

UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE



Departamento de Ciencias de la Computación

Análisis y Diseño de Software

Manual de Instalación

Versión 1

Elaborado por:

Kevin Asmal

Diego Delgado

Gabriel López

Marcelo Pareja

Ecuador 2026-02-04

Introducción

El presente documento presentará como instalar las dependencias como desarrollador para mantener y probar el sistema en un espacio de desarrollo, además de configuraciones necesarias de las que depende el sistema en el correcto funcionamiento.

Desarrollo

1. Configuración base de datos

Es necesario en la base de datos ser Postgres, sea en la nube o local, con el link interno de acceso que da en postgres el url se compone de:

```
postgresql://usuario:contraseña@servidor/hoptolt_db?sslmode=require
```

El nombre de la base de datos que se genera debe ser exacta para permitir la lectura en los archivos de backend.

2. Instalación de dependencias

Con el comando npm install en las carpetas de /codigo/backend y /codigo/frontend

```
PS C:\Users\gmlop\Desktop\ESPE\Sexto Semestre\Analisis y desarrollo\github\27835-G5-ADS\codigo\backend> npm install
92 packages are looking for funding
  run `npm fund` for details

PS C:\Users\gmlop\Desktop\ESPE\Sexto Semestre\Analisis y desarrollo\github\27835-G5-ADS\codigo\frontend> npm install
removed 1 package, and audited 339 packages in 3s

50 packages are looking for funding
  run `npm fund` for details

found 0 vulnerabilities
```

3. Generación archivo dependencias

Para el correcto funcionamiento de el backend y frontend se necesita generar un archivo .env en /codigo/backend y /codigo/frontend, cada uno con los siguientes apartados:

- **Backend**

- **DATABASE_URL**,el url de conexión a la base de datos de Postgres
- **NODE_ENV**,valor que debe ser development en local o deployment si es en la nube

- **Frontend**

- **VITE_API_BASE_URL**, el url de acceso al backend en un servicio de nube o local para facilitar dentro del proyecto con un servidor habilitado

4. Corrida del servidor

Para levantar los servidores en local o desarrollo, según las configuraciones necesarias explicadas en apartados anteriores e implementar los siguiente comandos, cada uno al funcionar mostrará un mensaje de conexión y proporciona el link de prueba que se puede usar en cualquier navegador

- **Backend**

- node [main.js](#)

```
PS C:\Users\gmllop\Desktop\ESPE\Sexto Semestre\Analisis y desarrollo\Github\27835-G5-ADS\codigo\backend> node main.js
[dotenv@17.2.3] injecting env (2) from .env -- tip: ⚡ specify custom .env file path with { path: '/custom/path/.env'}
[dotenv@17.2.3] injecting env (0) from .env -- tip: ⚡ write to custom object with { processEnv: myObject }
(node:17084) Warning: Accessing non-existent property 'options' of module exports inside circular dependency
(Use `node --trace-warnings ...` to show where the warning was created)
Sistema corriendo en --> 3001
```

- Frontend

- npm run

```
PS C:\Users\gmllop\Desktop\ESPE\Sexto Semestre\Analisis y desarrollo\Github\27835-G5-ADS\codigo\frontend> npm run
> frontend@0.0.0 dev
> vite

9:53:52 [vite] (client) Re-optimizing dependencies because lockfile has changed
Port 5173 is in use, trying another one...

VITE v7.2.7 ready in 1157 ms

→ Local: http://localhost:5174/
→ Network: use --host to expose
→ press h + enter to show help
```