

**PRIMERA ENTREGA: PROPUESTA DE PROYECTO**

# Información básica de tu proyecto

* Título del proyecto Bot de Bienvenida y Comandos para Discord
* Nombre(s) del/los estudiante(s) : Andres Felipe Eusse y Jhon Pulgarin Restrepo
* Fecha : 21/10/2025
* Curso: Python de 0 a 100
* Breve descripción (2-3 líneas) : un bot simple para un servidor de Discord que automatiza dos tareas clave: dar la bienvenida a nuevos miembros y responder a comandos básicos predefinidos por el chat.

# 2. Describe el proyecto en el que vas a trabajar

**2.1 Problemática o Necesidad**

* ¿Qué problema resuelve tu proyecto? : En muchos servidores de Discord, los moderadores siempre terminan haciendo lo mismo: dar la bienvenida a los nuevos y responder preguntas repetidas como “¿cuáles son las reglas?”. Este proyecto busca automatizar todo eso para que esas interacciones iniciales sean más rápidas y no tengan que hacerlo manualmente.
* ¿Por qué es importante o relevante? : Ahorra tiempo a los administradores del servidor y hace que la experiencia de los nuevos usuarios sea mucho mejor, ya que se sienten bienvenidos desde el primer momento. También es una buena forma de empezar a conocer cómo funcionan las APIs.
* ¿A quién está dirigido? (usuario objetivo) : Administradores y moderadores de servidores de Discord que quieren automatizar tareas básicas de la comunidad y simplificar su trabajo.

**Ejemplo:**

*"El proyecto busca automatizar el análisis de ventas mensuales de pequeños negocios que actualmente realizan este proceso manualmente en Excel, consumiendo 4-5 horas semanales."*

**2.2 Descripción General**

* ¿Qué hace tu proyecto? (funcionalidad principal) : El proyecto es un script de Python que se conecta a la API de Discord. Su función principal es escuchar dos tipos de eventos: la entrada de un nuevo miembro al servidor y los mensajes enviados en los canales de chat.
* ¿Cómo funcionará? (descripción del flujo) : El script se ejecutará en un servidor (o localmente para pruebas) y usará la librería **Discord.py** para conectarse con la API de Discord mediante un token de bot. Cuando un nuevo usuario se una al servidor, el bot detectará el evento **on\_member\_join** y enviará automáticamente un mensaje de bienvenida predefinido en un canal específico (por ejemplo, **#bienvenidas**). Además, si un usuario escribe un comando como **!reglas**, el bot lo reconocerá a través del evento **on\_message** y responderá con la información correspondiente.
* ¿Qué resultados generará? (salidas esperadas) : La salida principal son mensajes automáticos publicados por el bot dentro del servidor de Discord, ya sea en el canal de bienvenida o como respuesta a comandos en los canales de chat.

**2.3 Alcance del Proyecto**

**Qué SI incluirá:**

* Listar funcionalidades específicas : El proyecto incluye una conexión y autenticación segura del bot con la API de Discord, además del envío automático de un mensaje de bienvenida personalizado cada vez que un nuevo miembro se une al servidor. El bot también puede responder a varios comandos básicos de chat (por ejemplo, **!hola**, **!info**, **!reglas**) y maneja el token de forma segura, leyéndolo desde un archivo externo o una variable de entorno en lugar de incluirlo directamente en el código. Además, cuenta con un sistema de **logging** básico que muestra en la consola cuándo el bot se conecta y cuándo envía un mensaje de bienvenida.
* Mínimo 4-6 funcionalidades principales

**Qué NO incluirá (limitaciones):**

* Características que quedan fuera del alcance :

El proyecto no incluirá funciones de **moderación avanzada**, por lo que el bot no podrá expulsar, banear ni silenciar usuarios. Esto se debe a que dichas acciones requieren una gestión compleja de permisos y roles, lo cual supera el alcance de un proyecto introductorio.

Tampoco se realizará **integración con otras APIs** (como Twitter o bases de datos), ya que el objetivo principal es enfocarse en comprender y manejar correctamente una sola API: **Discord.py**.

Finalmente, el bot **no contará con una interfaz gráfica o web**, pues se ejecutará directamente desde la terminal o consola. El desarrollo de una UI queda fuera del alcance de las librerías utilizadas en este proyecto.

**3. Selecciona una categoría para tu proyecto**

Selecciona UNA categoría principal:

* **Análisis de Datos**: Procesamiento, limpieza y análisis de datasets
* **Automatización**: Scripts que automatizan tareas repetitivas
* **Visualización de Datos**: Dashboards o reportes visuales
* **Aplicación de Consola**: Herramienta interactiva por terminal
* **Web Scraping**: Extracción y análisis de datos web
* **API/Servicio**: Desarrollo de API REST o microservicio
* **Machine Learning**: Modelo predictivo o clasificación
* **Juego/Simulación**: Aplicación lúdica o simulador
* **Otra**: Especificar

**Escogida**: Automatización.

1. **¿Qué tecnologías y/o herramientas usarás?**

*Pueden ser vistas en el curso o que hayas investigado por tu cuenta*

**4.2 Librerías de Python**

Listar las librerías que usarás y JUSTIFICAR su uso:

**Mínimo requerido:** 2 librerías

**Discord.py:** Es la librería principal y esencial para interactuar con la API de Discord. Permite conectar el bot, escuchar eventos como la llegada de nuevos miembros y enviar mensajes de manera sencilla y estructurada.

**python-dotenv** (o el módulo integrado **os**): Se usa para manejar de forma segura el token del bot. Gracias a esta librería, el token puede almacenarse en un archivo**.env** y cargarse desde allí, evitando escribirlo directamente en el código, lo cual sería una mala práctica de seguridad.

**4.4 Fuentes de Datos**

* + ¿De dónde obtendrás los datos?
  + Datasets públicos, APIs, archivos CSV, web scraping, etc. ● Incluir enlaces o referencias

# 5. Arquitectura y modelación de tu proyecto(opcional)

**5.1 Estructura del Proyecto**

Diagrama, digital o a mano, cómo se relacionarán los diferentes módulos y/o funciones de tu proyecto

**5.2 Componentes Principales**

Listar y describir los módulos/funciones clave:

1. **Módulo de carga de datos** 
   1. Funciones para leer CSV, JSON, o APIs
2. **Módulo de procesamiento** 
   1. Limpieza y transformación de datos
3. **Módulo de análisis** 
   1. Cálculos estadísticos y lógica de negocio
4. **Módulo de visualización** 
   1. Generación de gráficos y reportes
5. **Módulo principal (main)** 
   1. Orquestación del flujo del programa

# 6. Lista los conceptos del curso que vas a usar en tu proyecto

Indica cómo aplicarás los conceptos vistos en clase:

* **Variables y tipos de datos**: Para almacenar [especificar qué]
* **Estructuras de control**: Validaciones y flujos condicionales
* **Funciones**: Modularización del código en [X] funciones
* **Funciones Lambda**: Transformaciones de datos con map/filter
* \*\*\*args y **kwargs**: Funciones flexibles para [especificar]
* **Pandas**: Manipulación de DataFrames con [X] operaciones
* **Matplotlib/Seaborn**: [X] tipos de gráficos
* **Manejo de archivos**: Lectura/escritura de CSV, JSON, etc.
* **Manejo de errores**: Try/except para casos [especificar]
* **Flet**
* **UV**
* **FastAPI**
* **IA con Python**