

# Guía paso a paso para levantar el proyecto

---

Esta guía cubre todo lo necesario para ejecutar el proyecto en local:

- Frontend
- Backend
- Base de datos PostgreSQL
- Pruebas con Postman

Incluye requisitos, enlaces de descarga e instalación de dependencias.

---

## 1) Requisitos del sistema

### 1.1 Software obligatorio

1. Git
  - Descarga: <https://git-scm.com/downloads>
2. Node.js LTS (recomendado 20.x o 22.x)
  - Descarga: <https://nodejs.org/en/download>
3. pnpm
  - Guía oficial: <https://pnpm.io/installation>
4. Docker Desktop (recomendado para levantar PostgreSQL fácilmente)
  - Descarga: <https://www.docker.com/products/docker-desktop/>
5. Postman
  - Descarga: <https://www.postman.com/downloads/>

### 1.2 Verificar instalación

Ejecuta en terminal:

```
git --version  
node --version  
npm --version  
pnpm --version  
docker --version  
docker compose version
```

Si algún comando falla, instala ese componente antes de continuar.

---

## 2) Clonar y abrir el proyecto

```
git clone <URL_DEL_REPO>
cd Assessment_ToDo
```

En VS Code abre la carpeta raíz Assessment\_ToDo.

---

## 3) Configurar variables de entorno del backend

En la carpeta backend crea el archivo .env.

Ruta esperada:

- backend/.env

Contenido base sugerido:

```
NODE_ENV=development
PORT=3000

DB_HOST=localhost
DB_PORT=5432
DB_NAME=assessment_todo
DB_USER=todo_user
DB_PASSWORD=todo_pass_123

JWT_SECRET=super_secret_key
JWT_EXPIRES_IN=1d
JWT_SUPERVISOR_INVITE_EXPIRES_IN=15m
```

Notas:

- DB\_PORT debe coincidir con el puerto expuesto por Docker Compose.
  - PORT define el puerto del backend (por defecto 3000).
- 

## 4) Instalar dependencias del proyecto

### 4.1 Backend

```
cd backend
pnpm install
```

### 4.2 Frontend

---

```
cd ../frontend  
pnpm install
```

## 5) Levantar el proyecto (opción recomendada)

Esta opción usa Docker para PostgreSQL y backend juntos.

### 5.1 Iniciar servicios

```
cd ../backend  
pnpm docker:up
```

Qué hace:

- Levanta contenedor PostgreSQL
- Ejecuta scripts SQL de backend/sql automáticamente
- Construye y levanta el backend

### 5.2 Ver logs (si quieres monitorear)

```
pnpm docker:logs
```

### 5.3 Verificar backend

Abre en navegador o Postman:

- <http://localhost:3000/health>

Respuesta esperada:

```
{  
  "status": "ok",  
  "db": "connected"  
}
```

### 5.4 Levantar frontend

En otra terminal:

```
cd ../frontend  
pnpm dev
```

URL esperada:

- <http://localhost:5173>
- 

## 6) Puertos y URLs del proyecto

- Backend API: <http://localhost:3000>
  - Healthcheck: <http://localhost:3000/health>
  - Frontend: <http://localhost:5173>
- 

## 7) Probar APIs con Postman

Referencia de pruebas y endpoints:

- Documentacion/Pruebas\_postman.md

Flujo mínimo recomendado:

1. Solicitar token supervisor
  2. Registrar supervisor
  3. Registrar standard
  4. Login/me
  5. Crear y aprobar solicitudes de tareas
- 

## 8) Solución de problemas

### 8.1 Error de conexión a BD

- Revisa que Docker Desktop esté encendido.
- Ejecuta:

```
docker ps
```

- Verifica que assessment\_postgres esté corriendo.
- Confirma DB\_HOST, DB\_PORT, DB\_NAME, DB\_USER, DB\_PASSWORD en backend/.env.

### 8.2 El backend no inicia por variables faltantes

El backend valida variables requeridas. Si falta una, se detiene al iniciar.

Verifica que existan al menos:

- DB\_HOST
  - DB\_PORT
-

- DB\_NAME
- DB\_USER
- DB\_PASSWORD
- JWT\_SECRET

## 8.3 El frontend no conecta con backend

- Asegúrate de que backend esté en http://localhost:3000.
- Revisa CORS y que el backend responda /health.

## 8.4 Limpiar y reconstruir contenedores

```
cd backend
pnpm docker:down
docker volume rm assessment_todo_postgres_data
pnpm docker:up
```

Si el nombre del volumen difiere, consulta primero:

```
docker volume ls
```