# Prueba Técnica

Guía de Instalación, Configuración y Uso del Sistema

## Requerimientos Previos

1. Tener Docker instalado.

2. Tener Node.js, npm y NestJS CLI instalados.

**npm install -g @nestjs/cli**

3. Contar con SSIS para la carga del CSV (si vas a usar la parte ETL).

4. La base de datos Oracle se levantará con Docker.

## Levantamiento de la Base de Datos

Para levantar la base de datos, asegúrate de tener Docker instalado en tu máquina. Luego, ejecuta el siguiente comando en la terminal:

**docker-compose up -d**

Esto iniciará un contenedor con Oracle y creará la base de datos necesaria para la aplicación.

## Scripts

Dentro del archivo `scripts.sql` se encuentra un script que trae la productividad por ejecutivo.

## Configuración del Proyecto de NestJS

Antes de ejecutar la aplicación, asegúrate de configurar las variables de entorno necesarias. Crea un archivo `. env` en la raíz del proyecto con el siguiente contenido:

ORACLE\_HOST=localhost  
ORACLE\_PORT=1521  
ORACLE\_USER=crm\_user  
ORACLE\_PASSWORD=crm123  
ORACLE\_SERVICE\_NAME=XEPDB1

## Ejecución de la Aplicación

### Instalación de Dependencias

**npm install**

### Inicio de la Aplicación

**npm run start:dev**

Esto iniciará el servidor en modo de desarrollo. La aplicación estará disponible en http://localhost:3000.

## Endpoints Disponibles

• GET /api/porductividad/all — Obtiene la productividad de todos los Ejecutivos.

• GET /api/porductividad/:ejecutivo — Obtiene la productividad de un Ejecutivo específico.

## Modo de Uso del SSIS

Una vez abierto el proyecto de SSIS, cargar el archivo "carga ventas.siss" y usar las siguientes credenciales:

• Usuario: crm\_user

• Contraseña: crm123

Luego, ajustar la ruta del archivo CSV y ejecutar el proceso para cargar los datos en la base de datos Oracle.

### Funcionamiento del SSIS

1. Limpieza de la tabla VENTAS\_STAGING: elimina registros existentes para evitar duplicados.

2. Lectura del archivo CSV: lee el archivo desde la ruta especificada, extrayendo los datos de ventas.

3. Validación de datos: se validan los datos mediante un Script Component.

4. Inserción en la tabla STAGING: inserta los datos validados en VENTAS\_STAGING y registra errores en VENTAS\_ERRORES.

5. Inserción en la tabla final: los datos correctos se transfieren desde la tabla de staging a la tabla final VENTAS.