

# Manual de Ejecución de la Aplicación "SEMINARIO\_FINAL"

## Prerrequisitos

Antes de comenzar, asegúrate de tener instalados:

- [Python 3.x](#)
  - [Node.js y npm \(Node Package Manager\)](#)
  - [Visual Studio Code](#) (opcional, pero recomendado)
- 

## Verificación de Versiones de Prerrequisitos

Antes de proceder con la configuración del proyecto, verifica que tienes las versiones correctas de Python, Node.js y npm instaladas.

### 1. Verificar la versión de Python

Abre una terminal y ejecuta:

- `python --version`
- Python 3.12.4

Asegúrate de tener instalada una versión compatible, como Python 3.x.

### 2. Verificar la versión de Node.js

En la misma terminal, ejecuta:

- `node -v`
- v20.12.2

Confirma que tienes una versión compatible, por ejemplo, v14.x.x o superior.

### 3. Verificar la versión de npm

Ejecuta:

- `npm -v`
- 10.6.0

Verifica que tengas una versión adecuada, como 6.x.x o superior.

**Nota:** Si alguna de estas versiones no está instalada o no es la adecuada, sigue los pasos para instalar o actualizar [Python](#) y [Node.js](#).

---

## 1. Clonar el Repositorio

1. Clona el repositorio desde GitHub:
  - git clone [https://github.com/robertosl77/SEMINARIO\\_FINAL.git](https://github.com/robertosl77/SEMINARIO_FINAL.git)
2. Navega a la carpeta raíz del proyecto:
  - cd seminario-final

---

## 2. Configurar y Ejecutar el Backend

### Activar el Entorno Virtual

1. Abre el proyecto con visual studio code:
  - Deberá observarse la carpeta SEMINARIO\_FINAL
  - Deben observarse las subcarpetas backend y frontend.
2. Abrir un terminal y mantenerse posicionado en la carpeta SEMINARIO\_FINAL.
3. Activa el entorno virtual venv existente:
  - En sistemas Unix/Mac:
    - source backend/venv/bin/actívale
  - En sistemas Windows:
    - .\backend\venv\Scripts\activate

```
● PS E:\Bibliotecas\Documents\GitHub\SEMINARIO_FINAL> .\backend\venv\Scripts\activate
○ (venv) PS E:\Bibliotecas\Documents\GitHub\SEMINARIO_FINAL> █
```

4. ¡Opcional! verifica que las dependencias necesarias ya estén instaladas (si no, usa el siguiente comando):
  - pip install -r requirements.txt
5. En caso de error se puede ejecutar las librerías individualmente:
  - pip install flask
  - pip install flask\_cors

## Ejecutar el Servidor del Backend

1. Desde la carpeta SEMINARIO\_FINAL, ejecuta la aplicación:
  - Python.exe .\backend\app.py

```
(venv) PS E:\Bibliotecas\Documents\GitHub\SEMINARIO_FINAL> python.exe .\backend\app.py
* Serving Flask app 'controladores.Controlador'
* Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit
█
```

---

## 3. Configurar y Ejecutar el Frontend

### Navegar a la Carpeta del Frontend

1. Abre una nueva terminal (manteniendo el backend corriendo) y navega a la carpeta del frontend:
  - En vscode se puede abrir con ctrl+ñ
  - cd ../Frontend

### Instalar Dependencias de React

1. Instala las dependencias usando npm:
  - npm install

### Ejecutar la Aplicación de React

1. Inicia el servidor de desarrollo:
  - npm start

```
● PS E:\Bibliotecas\Documents\GitHub\SEMINARIO_FINAL> cd .\frontend\
○ PS E:\Bibliotecas\Documents\GitHub\SEMINARIO_FINAL\frontend> npm start
```

```
● Compiled successfully!
○
You can now view frontend in the browser.

  Local:            http://localhost:3000
  On Your Network:  http://192.168.1.39:3000

Note that the development build is not optimized.
To create a production build, use npm run build.

webpack compiled successfully
```

---

#### 4. Acceso a la Aplicación

- El backend debería estar corriendo en un puerto específico (por ejemplo, `http://localhost:5000`).
- El frontend normalmente correrá en `http://localhost:3000` (a menos que esté configurado de manera diferente).
- Asegúrate de que ambos servicios estén activos y que el frontend pueda comunicarse con el backend correctamente.

---

#### 5. Consideraciones Adicionales

- Si experimentas errores de conexión entre el frontend y el backend, asegúrate de que las URLs estén configuradas correctamente en las solicitudes API del frontend.
- Si surgen errores de dependencias en el backend, verifica que el entorno virtual `venv` esté correctamente activado y que las dependencias necesarias estén instaladas.

---

#### 6. Usuarios

Para ingresar a la app requerirá de los siguientes usuarios/roles:

- Consulta
  - Usuario: consulta
  - Contraseña: 123
- Operador
  - Usuario: Operador
  - Contraseña: 123
- Admin
  - Usuario: Admin
  - Contraseña: 123