**PRUEBA TÉCNICA**

**HIBERUS**

**DOCUMENTACIÓN**

**GESTIÓN DE EMPLEADOS**

[**https://github.com/nataly-morales12/hiberus.git**](https://github.com/nataly-morales12/hiberus.git)

**BY**

**NATALY MORALES**

Contenido

[ARQUITECTURA 1](#_Toc198043226)

[BACK-END 1](#_Toc198043227)

[FRONT-END 1](#_Toc198043228)

[BASE DE DATOS (BDD) 1](#_Toc198043229)

[DISEÑO DE BASE DE DATOS 2](#_Toc198043230)

[CREACCIÓN DE DIAGRAMAS: MODELO CONCEPTUAL 2](#_Toc198043231)

[CREACCIÓN DE DIAGRAMAS: MODELO LÓGICO 3](#_Toc198043232)

[CREACCIÓN DE DIAGRAMAS: MODELO FÍSICO 3](#_Toc198043233)

[APLICACIÓN SERVIDOR 3](#_Toc198043234)

[DESARROLLO 3](#_Toc198043235)

[EJECUCIÓN 4](#_Toc198043236)

[APLICACIÓN CLIENTE 6](#_Toc198043237)

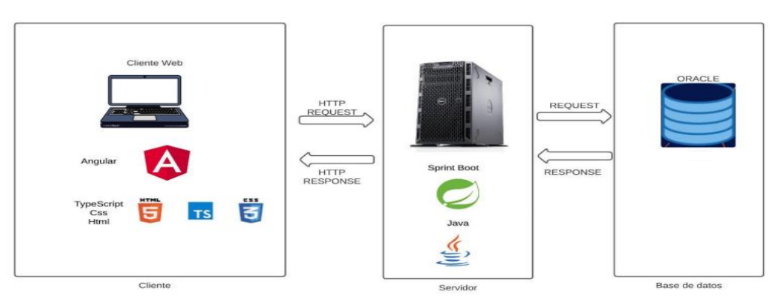
[DESARROLLO 7](#_Toc198043238)

[EJECUCIÓN 7](#_Toc198043239)

# ARQUITECTURA

La arquitectura permite a los profesionales conocer de forma temprana a estructura del software, de tal manera que puede ayudar a identificar algún cambio técnico que pudo no se considerado anteriormente.

Dentro de la arquitectura selecciona y además acoplándose al ejercicio plateado se sigue una arquitectura **Cliente-Servidor**, y como referencia se tiene la siguiente imagen donde se mencionan, además, las tecnologías a utilizarse.



## BACK-END

|  |  |  |
| --- | --- | --- |
| **Tecnología** | **Nombre** | **Versión** |
| **Lenguaje de programación** | Java | 8 o superior |
| **Framework** | Sprint Boot | 2 o superior |
| **Entorno de desarrollo** | IntelliJ IDEA |  |

Para la comunicación entre el cliente y servidor se procedió a implementar **API Rest**, pues permiten enviar y recibir información en ambos sentidos a través del protocolo http.

## FRONT-END

|  |  |  |
| --- | --- | --- |
| **Tecnología** | **Nombre** | **Versión** |
| **Lenguaje de programación** | TypeScript | cualquiera |
| **Framework** | Angular | 14 o superior |
| **Entorno de desarrollo** | Visual Studio Code |  |

## BASE DE DATOS (BDD)

|  |  |  |
| --- | --- | --- |
|  | **Nombre** | **Versión** |
| **BDD** | Oracle | Cualquiera (en el ejemplo 21 c express) |
| **Tipo BDD** | Relacional |  |
| **Lenguaje** | SQL |  |

# DISEÑO DE BASE DE DATOS

Empleado

|  |
| --- |
| **Atributo** |
| employee\_id |
| employee\_last\_name |
| age |
| salary |
| init\_date |
| end\_date |
| employee\_status |

Departamento

|  |
| --- |
| **Atributo** |
| departament\_id |
| departamento\_name |
| departament\_status |

## CREACCIÓN DE DIAGRAMAS: MODELO CONCEPTUAL

Diagrama

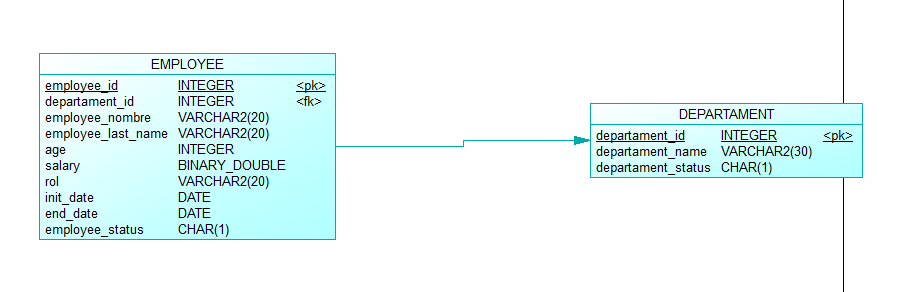
Descripción generada automáticamente

## CREACCIÓN DE DIAGRAMAS: MODELO LÓGICO

Diagrama

Descripción generada automáticamente

## CREACCIÓN DE DIAGRAMAS: MODELO FÍSICO



# APLICACIÓN SERVIDOR

## DESARROLLO

Creación de proyecto mediante la ayuda de la siguiente página, donde se genera el proyecto Java con Sprint Boot. Una alternativa fácil y rápida para crear este tipo de proyectos donde se pueden seleccionar las librerías o dependencias a utilizarse.

<https://start.spring.io/>

Captura de pantalla de computadora

Descripción generada automáticamente

Las dependencias utilizadas son las siguientes:

1. Oracle Driver
2. Spring Data JPA
3. Spring Web
4. Lombok

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Para finalizar el proyecto es generado y se descarga en la ruta que desea colocar el mismo.

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

## EJECUCIÓN

1. Clonar repositorio:
2. Crear usuario, con la siguiente sentencia:

--CREACION DE USUARIO

alter session set "\_ORACLE\_SCRIPT"=true;

-- Crear el usuario hiberus

CREATE USER hiberus IDENTIFIED BY hiberus;

-- Otorgar privilegios de DBA (equivalente a "root" en otros sistemas)

GRANT DBA TO hiberus;

-- Otorgar privilegios adicionales segun sea necesario

-- Por ejemplo:

GRANT CREATE SESSION TO hiberus;

GRANT CREATE TABLE TO hiberus;

GRANT CREATE VIEW TO hiberus;

GRANT CREATE PROCEDURE TO hiberus;

GRANT CREATE TRIGGER TO hiberus;

GRANT CREATE SEQUENCE TO hiberus;

GRANT CREATE TYPE TO hiberus;

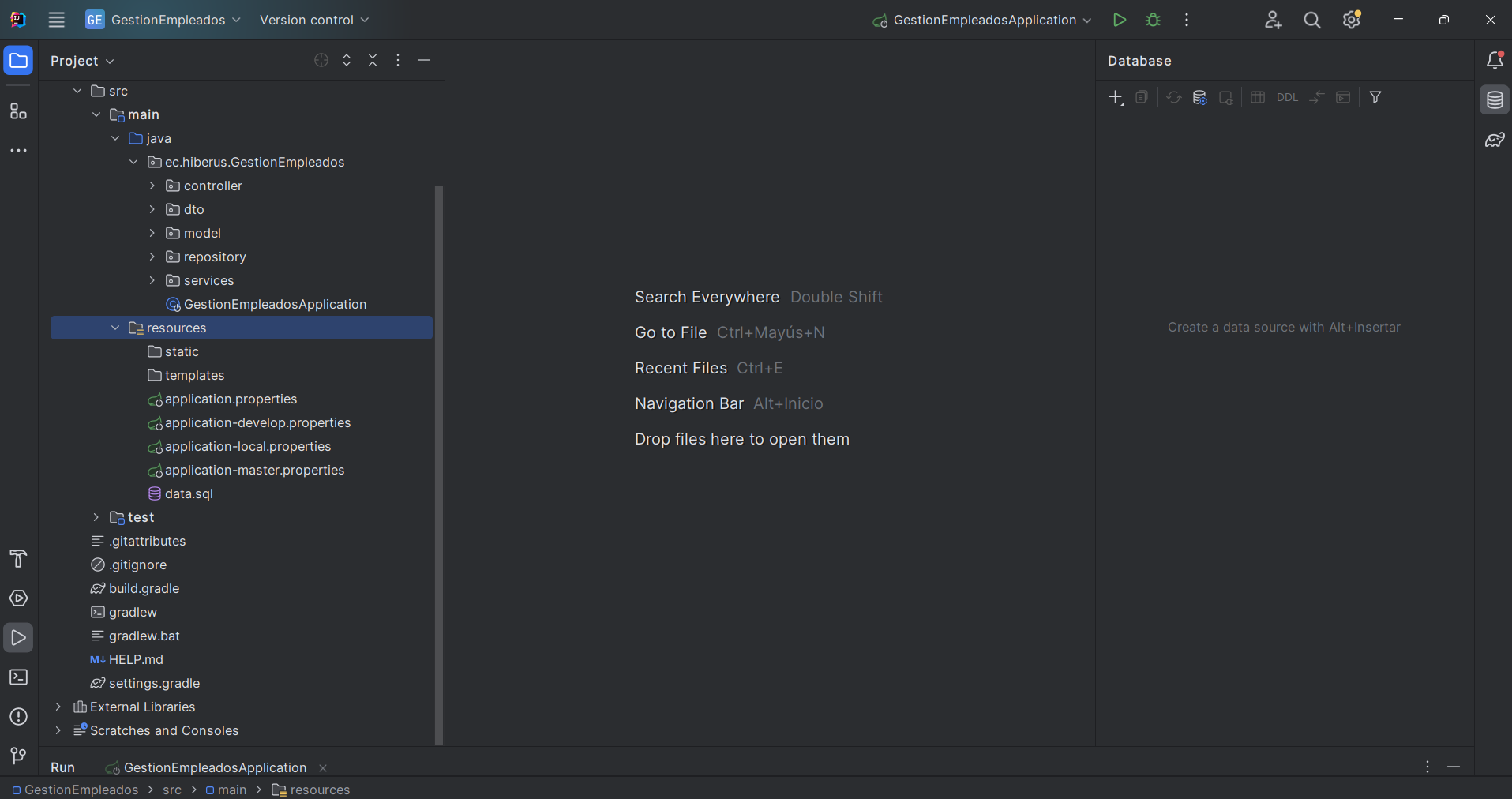
GRANT CREATE ANY DIRECTORY TO hiberus;

-- Y otros privilegios según los requisitos de tu aplicación

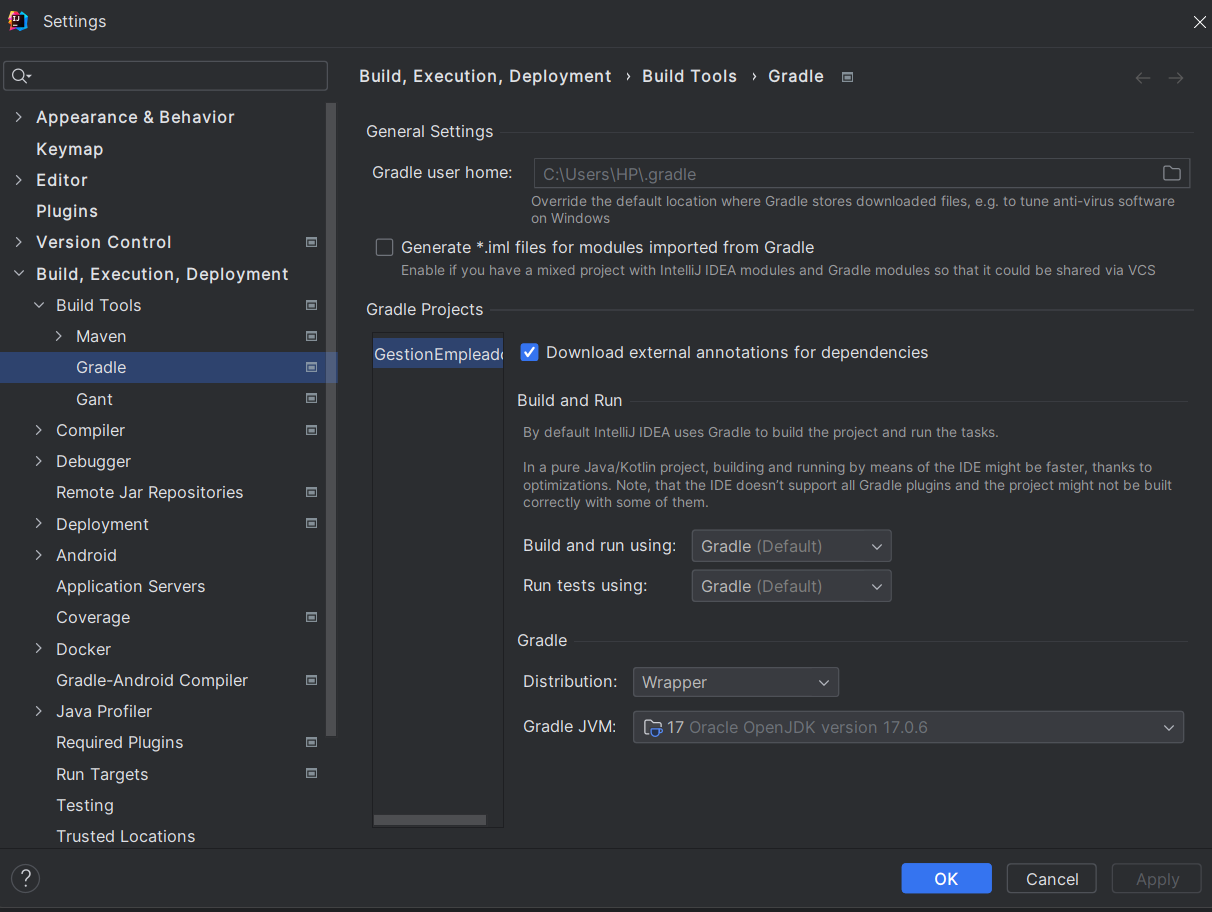
-- Finalmente, confirmar los cambios

COMMIT;

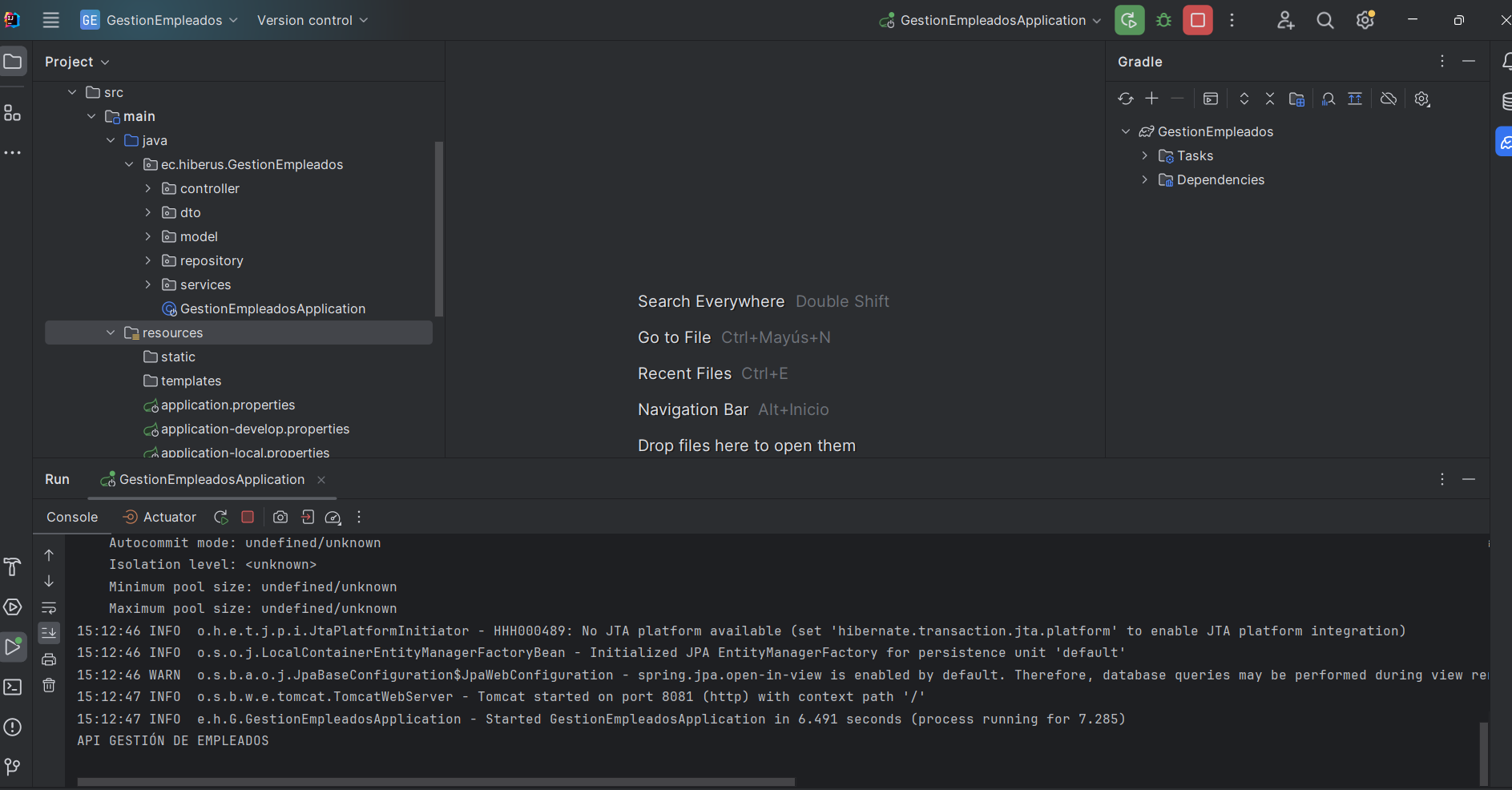
1. Crear base de datos en su motor de bdd Oracle. El documento se encuentra en la dirección 03. BDD/02. SCRIPT/01. CREACION/BDDV1.0.sql. Ejecutar el script en el usuario creado.
2. Abrir el proyecto en el entorno de desarrollo de preferencia, en este caso se usa IntelliJ IDEA.



1. Verificar que dentro de las configuraciones de Grade se este ejecutando con Java 17. En caso de no estar, seleccionar Java 17 y sincronizar las propiedades de gradle.



1. Habiéndose verificado, lo siguiente es ejecutar la aplicación. En la consola se vera la siguiente imagen.



# APLICACIÓN CLIENTE

Crear una aplicación con angular es fácil, lo importante es considerar tener instalado previamente node y angular.

Lo siguiente es ejecutar el siguiente comando en la ruta de preferencia: ng new nombre-proyecto. La única librería instalada fue Bootstrap mediante el comando npm i Bootstrap.

## DESARROLLO

Una vez ejecutado el comando mencionado anteriormente la aplicación se vera de la siguiente manera:  
Texto

Descripción generada automáticamente

A la misma se agregar las páginas, configuraciones y componentes necesarios. La carpeta donde se guardarán las páginas, modelos y servicios será módulos/main.

Texto

Descripción generada automáticamente con confianza media

## EJECUCIÓN

1. Clonar repositorio
2. Instalar módulos, para lo cual abrir una terminal en la raíz del proyecto y ejecutar npm i.
3. Ejecutar en la terminar ng serve y la aplicación se empezará a ejecutar en la dirección: localhost:4200.

Interfaz de usuario gráfica, Tabla

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación, Word

Descripción generada automáticamente