



## Tarea 4

### Entrega

- **Fecha y hora:** Lunes 11 de diciembre del 2023, a las 22:00
- **Lugar:** Repositorio individual en la organización del curso en GitHub

### Objetivos

- **Comprender** los fundamentos de la automatización en aplicaciones de mensajería
- **Aprender** a generar sus propios bots de Telegram
- **Producir** documentación efectiva y clara del proceso realizado.

### Descripción

En esta tarea, tendrán la oportunidad de crear sus propios bots de Telegram e integrarlos con una API externa de su elección. Podrán comunicarse con este bot y este será capaz de responder con la información que reciba de la API. De esta manera, podrán comprender cómo agilizar procesos, mejorar las experiencias de usuario y crear soluciones eficientes a través de la automatización en aplicaciones de mensajería.

Para esta tarea, deberán crear un bot de Telegram capaz de responder cada vez que un usuario le envíe algún mensaje con un comando específico. Por ejemplo, que se active cuando le llegue la instrucción `/start`. Esta respuesta debe contener información extraída de alguna API externa mediante una solicitud de tipo GET. Su bot puede realizar más solicitudes a la API si así lo desean, siempre y cuando se cumpla con el mínimo de una solicitud tipo GET.

Tendrán que crear un archivo `index.js` que maneje la lógica del bot y la conexión a la API, así como también una documentación completa acerca de su código. Este archivo debe estar escrito utilizando el *framework* Koa.

### Conexión a API

Existen muchas APIs externas de diversos temas disponibles en la web, por lo que esta elección queda a su criterio. De todas formas, les dejamos tres sugerencias de APIs públicas y gratis que pueden utilizar:

- **[PokéAPI](#):** Esta API posee una documentación muy bien detallada para que aprendas a consumirla. Además, tiene mucha información de los Pokémon como sus movimientos, habilidades, tipos, poderes, habitat y más.
- **[Frankfurter](#):** Esta es una API de código abierto que te permite conocer los tipos de cambios de divisas, ya sean los actuales o históricos, publicados por el Banco Central Europeo.
- **[Fake Store API](#):** se utiliza principalmente para prototipos de sitios web, de comercio electrónico o de compras. Esta API te proporcionará datos ficticios para plataformas de este tipo.

Puede que para hacer uso de alguna API deban generar una key personal, normalmente esta información se encuentra documentada en la misma API.

## Uso de Ngrok

Para que su bot de Telegram pueda recibir actualizaciones (mensajes, comandos, etc.) de los usuarios, se debe hacer uso de webhooks. Para esto, se necesita una URL accesible públicamente a la que Telegram pueda enviar actualizaciones. Aquí es donde la librería **Ngrok** resulta útil, ya que les permite exponer su servidor local de desarrollo (donde su bot está corriendo) a internet y proporcionar una URL pública que puedes usar para configurar tu webhook.

Para utilizar Ngrok, deben descargarlo desde su [página oficial](#), hay distintas opciones según su sistema operativo o ambiente de trabajo que usen.

Con *Ngrok* ya instalado, deben ejecutar el siguiente comando para generar un enlace de webhook:

```
ngrok http [puerto]
```

Donde [puerto] es el puerto en que tu servidor local está escuchando (donde se está ejecutando tu bot de Telegram). La URL generada por este comando es la que usarás como endpoint para tu webhook en Telegram.

## Documentación

Este ítem tiene como propósito:

- Explicar cómo funciona el código que implementaron.
- Indicar apropiadamente cómo correr su programa y que librerías hay que instalar (si es que aplica).

Además, en sus respectivos repositorios encontrarán una serie de preguntas teóricas acerca de su tarea. Deberán responder estas preguntas de forma más completa posible.

Para un mejor entendimiento sobre cómo escribir la documentación, se les hará entrega de una plantilla con la estructura que debe seguir, su descripción y las preguntas que deben responder.

## Consideraciones

- Prohibición de bibliotecas:

Está estrictamente prohibido la utilización de cualquier librería más allá de `koa-router` y `koa-parser`. La implementación de su bot debe ser realizada desde cero, por lo que no pueden utilizar librerías externas que lo hagan por ustedes. Cualquier duda de librerías específicas que tengan pueden hacerla por las *issues* del curso.

- Variables de entorno:

Es de suma importancia que utilicen un archivo `.env` para guardar todas las variables de entorno que deban utilizar en el desarrollo de su bot. Utilizar las variables directamente en el código significará un descuento.

- Política de revisión antiplagio:

Les recordamos que se aplicará una política de revisión antiplagio rigurosa en esta tarea. Se espera que trabajen de manera independiente y cumpla con los principios éticos de la integridad académica. Cualquier forma de copia o plagio, ya sea de fuentes en línea, de otros estudiantes o de cualquier otra fuente, resultará en consecuencias académicas negativas, que pueden incluir la anulación de la tarea y medidas disciplinarias adicionales, de acuerdo con las políticas de integridad académica de la institución.

## Entregables

Cada entrega deberá incluir los siguientes archivos:

- `README.md` con la documentación de tu programa. Deben mencionar brevemente aquello que pudieron implementar (y cómo lo hicieron) y también lo que no. Deben indicar cómo correr tu programa y responder las preguntas indicadas en el *template* que les facilitamos. Para finalizar, debes explicar claramente cómo funciona cada función de tu programa.
- Archivo `index.js` que contenga el servidor de su bot de Telegram.

## Rúbrica

A continuación, se describe el criterio de cada uno de los ítems de la rúbrica con la que se evaluará esta entrega. La nota se calcula al 50 % considerando un total de 15 puntos.

- **Servidor** [1 punto]: Se debe crear un bot de Telegram a través de un servidor utilizando Koa.
- **Variables de entorno** [1 punto]: se define correctamente las variables de entorno necesarias (incluyendo el *token* del bot y el URL del *webhook*) en un archivo `.env`.
- **URL API** [1 punto]: se espera que configuren correctamente la URL que conecta su servidor de Koa con su bot de telegram, utilizando los parámetros correspondientes.
- **Recepción y envío de mensajes** [3 puntos]: se debe crear un endpoint que reciba mensajes de Telegram. Este identifica correctamente el comando especificado para empezar la conversación (como por ejemplo, `/start`). Además, se logra enviar una respuesta al detectar el comando adecuado.
- **Acceso API pública** [2 puntos]: se espera que su servidor pueda acceder a una API pública para buscar información, parsear adecuadamente la respuesta de la API y enviarla.
- **Documentación** [7 puntos]: se espera que se explique de manera clara, ordenada y suficiente cada parte del programa, además de mencionar lo que se logró hacer y lo que no. Se debe notar un claro entendimiento de cómo se realizó (y cómo se corre) el código entregado. También, se esperan respuestas claras y completas explicando las preguntas teóricas asociadas a la tarea.

## Dudas

Cualquier duda que se presente acerca del enunciado debe consultarla en las [issues](#) del repositorio del curso. Recuerden que como equipo docente estaremos atentos para poder ayudarlos :)