

**Laboratorio 8**  
**Exploración y Uso Avanzado de Plataformas IA, Repositorios**  
**Profesionales y Herramientas Globales para el desarrollo de IA**  
**y de SW**

**Materia:**

Profundización de inteligencia artificial

**Participantes:**

Ana Maria Navarro Fernandez

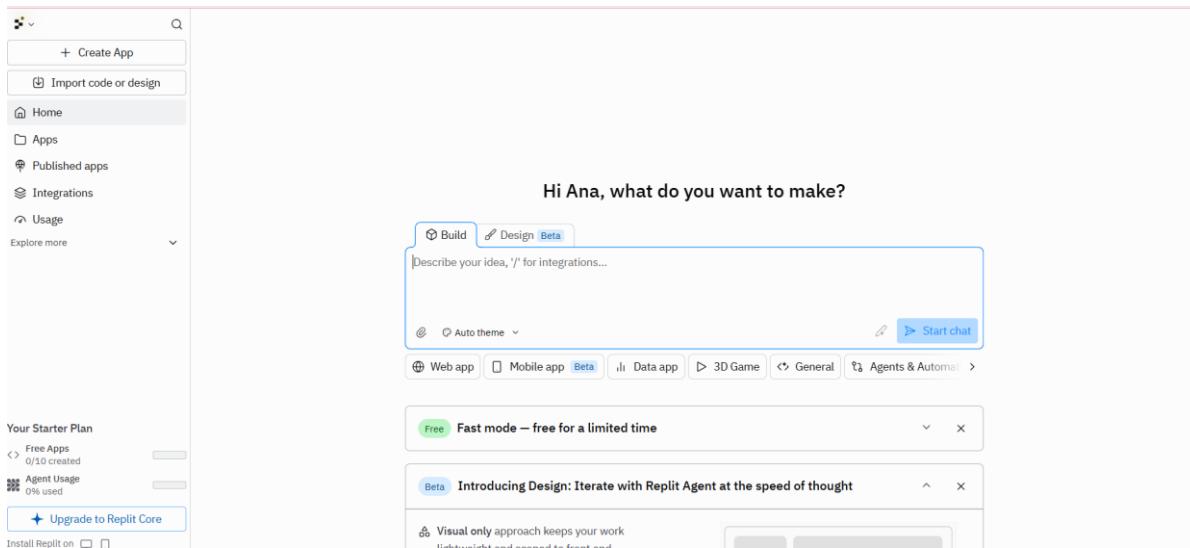
**Profesor:**

Carlos Betancourt correa

Universidad de Manizales  
ingeniería en sistemas y telecomunicaciones  
Manizales, Caldas, Colombia

## Laboratorio 8

### 2.12 Replit



#### 1. Descripción de la herramienta

Replit es una plataforma en línea que permite ejecutar, desarrollar y desplegar aplicaciones sin necesidad de instalar nada en el computador. Incluye integración con Replit AI Agent, el cual puede crear, corregir y desplegar código automáticamente basado en instrucciones del usuario.

#### 2. ¿Qué se hizo en el laboratorio?

Paso 1 – Ingreso a Replit

Se accedió a la plataforma desde: <https://replit.com/>

Una vez en la pantalla inicial, Replit muestra el mensaje “Hi Ana, ¿what do you want to make?”, donde se puede iniciar un proyecto usando Replit Agent.

#### Paso 2 – Crear un proyecto con Replit Agent

El agente fue instruido para crear un:

Microproyecto en Python usando Flask con dos endpoints:

- GET / → devuelve: “API del laboratorio 8 lista”
- GET /suma?a=2&b=3 → devuelve la suma en formato JSON

Replit Agent generó automáticamente:

- main.py
- requirements.txt
- .gitignore
- Configuración del servidor Flask
- Pruebas automáticas del código

#### Paso 3 – Revisión automática del código

El agente ejecutó tareas como:

- Revisar implementación de la API
- Editar dependencias
- Verificar logs del servidor
- Confirmar que los endpoints respondieran correctamente

En las capturas se observan mensajes como:

- “Reviewed Flask API implementation”
- “Configured Flask API to run python main.py”
- “Verified Flask app logs”

#### Paso 4 – Ejecución de la API

La API se ejecutó exitosamente en el puerto 5000.  
Los endpoints funcionan correctamente:

## Laboratorio 8

Endpoint raíz

GET /

→ "API del laboratorio 8 lista"

Endpoint suma

Ejemplo:

GET /suma?a=2&b=3

→ { "resultado": 5 }

### Paso 5 – Publicación

Replit permitió configurar un dominio para la API. Sin embargo, para publicarla de forma completa solicita actualizar el plan de la cuenta, por lo cual el despliegue quedó configurado pero no publicado.

Captura:

flask-sum-api-anamariar.replit.app

Status: Available (pero requiere upgrade para publicar)

### 3. Resultados

- Se creó una API funcional usando Flask con dos endpoints.
- El agente de Replit completó todas las tareas de desarrollo automáticamente.
- Los endpoints se probaron y respondieron correctamente.
- Se configuró la opción de publicación de dominio.
- El proyecto quedó completamente funcional dentro del entorno de Replit.

### 4. Conclusión

El uso de Replit, combinado con su agente inteligente, permite desarrollar aplicaciones completas sin necesidad de configuración manual. Este laboratorio evidencia cómo la IA puede automatizar tareas de programación, depuración, pruebas y documentación, acelerando significativamente el desarrollo de APIs y proyectos web.

### Evidencias:

The screenshot shows the Replit interface during project setup. A modal window titled "Set up program and project files" provides instructions for creating a Python Flask microproject with two endpoints: GET / (returns "API del laboratorio 8 lista") and GET /suma?a=2&b=3 (returns the sum of a + b). It also notes that the user selected the Streamlit stack blueprint instead of Flask. Below the modal, a message states that Flask and Streamlit are different frameworks. A tooltip suggests using the "Effort Fallback" stack. At the bottom, a note mentions creating a plan for a Flask API with REST endpoints.



## Laboratorio 8

### Project Files

**main.py** - Your Flask API with two endpoints:

- **GET /** → Returns "API del laboratorio 8 lista"
- **GET /suma?a=2&b=3** → Returns the sum as JSON (e.g.,  
`{"resultado": 5.0}`)

**requirements.txt** - Lists Flask 3.0.0 as the dependency

**.gitignore** - Configured for Python projects

#### ✓ What's Working

Your API is currently **running** on port 5000! I can see from the logs that:

- The Flask server started successfully
- It's accepting requests
- The root endpoint (`/`) is responding with HTTP 200 OK

#### 💡 Testing Your API

You can test your endpoints:

**1. Health check:** Open the webview to see "API del laboratorio 8 lista"

**2. Sum operation:** Add `/suma?a=2&b=3` to the URL to get:

The screenshot shows a web browser window with a pink header bar. The header bar has tabs for 'Preview' and 'Publishing'. Below the header, the main content area is titled 'Publish your app'. It features a 'Primary URL' input field containing 'flask-sum-api-anamarier .replit.app' with a green checkmark indicating it is available. A blue button at the bottom of this section says 'Upgrade your plan to publish'. Below this, there is a collapsed accordion section with the question 'What does publishing do?'. Inside the accordion, there are two items: 'Publishing your app makes it available for anyone to use' (with a note about friends and users around the world) and 'Costs are included in Core subscriptions' (with a note about varying costs depending on technology). At the bottom right of the accordion, there are 'Watch video' and 'Learn more' buttons.