

# PROYECTO SQL ORACLE

---

TRABAJO CON BASES DE DATOS

GIT CLONE [HTTPS://GITHUB.COM/JVIEJO/CURSO-DBS-14.GIT](https://github.com/jviejo/curso-dbs-14.git)

```
docker run -d -p 1521:1521 --name curso-oracle-14  
store/oracle/database-enterprise:12.2.0.1-slim
```

USUARIO:SYS AS SYSDBA password: Oradoc\_db1

carga de datos oracle

[https://raw.githubusercontent.com/dshifflet/NorthwindOracle\\_DDL/master/northwind\\_export.sql](https://raw.githubusercontent.com/dshifflet/NorthwindOracle_DDL/master/northwind_export.sql)

Configuración de la conexión "ORCLCDB.localdomain"

**Ajustes de conexión**

Oracle ajustes de conexión

ORACLE

▼ Ajustes de conexión

- Inicialización
- Comandos de s
- Identificación d
- Transacciones
- General
- Metadatos
- Errores y timeouts
- > Editor de datos
- > Editor SQL

General Oracle properties Driver properties SSH Proxy

Connection Type:

Basic TNS Custom

Host: localhost Port: 1521

Database: ORCLCDB.localdomain Service Name

Authentication

Authentication: Oracle Database Native

Nombre de usuario: SYS Role: SYSDBA

Contraseña: ●●●●●●●●  Save password loca

Client:

ⓘ You can use variables in connection parameters.

Driver name: Oracle Edit Driver Settings

Probar conexión ... Aceptar Cancelar

HOST: localhost  
SERVICE: ORCLCDB.localdomain  
PORT: 1521  
Usuario: SYS  
Role: SYSDBA  
Contraseña: Oradoc\_db1

carga de datos oracle

[https://raw.githubusercontent.com/dshifflet/NorthwindOracle\\_DDL/master/northwind\\_export.sql](https://raw.githubusercontent.com/dshifflet/NorthwindOracle_DDL/master/northwind_export.sql)

HAY QUE INSTALAR EL CLIENTE DE ORACLE

<https://www.oracle.com/database/technologies/instant-client/winx64-64-downloads.html>

```
const oracledb = require('oracledb');

var pool = null
try {
  oracledb.initOracleClient({ libDir: 'C:\\Users\\jviejo\\Downloads\\instantclient-
basiclite-windows.x64-21.3.0.0.0\\instantclient_21_3' });
} catch (err) {
  console.error('Whoops!');
  console.error(err);
  process.exit(1);
}
```

```
async function getPool(con) {
  return new Promise(async (resolve, reject) => {
    if (pool) resolve(pool)
    try {
      console.log("obtengo pool")
      pool = await oracledb.createPool(con)
      resolve(pool)
    } catch (error) {
      reject(error)
    }
  });
}
```

```
async function q(sql, parametros) {
  let connection;
  try {
    await getPool({
      user: process.env.ORACLE_USER, password: process.env.ORACLE_PASSWORD,
      connectString: process.env.ORACLE_CN, poolAlias: "curso"
    })
    connection = await oracledb.getConnection("curso");
    const result = await connection.execute(
      sql,
      parametros, { outFormat: oracledb.OBJECT},
    );
    return (result.rows);
  } catch (err) {
    return err;
  } finally {
    if (connection) {
      try {
        await connection.close();
      } catch (err) {
        return err;
      }
    }
  }
}
```