Tarea 1

## Fecha de entrega

Hasta el domingo 5 de enero de 2025, 23:59

## Formato de entrega

Para cada grupo se solicitará lo siguiente:

1. Enlace a repositorio público en GitHub
2. Enlace a URL pública de su proyecto en vivo (ejemplo: en render.com)

Existirá un formulario en la plataforma del curso donde podrás enviar esta información.

## Proyectos

Habrá dos opciones de proyectos, deben escoger solo uno:

1. **Opción 1 (recomendado): Recomendador de películas.**
2. **Opción 2: Proyecto abierto.** Para los grupos que querían explorar otro tipo de proyecto.

Se recomienda realizar el recomendador de películas, ya que en las clases usamos este proyecto como base. Pueden usar como apoyo el [código de Next Muby](https://github.com/DIAG-TALLER-2024/next-muby), aunque se les solicitará personalización (para que no todos los proyectos se vean iguales) y funcionalidades adicionales (para que no sea simplemente entregar el proyecto base).

### Opción 1: Proyecto de recomendación de películas

#### Incremento 1: El chat con historial (3 puntos)

Aquí se evaluará que cada grupo puede estar a la par con el código visto de Next Muby, por lo que las funcionalidades a desarrollar son las siguientes:

1. Comenzar la conversación con el Chat Bot
2. Enviar mensaje al Chat Bot (para solicitar una recomendación)
3. El Chat Bot debe responder a los mensajes del usuario
4. Mantener un historial de mensajes: si recargo la página, debo seguir viendo el historial de mensajes.

Requisitos técnicos:

* **La aplicación debe crearse utilizando el framework Flask.** Pueden instalar los packages que deseen convenientes.
* **El historial de mensajes debe realizarse con una base de datos.** No es obligatorio que usen Turso ni tampoco SQLAlchemy como vimos en clases (pueden usar la base de datos y ORM de su preferencia).

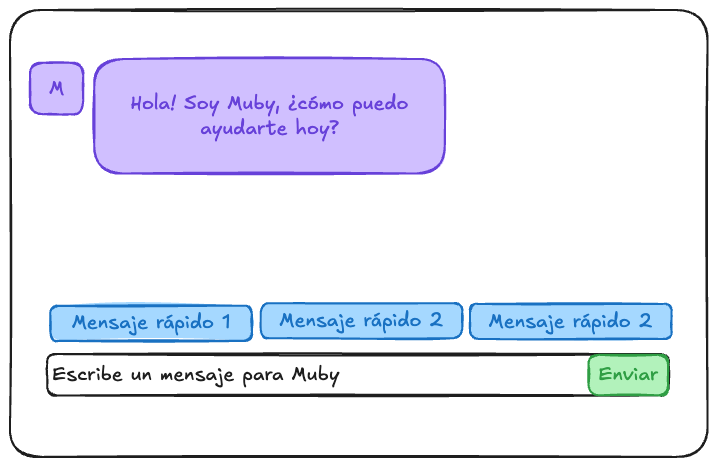
Requisitos no técnicos:

* **La aplicación debe estar personalizada**. Es decir, el bot debe tener un nombre y una imagen que lo caracterice.

#### Incremento 2: Mensajes pre-hechos (1 punto)

Agrega a la aplicación 3 mensajes pre-hechos, para que los usuarios no tengan que escribir la pregunta completa.

A continuación verás una imagen de referencia para esta funcionalidad, donde “Mensaje rápido X” son botones que al presionar haces que el usuario se ahorre en escribir el mensaje:



Tip: esto se puede hacer parecido a lo que hicimos en la clase 3 y comienzos de la clase 4.

#### Incremento 3: Perfil de usuario (2 puntos)

Agregarás a tu aplicación una página donde existirá el perfil del usuario, que él o ella pueden editar. La idea es que el perfil de usuario incluya cosas como “película favorita” o “género favorito”, de manera que el bot tenga conocimiento de algunos gustos de la persona.

Requisitos técnicos:

* **El perfil de usuario debe guardarse en la base de datos**. Así, los datos se persisten y pueden ser consultados.
* **Debe existir una página para actualizar los datos del usuario**. Es decir, la página deberá mostrar un formulario con las preguntas que quieres que responda.
* **El bot debe tener conocimientos de las preferencias del usuario**. Es decir, debes modificar el *prompt* para que incluya los datos del usuario, de tal manera de que el bot tenga contexto adicional para la conversación.

### Opción 2: Otros proyectos

Si decidieron hacer otro proyecto, se evaluará un avance equivalente al proyecto de Next Muby, por lo tanto:

#### Incremento 1: funcionalidad base (3 puntos)

* Debe existir una interfaz web mediante la cual tu aplicación ofrezca alguna funcionalidad que utilice la API de OpenAI.
  + Por ejemplo, puede ser un Chat Bot como el recomendador de películas
  + Puede ser un formulario donde envíes una imagen y otra información, para luego con OpenAI generar una respuesta hacia el usuario.
* Se debe persistir información en una base de datos, que el usuario luego puede consultar.
  + Por ejemplo, si es un Chat Bot, basta con colocar la conversación completa, al igual que en “Next Muby”.

#### Incremento 2: acciones rápidas (1 punto)

Agregar una opción para interactuar rápidamente con el LLM.

* Por ejemplo, si es un Chat Bot, agregar mensajes pre-hechos
* Si tienes dudas qué implementar aquí, puedes preguntar en el foro de curso o por correo a [ngomez@hey.com](mailto:ngomez@hey.com)

### Incremento 3: personalización (2 puntos)

En tu aplicación, debe existir una forma en que el usuario edite su perfil o preferencias de manera de que el agente tenga más contexto al momento de escribir una respuesta.

## Consideraciones técnicas

### Tecnologías

Para construir la aplicación deben utilizar el framework Flask y para el bot la API de OpenAI.

Esas son las únicas exigencias técnicas. En cuanto a la base de datos, pueden utilizar cualquiera, al igual que con el estilo de la aplicación. Se sugiere usar SQLite con Turso porque es lo que vimos en clases, y porque su capa gratuita es muy permisiva, y Bootstrap. Pero si tienes experiencias con otras tecnologías, puedes usarlas (mientras sigas usando Flask y API de OpenAI).

### Inicio de sesión

No es necesario que para esta entrega tengan un inicio de sesión, y tampoco habrá bonificación si lo hacen. Así que pueden asumir que existirá solo un usuario en la aplicación por ahora.

## Pauta de evaluación

Puedes verla [aquí](https://docs.google.com/spreadsheets/d/1wR2UUsHGsgIhWaZ0lC_8Yz3kRCjGLYFnX3feBroOymw/edit?usp=sharing).

## Nota final

Puntos obtenidos + 1