

- **¿Cuánto cuesta usar el modelo?**
- **¿Es el costo sostenible para el volumen de interacciones que esperas manejar?**

Importancia: Los costos de los LLMs pueden variar considerablemente, dependiendo del proveedor y del uso. Algunos modelos cobran por token o por volumen de uso, lo que puede sumar costos si hay muchas interacciones.

c) Escalabilidad

- **¿Cómo maneja el modelo grandes volúmenes de datos o solicitudes simultáneas?**
- **¿Es capaz de ajustarse al crecimiento futuro de tu aplicación?**

Importancia: La escalabilidad es importante para asegurar que tu servicio pueda manejar el aumento en el número de usuarios o interacciones sin sacrificar el rendimiento.

d) Flexibilidad y Personalización

- **¿Puedes personalizar el modelo para ajustarlo a las necesidades específicas de tu aplicación?**
- **¿El modelo permite agregar o adaptar comportamientos específicos (por ejemplo, entrenar con datos adicionales)?**

Importancia: La flexibilidad es importante si tu aplicación necesita características específicas que no están disponibles de manera predeterminada.

e) Accesibilidad y Documentación

- **¿Qué tan fácil es integrar y usar el modelo?**
- **¿La documentación y la comunidad de desarrolladores son lo suficientemente buenas para implementar y solucionar problemas rápidamente?**

Importancia: Una buena documentación facilita la integración y mejora la eficiencia del equipo de desarrollo.

f) Mantenimiento y Actualización

- **¿Con qué frecuencia se actualiza el modelo?**
- **¿Cómo maneja las actualizaciones, mejoras y correcciones de errores?**

Importancia: Un modelo bien mantenido asegura que puedes aprovechar nuevas funcionalidades y correcciones de errores sin complicaciones.

4. Comparación de Modelos de LLM Populares

a) OpenAI GPT (incluido GPT-4)

Pros:

- **Amplio soporte y adopción:** GPT es uno de los modelos de lenguaje más populares y ampliamente utilizados, con una gran comunidad de desarrolladores.
- **Versatilidad:** Se utiliza para una amplia variedad de tareas, desde generación de texto hasta análisis de sentimientos y generación de código.
- **Documentación y recursos:** OpenAI tiene documentación detallada, tutoriales, y herramientas como el Playground, lo que facilita la integración.

Contras:

- **Costo:** Puede ser costoso, especialmente si manejas grandes volúmenes de solicitudes, ya que OpenAI cobra por tokens procesados.
- **Dependencia de la nube:** Solo está disponible a través de la API en la nube, lo que implica dependencias de conexión a Internet y límites de uso.

Casos de uso recomendados:

- Generación de texto (chatbots, artículos, contenido)
- Respuestas automáticas
- Análisis de sentimiento

Recomendación: Si buscas un modelo robusto y con gran soporte para una amplia gama de tareas, **OpenAI GPT** es una excelente opción.

b) Google PaLM (Pathways Language Model)

Pros:

- **Alta precisión y razonamiento:** PaLM ha demostrado ser particularmente bueno para tareas que requieren un razonamiento más complejo y comprensión profunda.
- **Versatilidad:** Similar a GPT, es adecuado para tareas variadas como traducción, análisis y resumen de textos.

Contras:

- **Accesibilidad limitada:** Aunque Google PaLM es potente, no está tan accesible como otros modelos y está más limitado a Google Cloud.
- **Costo:** Similar a GPT, el costo de uso a gran escala puede ser elevado.

Casos de uso recomendados:

- Generación de texto compleja
- Búsqueda avanzada y análisis de datos
- Tareas de razonamiento y abstracción

Recomendación: Si necesitas un modelo de **alto rendimiento para tareas de razonamiento complejo**, **PaLM** es una opción fuerte.

c) Anthropic Claude

Pros:

- **Ética y seguridad:** Claude es conocido por su enfoque en seguridad y la reducción de sesgos en las respuestas generadas. Es una buena opción si la seguridad es una prioridad.
- **Interacción segura:** El modelo está diseñado para ofrecer una interacción más ética y controlada con los usuarios.

Contras:

- **Menos popular:** Aunque es una excelente opción en términos de seguridad, su adopción es más limitada en comparación con GPT.
- **Menor flexibilidad:** En comparación con GPT y PaLM, Claude tiene una menor flexibilidad en términos de personalización.

Casos de uso recomendados:

- Aplicaciones donde la seguridad y la ética son prioritarias
- Chatbots que deben interactuar de forma controlada y ética con los usuarios

Recomendación: Si la **seguridad y la ética** son críticas en tu proyecto (por ejemplo, aplicaciones con contenido sensible), **Claude** es una buena opción.

5. Elección recomendada para el proyecto

Para un proyecto que maneja múltiples **APIs sociales**, **OpenAI GPT (GPT-4)** por varias razones:

- **Versatilidad:** Es adecuado para una variedad de tareas como generación de texto, análisis de sentimientos, y respuestas automatizadas.
- **Ecosistema de integración:** Su comunidad y documentación extensa facilitan la integración con diversas APIs y plataformas.
- **Escalabilidad:** Aunque es costoso, ofrece escalabilidad según el volumen de interacciones que manejes.

4. PROPUESTA DE ARQUITECTURA

La arquitectura del proyecto debe estar diseñada de manera modular, escalable y eficiente, asegurando que cada componente pueda comunicarse de forma fluida y que el sistema pueda crecer conforme a la demanda.

Arquitectura Propuesta:

Visión General de la Arquitectura

Componentes Clave:

1. **Frontend (Portal Web):** Una aplicación web que interactúa con los usuarios a través de una interfaz React/Vue.
2. **Backend Central:** Un backend centralizado que maneja la lógica de negocio, integra las APIs sociales y se comunica con el servicio LLM.
3. **Servicio LLM:** Un servicio dedicado a interactuar con un modelo de lenguaje grande (LLM), como GPT, PaLM o Claude.
4. **APIs Sociales:** Integración con Facebook, Instagram, LinkedIn, TikTok, WhatsApp a través de sus respectivas APIs.

1. Frontend (Portal Web):

El **Portal Web** será la interfaz del usuario donde interactuarán los usuarios finales, tendrá la siguiente estructura:

- **Interfaz de Usuario:** Un diseño interactivo que permite a los usuarios ingresar datos, recibir respuestas automáticas generadas por el LLM, y visualizar interacciones con las redes sociales.

- **Conexión con el Backend:** La comunicación entre el **Frontend** y el **Backend Central** se hará mediante **RESTful APIs** o **GraphQL**. Esto permitirá realizar solicitudes (como enviar datos o recibir respuestas del LLM) y mostrar los resultados de forma eficiente.

2. Backend Central (FastAPI / NestJS):

El **Backend Central** es el cerebro del sistema, que orquesta la comunicación entre los diferentes componentes y maneja la lógica de negocio. Aquí te doy los detalles clave:

- **Framework:** Utilizar **FastAPI** (si prefieres algo ligero y rápido) o **NestJS** (si prefieres un enfoque basado en clases y una mayor estructura) para desarrollar el backend.
- **Funciones Principales:**
 - **Autenticación y autorización** de usuarios mediante OAuth 2.0 para acceder a las redes sociales (Facebook, Instagram, LinkedIn, etc.).
 - **Gestión de solicitudes y respuestas** del usuario. El backend recibe las solicitudes del frontend, las procesa y llama a los servicios apropiados.
 - **Integración con las APIs Sociales:** El backend se encargará de interactuar con las APIs externas de Facebook, Instagram, LinkedIn, TikTok y WhatsApp para obtener datos del usuario, publicar contenido o realizar otras acciones.
 - **Interacción con el Servicio LLM:** El backend también manejará la comunicación con el **Servicio LLM** para generar respuestas automáticas a partir de las interacciones con las redes sociales.

3. Servicio LLM (Modelo de Lenguaje Grande):

El **Servicio LLM** es donde ocurren las interacciones de inteligencia artificial. Este servicio puede ser un microservicio independiente que se comunica con el backend central.

- **Modelo de Lenguaje Grande:** El LLM puede ser uno de los siguientes:
 - **GPT** (por ejemplo, **GPT-4** de OpenAI): Potente en generación de texto, respuestas inteligentes y análisis semántico.
 - **PaLM** (Google): Ideal para tareas que requieren razonamiento profundo.
 - **Claude** (Anthropic): Orientado a la ética y control de seguridad.
- **Interacción con Backend:** El backend envía los datos necesarios al servicio LLM (por ejemplo, texto de redes sociales) y recibe la respuesta generada. Esta respuesta se puede utilizar para realizar tareas como respuestas automáticas, análisis de sentimiento, o generación de contenido.
- **Escalabilidad:** Este servicio debería ser **escalable** mediante contenedores (usando Docker) o microservicios (con Kubernetes) para manejar un alto volumen de interacciones. Además, puede estar alojado

en **cloud** (AWS, GCP, Azure) para asegurar la disponibilidad y escalabilidad.

4. Integración de APIs Sociales:

Cada red social que mencionas (Facebook, Instagram, LinkedIn, TikTok, WhatsApp) tiene su propia API, y el backend central será el encargado de interactuar con ellas.

- **Facebook e Instagram:** A través de la **Facebook Graph API**, el backend podrá interactuar con las cuentas de Facebook e Instagram. Podrá obtener datos de publicaciones, obtener perfiles de usuario, gestionar contenido y realizar publicaciones.
- **LinkedIn:** La **LinkedIn API** permitirá obtener información sobre el perfil del usuario, así como publicar contenido en su nombre.
- **TikTok:** Usando la **TikTok API**, el backend podrá acceder a los videos del usuario, cargar nuevos videos y obtener información básica del perfil.
- **WhatsApp:** A través de la **WhatsApp Business API**, el backend puede enviar y recibir mensajes, gestionar contactos y realizar interacciones a través de WhatsApp.
- **Flujos de datos:** El flujo de datos será el siguiente:
 - El **Frontend** hace una solicitud a la API del **Backend Central**.
 - El **Backend** puede procesar esa solicitud directamente o interactuar con las APIs sociales para obtener los datos requeridos.
 - El **Backend** también puede enviar los datos a través del servicio LLM para generar respuestas automáticas o análisis avanzados.
 - El **Backend** devuelve los resultados al **Frontend**, que los muestra al usuario.