

## Contenido

<b>Instalación de NODO JS y NPM- Frontend .....</b>	<b>3</b>
Instalación de angular cli.....	5
Creación de proyecto de angular. ....	6
Instalación y configuración de Spring Boot – Java .....	8
Instalación de OpenJDK11.....	8
Creación de proyecto en Spring Initializr .....	11
Creación de repositorio en GIT. ....	12
Configuración de repositorio en equipo de desarrollo. ....	13
Clonación de repositorio. ....	15
Commid de carga de archivos .....	16
Instalación de MYSQL Workbench .....	18
Creación de usuario a nivel BD.....	23
Administrador .....	23
SoporteN2 .....	23
Almacén.....	23
Backend APIS REST .....	24
API Almacén .....	25
API Cliente .....	26
API Cuenta .....	26
API Estados Republica .....	27
API Historia de Asignación .....	27
API Hstoria de Ticket .....	27
API Perfil .....	27
API Servicios .....	27
API SLA Tikect.....	28
API Solicitud a Almacen.....	28
API Estatus Ticket .....	28
API Ticket.....	28
API Usuario .....	28
API Zona Estado.....	28



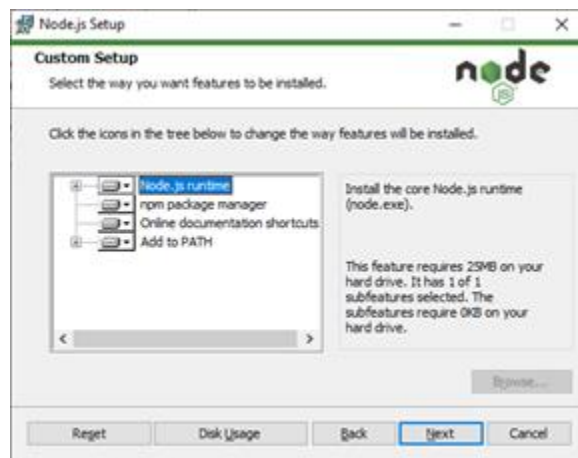
## Instalación de NODO JS y NPM- Frontend

A continuación, los 5 pasos para instalar Node.js y actualizar NPM:

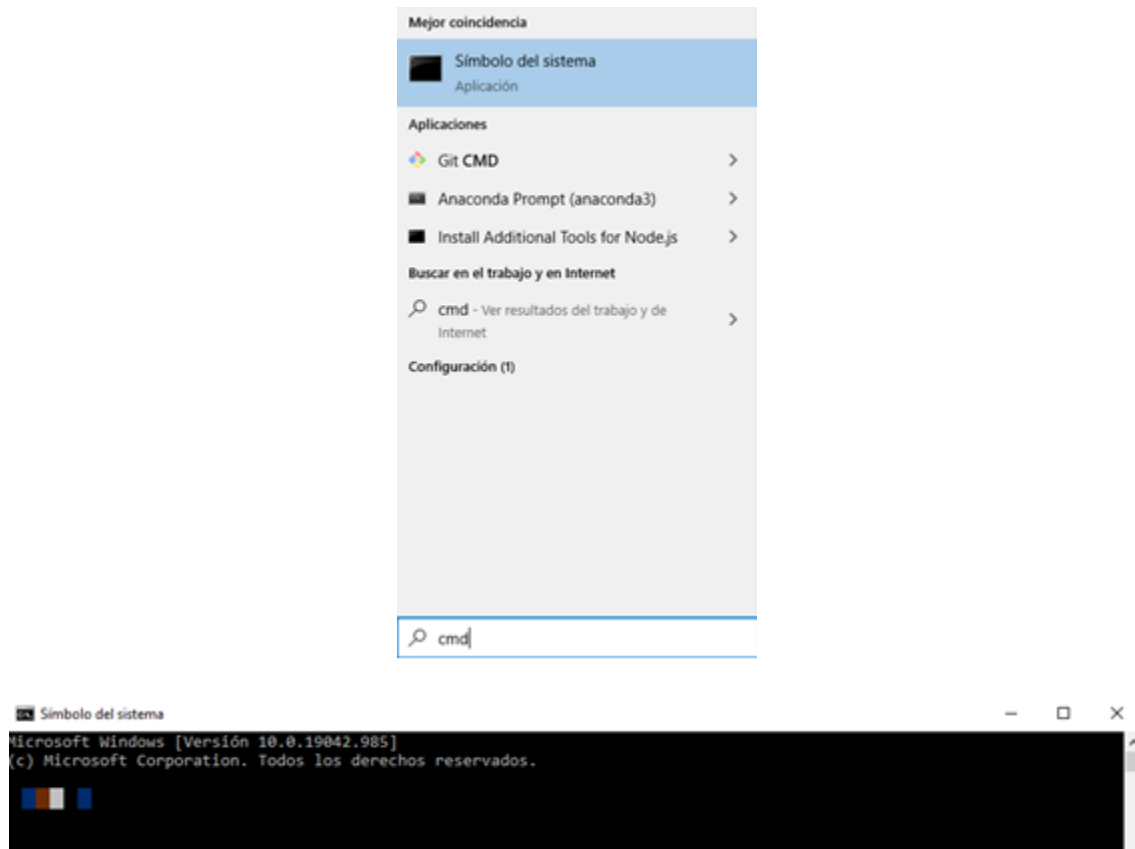
1. Entrar en <https://nodejs.org/es/download/> y descargar el instalador de Node.js en el sistema operativo deseado. Podemos elegir entre Windows, Mac y Linux.



2. Ejecutar el instalador que acabamos de descargar. Simplemente debemos avanzar en el proceso de instalación.



3. Una vez finalizado el proceso de instalación, podemos comprobar fácilmente si se nos ha instalado correctamente. Para ello, vamos al intérprete de comandos de nuestro ordenador (en Windows, por ejemplo, escribir "cmd" en la barra de búsqueda y abrir la aplicación de "Símbolo del sistema").



4. En la ventana de comandos, escribir `node -v` y pulsar la tecla *Enter*. Nos debería aparecer la versión que tenemos instalada de Node.js (en mi caso la versión 16.14.0).

```
C:\Users\BID>node -v
v16.14.0
C:\Users\BID>
```

5. Para comprobar que se nos ha instalado también NPM, escribiremos `npm -v` y pulsaremos de nuevo *Enter*. Nos debería aparecer también en este caso la versión del *Node Package Manager* (en mi caso la versión 8.5.0).

```
C:\Users\BID>npm -v
8.5.0
C:\Users\BID>
```

## Instalación de angular cli

Una vez finalizada la instalación de NODE.js y NPM, se procederá a instalar angular CLI, este componente ayudara a generar con mayor eficiencia el desarrollo del Frontend del desarrollo.

1. Se desinstala la versión que tiene NPM de angular cli.

```
C:\Users\BID>npm uninstall -g angular/cli
up to date in 662ms
```

2. Se borrará cache de la NPM, para evitar errores de instalación con la nueva versión que se descara.

```
C:\Users\BID>npm cache clean --force
npm WARN using --force Recommended protections disabled.
```

3. Se procederá a la instalación de la versión más reciente de angular-cli.

```
C:\Users\BID>npm install -g @angular/cli@latest
[Progress Bar] \ idealTree:inquirer: timing idealTree:node_modules/@angular/cli/node_modules/inquirer Completed i
```

## Creación de proyecto de angular.

1. Se realiza la apertura de una línea de comando (CMD), el path donde se vaya a realizar el proyecto.

```
C:\Users\BID\Pictures\Proyecto> cd C:\Users\BID\Pictures\Proyecto
C:\Users\BID\Pictures\Proyecto>
```

2. Con el comando “**ng new**” se creará el proyecto con las librerías y dependencias necesarias para comenzar el desarrollo del frontend

```
C:\Users\BID\Pictures\Proyecto>ng new
? What name would you like to use for the new workspace and initial project? Helpdesk_F
? Would you like to add Angular routing? No
? Which stylesheet format would you like to use? CSS
CREATE Helpdesk_F/angular.json (3063 bytes)
CREATE Helpdesk_F/package.json (1073 bytes)
CREATE Helpdesk_F/README.md (1055 bytes)
CREATE Helpdesk_F/tsconfig.json (863 bytes)
CREATE Helpdesk_F/.editorconfig (274 bytes)
CREATE Helpdesk_F/.gitignore (548 bytes)
CREATE Helpdesk_F/.browserslistrc (600 bytes)
CREATE Helpdesk_F/karma.conf.js (1427 bytes)
CREATE Helpdesk_F/tsconfig.app.json (287 bytes)
CREATE Helpdesk_F/tsconfig.spec.json (333 bytes)
CREATE Helpdesk_F/.vscode/extensions.json (130 bytes)
CREATE Helpdesk_F/.vscode/launch.json (474 bytes)
CREATE Helpdesk_F/.vscode/tasks.json (938 bytes)
CREATE Helpdesk_F/src/favicon.ico (948 bytes)
CREATE Helpdesk_F/src/index.html (295 bytes)
CREATE Helpdesk_F/src/main.ts (372 bytes)
CREATE Helpdesk_F/src/polyfills.ts (2338 bytes)
CREATE Helpdesk_F/src/styles.css (80 bytes)
CREATE Helpdesk_F/src/test.ts (745 bytes)
CREATE Helpdesk_F/src/assets/.gitkeep (0 bytes)
CREATE Helpdesk_F/src/environments/environment.prod.ts (51 bytes)
CREATE Helpdesk_F/src/environments/environment.ts (658 bytes)
CREATE Helpdesk_F/src/app/app.module.ts (314 bytes)
CREATE Helpdesk_F/src/app/app.component.html (23332 bytes)
CREATE Helpdesk_F/src/app/app.component.spec.ts (968 bytes)
CREATE Helpdesk_F/src/app/app.component.ts (214 bytes)
CREATE Helpdesk_F/src/app/app.component.css (0 bytes)
√ Packages installed successfully.
warning: LF will be replaced by CRLF in .browserslistrc.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in .editorconfig.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in .gitignore.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in .vscode/extensions.json.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in .vscode/launch.json.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in .vscode/tasks.json.
The file will have its original line endings in your working directory
```

3. Terminando de la creación del proyecto se deberá de mostrar la siguiente salida de la consola.

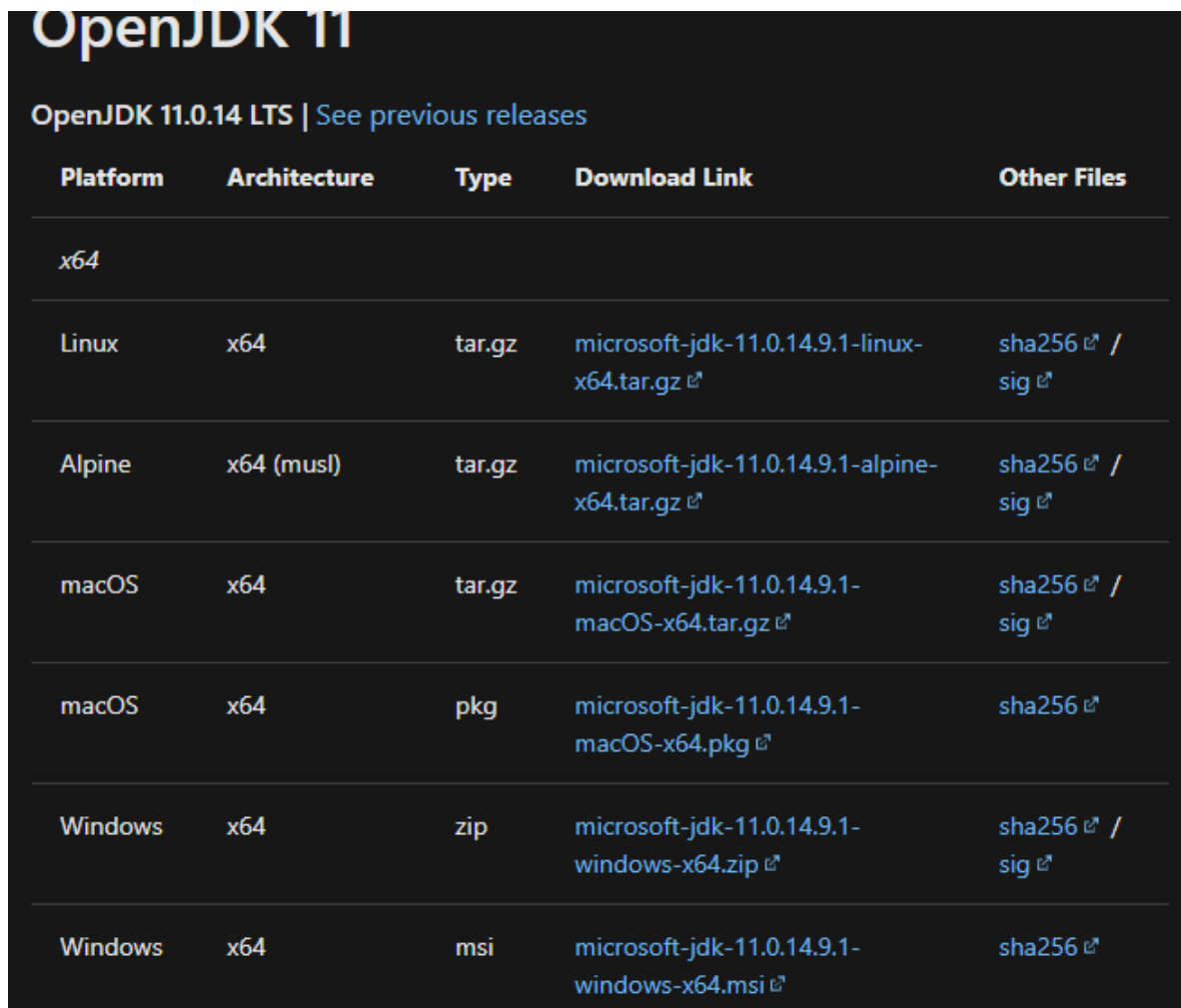
```
warning: LF will be replaced by CRLF in src/test.ts.  
The file will have its original line endings in your working directory  
warning: LF will be replaced by CRLF in tsconfig.app.json.  
The file will have its original line endings in your working directory  
warning: LF will be replaced by CRLF in tsconfig.json.  
The file will have its original line endings in your working directory  
warning: LF will be replaced by CRLF in tsconfig.spec.json.  
The file will have its original line endings in your working directory  
Author identity unknown  
  
*** Please tell me who you are.  
  
Run  
  
  git config --global user.email "you@example.com"  
  git config --global user.name "Your Name"  
  
to set your account's default identity.  
Omit --global to set the identity only in this repository.  
  
fatal: unable to auto-detect email address (got 'BID@DESKTOP-QBDB8BL.(none)')  
C:\Users\BID\Pictures\Proyecto>
```

## Instalación y configuración de Spring Boot – Java

A continuación se mostrarán los pasos a seguir para la instalación de los elementos necesarios del Backend del proyecto.

### Instalación de OpenJDK11

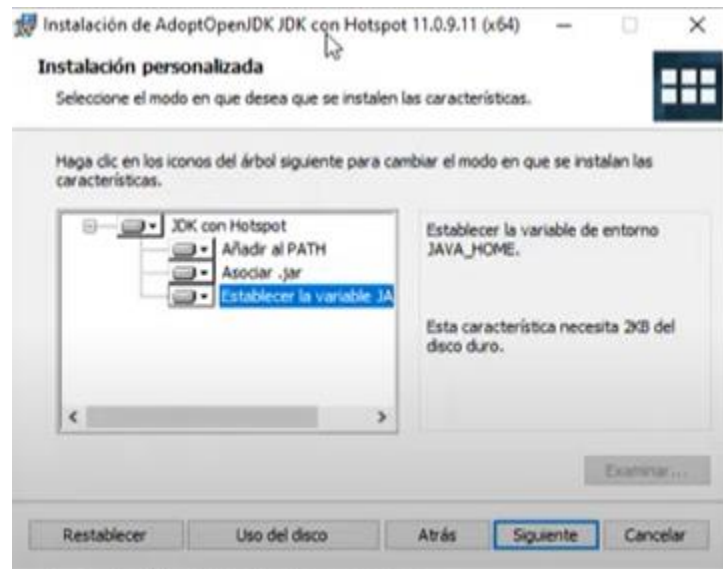
1. Entrar en <https://docs.microsoft.com/en-us/java/openjdk/download> y descargar el instalador de Node.js en el sistema operativo deseado. Podemos elegir entre Windows, Mac y Linux.



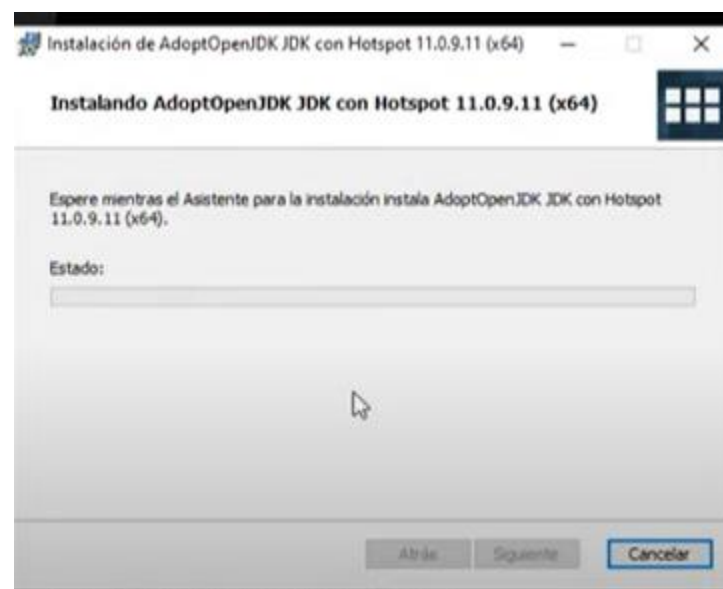
OpenJDK 11				
OpenJDK 11.0.14 LTS   <a href="#">See previous releases</a>				
Platform	Architecture	Type	Download Link	Other Files
x64				
Linux	x64	tar.gz	<a href="#">microsoft-jdk-11.0.14.9.1-linux-x64.tar.gz</a>	<a href="#">sha256</a> / <a href="#">sig</a>
Alpine	x64 (musl)	tar.gz	<a href="#">microsoft-jdk-11.0.14.9.1-alpine-x64.tar.gz</a>	<a href="#">sha256</a> / <a href="#">sig</a>
macOS	x64	tar.gz	<a href="#">microsoft-jdk-11.0.14.9.1-macos-x64.tar.gz</a>	<a href="#">sha256</a> / <a href="#">sig</a>
macOS	x64	pkg	<a href="#">microsoft-jdk-11.0.14.9.1-macos-x64.pkg</a>	<a href="#">sha256</a>
Windows	x64	zip	<a href="#">microsoft-jdk-11.0.14.9.1-windows-x64.zip</a>	<a href="#">sha256</a> / <a href="#">sig</a>
Windows	x64	msi	<a href="#">microsoft-jdk-11.0.14.9.1-windows-x64.msi</a>	<a href="#">sha256</a>

2. Ejecutar el instalador de openjdk.exe, y dar en siguiente

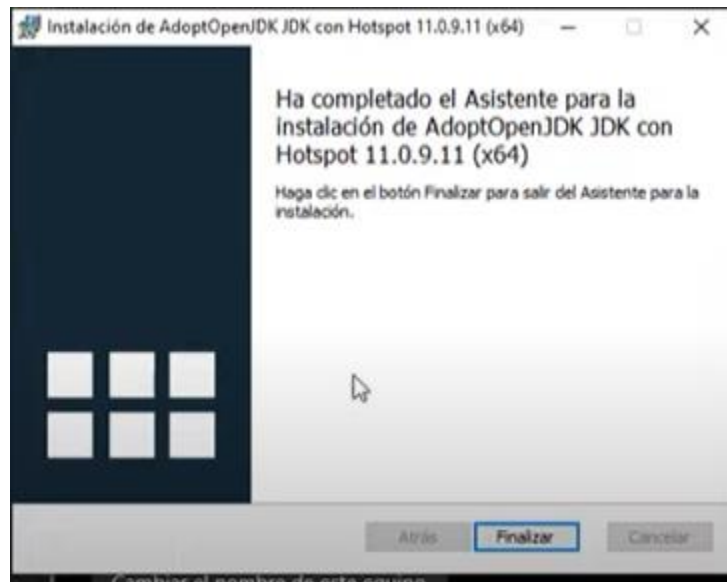




3. Daremos en instalar y comenzara el proceso de instalación



4. Cuando el proceso de instalación finalice, daremos en finalizar



5. Para verificar la versión de instalación, abriremos una línea de comandos "cmd", donde se ingresará el comando. "java --version"

```
C:\Users\BID\Pictures\Proyecto>java --version
openjdk 11.0.14 2022-01-18 LTS
OpenJDK Runtime Environment Microsoft-30257 (build 11.0.14+9-LTS)
OpenJDK 64-Bit Server VM Microsoft-30257 (build 11.0.14+9-LTS, mixed mode)

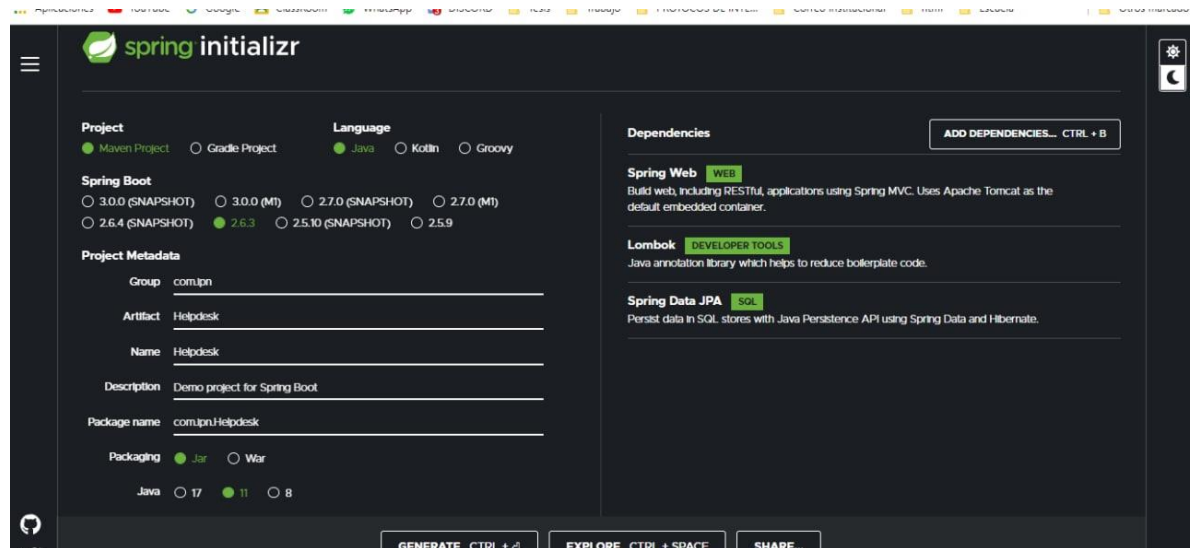
C:\Users\BID\Pictures\Proyecto>
```

## Creación de proyecto en Spring Initializr

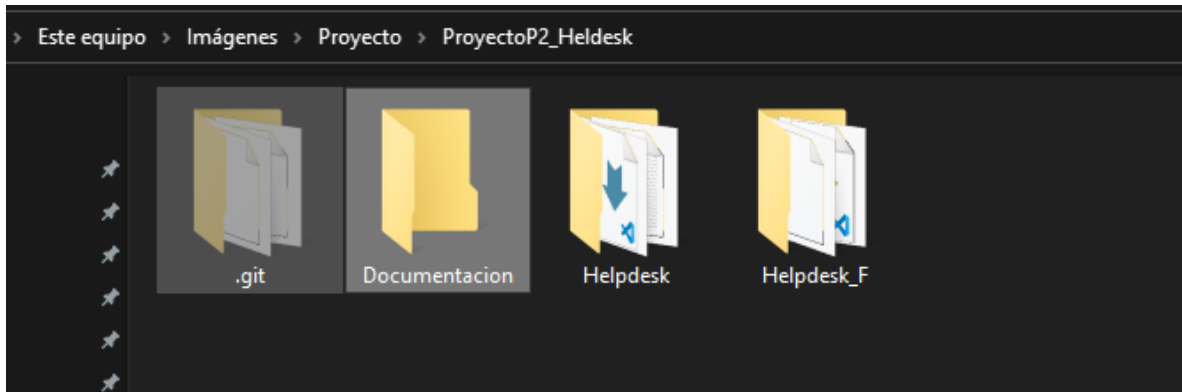
Para la creación del template de desarrollo de spring boot se realizara en spring Initializr, ingresar a la siguiente pagina <https://start.spring.io/>

1. En proyecto se seleccionará "Maven Project"
2. En lenguaje se seleccionará "Java"
3. En Spring Boot se seleccionará "2.6.4"
4. En Project Metadata se capturara:
  - a. Group: **com.ipn**
  - b. Artifact: **Helpdesk**
  - c. Name: **Helpdesk**
  - d. Package name: **com.ipn.heldesk**
5. En Packaging se seleccionará "Jar"
6. En Java se seleccionará "11"
7. En Dependencies se agregarán las dependencias
  - a. Spring web
  - b. Lombok
  - c. Spring Data JPA

Dicha configuración se muestra en la imagen siguiente.



Se seleccionara la opción de "Generate", la cual comenzara a generar el proyecto y se descara en un .zip, el cual descomprimiremos en nuestro directorio de trabajo .



### Creación de repositorio en GIT.

Se ingresara a la página <https://github.com/> con las credenciales ya conocidas.



Ingresar al aparato de nuevo repositorio “ <https://github.com/new> ”, donde se captura el nombre del repositorio, el cual por nomenclatura se eligio “**ProyectoP2\_Heldesk**”.

## Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

---


**Owner \*** **Repository name \***


 RicharFI / ProyectoP2\_Heldesk 

Great repository names are short and memorable. Need inspiration? How about [special-rotary-phone?](#)

**Description** (optional)

---

☒  **Public**  
Anyone on the internet can see this repository. You choose who can commit.

☐  **Private**  
You choose who can see and commit to this repository.

---



**Initialize this repository with:**  
Skip this step if you're importing an existing repository.

☒ **Add a README file**  
This is where you can write a long description for your project. [Learn more.](#)

☒ **Add .gitignore**  
Choose which files not to track from a list of templates. [Learn more.](#)

El cual al crear tendrá la salida siguiente


**Quick setup — if you've done this kind of thing before**

 Set up in Desktop or **HTTPS** **SSH** `https://github.com/RicharFI/ProyectoP2_Heldesk.git` 

Get started by [creating a new file](#) or [uploading an existing file](#). We recommend every repository include a [README](#), [LICENSE](#), and [.gitignore](#).


**...or create a new repository on the command line**

```
echo "# ProyectoP2_Heldesk" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/RicharFI/ProyectoP2_Heldesk.git
git push -u origin main
```



**...or push an existing repository from the command line**

```
git remote add origin https://github.com/RicharFI/ProyectoP2_Heldesk.git
git branch -M main
git push -u origin main
```



**...or import code from another repository**  
You can initialize this repository with code from a Subversion, Mercurial, or TFS project.

[Import code](#)

Configuración de repositorio en equipo de desarrollo.

Se deberá tener previamente instalado git, una vez instalado dicho software se proseguirá de la siguiente manera.

Se configurará el name y email de usuario, esto se realizara bajo los comandos.

```
$ git config --global user.name "RicharFl"  
$ git config --global user.email ricardo\_6\_11@hotmail.com
```

```
BID@DESKTOP-QBDB88L MINGW64 ~/Pictures/Proyecto  
$ git config --global user.name "RicharFl"  
  
BID@DESKTOP-QBDB88L MINGW64 ~/Pictures/Proyecto  
$ git config --global user.email ricardo_6_11@hotmail.com
```

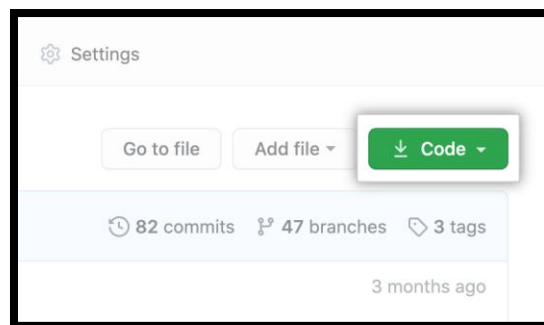
Con el comando “git config –list” se verifica la configuración.

```
BID@DESKTOP-QBDB88L MINGW64 ~/Pictures/Proyecto  
$ git config --list  
diff.astextplain.textconv=astextplain  
filter.lfs.clean=git-lfs clean -- %f  
filter.lfs.smudge=git-lfs smudge -- %f  
filter.lfs.process=git-lfs filter-process  
filter.lfs.required=true  
http.sslbackend=openssl  
http.sslcainfo=C:/Program Files/Git/mingw64/ssl/certs/ca-bundle.crt  
core.autocrlf=true  
core.fscache=true  
core.symlinks=false  
pull.rebase=false  
credential.helper=manager-core  
credential.https://dev.azure.com.usehttppath=true  
init.defaultbranch=master  
user.name=RicharFl  
user.email=ricardo_6_11@hotmail.com  
core.editor=emacs
```

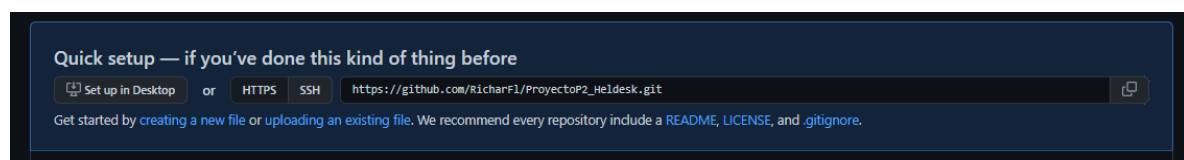
## Clonación de repositorio.

1. En GitHub.com, visita la página principal del repositorio.

Sobre la lista de archivos, da clic en Código.



2. Para clonar el repositorio utilizando HTTPS, debajo de "Clonar con HTTPS", da clic en code. Para clonar el repositorio utilizando una llave SSH, incluyendo un certificado emitido por la autoridad de certificados SSH de tu organización, haz clic en Utilizar SSH y luego en . Para clonar un repositorio utilizando el CLI de GitHub, haz clic en Utilizar el CLI de GitHub y luego en copiar .



3. El icono de portapapeles para copiar la URL para clonar un repositorio
4. Abrir la Git Bash.
5. Cambia el directorio de trabajo actual a la ubicación en donde quieres clonar el directorio.

```
BID@DESKTOP-Q8DB88L MINGW64 ~/Pictures/Proyecto
$ |
```

6. Escribe git clone, y luego pega la URL que copiaste antes.

***git clone https://github.com/RicharFl/ProyectoP2\_Heldesk.git***

7. Presiona Enter para crear tu clon local.

```
BID@DESKTOP-Q8DB88L MINGW64 ~/Pictures/Proyecto
git clone https://github.com/RicharFl/ProyectoP2_Heldesk.git
Cloning into 'ProyectoP2_Heldesk'...
warning: You appear to have cloned an empty repository.
BID@DESKTOP-Q8DB88L MINGW64 ~/Pictures/Proyecto
```

## Commid de carga de archivos

1. Se ingresa "git add ."

```
BID@DESKTOP-Q8DB88L MINGW64 ~/Pictures/Proyecto/ProyectoP2_Heldesk (main)
$ git add .
warning: LF will be replaced by CRLF in Helpdesk/.mvn/wrapper/maven-wrapper.properties.
The file will have its original line endings in your working directory
```

2. Se ingresa "git commit -m """



```

BID@DESKTOP-Q8DB88L MINGW64 ~/Pictures/Proyecto/ProyectoP2_Helpdesk (main)
$ git commit -m "Creacion de Proyecto Back y front"
[main 76bdbfd] Creacion de Proyecto Back y front
48 files changed, 21663 insertions(+)
create mode 100644 Helpdesk/.mvn/wrapper/maven-wrapper.jar
create mode 100644 Helpdesk/.mvn/wrapper/maven-wrapper.properties
create mode 100644 Helpdesk/HELP.md
create mode 100644 Helpdesk/mvnw
create mode 100644 Helpdesk/mvnw.cmd
create mode 100644 Helpdesk/pom.xml
create mode 100644 Helpdesk/src/main/java/com/ipn/Helpdesk/HelpdeskApplication.java
create mode 100644 Helpdesk/src/main/resources/application.properties
create mode 100644 Helpdesk/src/test/java/com/ipn/Helpdesk/HelpdeskApplicationTests.java
create mode 100644 Helpdesk/target/classes/application.properties
create mode 100644 Helpdesk/target/classes/com/ipn/Helpdesk/HelpdeskApplication.class
create mode 100644 Helpdesk/target/maven-status/maven-compiler-plugin/compile/default-compile/createdFiles.lst
create mode 100644 Helpdesk/target/maven-status/maven-compiler-plugin/compile/default-compile/inputFiles.lst
create mode 100644 Helpdesk/target/maven-status/maven-compiler-plugin/testCompile/default-testCompile/createdFiles.lst
create mode 100644 Helpdesk/target/maven-status/maven-compiler-plugin/testCompile/default-testCompile/inputFiles.lst
create mode 100644 Helpdesk/target/surefire-reports/2022-02-15T22-13-08_470-jvmRun1.dump
create mode 100644 Helpdesk/target/surefire-reports/2022-02-15T22-13-19_347-jvmRun1.dump
create mode 100644 Helpdesk/target/surefire-reports/TEST-com.ipn.Helpdesk.HelpdeskApplicationTests.xml
create mode 100644 Helpdesk/target/surefire-reports/com.ipn.Helpdesk.HelpdeskApplicationTests.txt
create mode 100644 Helpdesk/target/test-classes/com/ipn/Helpdesk/HelpdeskApplicationTests.class
create mode 100644 Helpdesk_F/.browserslistrc
create mode 100644 Helpdesk_F/.editorconfig
create mode 100644 Helpdesk_F/.gitignore
create mode 100644 Helpdesk_F/.vscode/extensions.json
create mode 100644 Helpdesk_F/.vscode/launch.json
create mode 100644 Helpdesk_F/.vscode/tasks.json
create mode 100644 Helpdesk_F/README.md
create mode 100644 Helpdesk_F/angular.json
create mode 100644 Helpdesk_F/karma.conf.js
create mode 100644 Helpdesk_F/package-lock.json
create mode 100644 Helpdesk_F/package.json
create mode 100644 Helpdesk_F/src/app/app.component.css
create mode 100644 Helpdesk_F/src/app/app.component.html
create mode 100644 Helpdesk_F/src/app/app.component.spec.ts
create mode 100644 Helpdesk_F/src/app/app.component.ts
create mode 100644 Helpdesk_F/src/app/app.module.ts
create mode 100644 Helpdesk_F/src/assets/.gitkeep
create mode 100644 Helpdesk_F/src/environments/environment.prod.ts
create mode 100644 Helpdesk_F/src/environments/environment.ts
create mode 100644 Helpdesk_F/src/favicon.ico
create mode 100644 Helpdesk_F/src/index.html
create mode 100644 Helpdesk_F/src/main.ts
create mode 100644 Helpdesk_F/src/polyfills.ts
create mode 100644 Helpdesk_F/src/styles.css
create mode 100644 Helpdesk_F/src/test.ts
create mode 100644 Helpdesk_F/tsconfig.app.json
create mode 100644 Helpdesk_F/tsconfig.json
create mode 100644 Helpdesk_F/tsconfig.spec.json

```

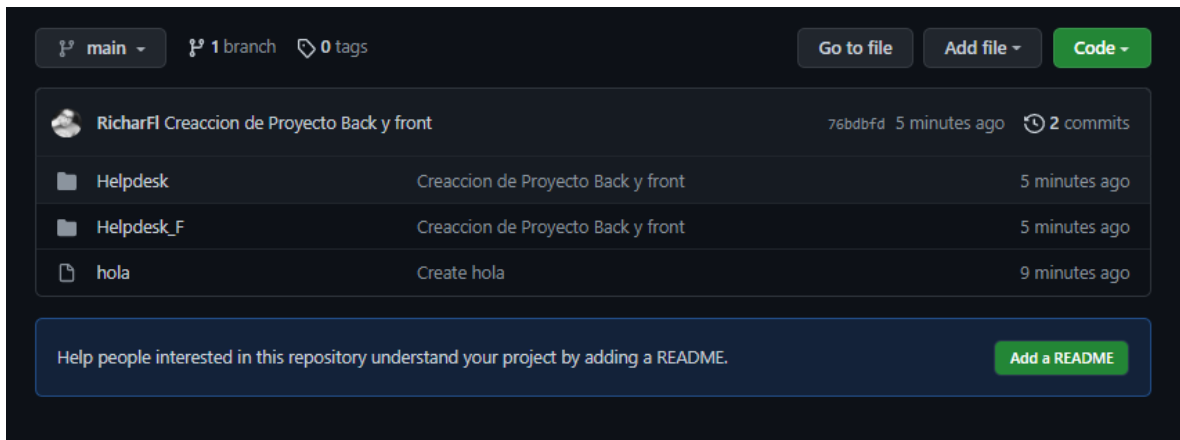
3. Se ingresa “git push -u origin main”

```

BID@DESKTOP-Q8DB88L MINGW64 ~/Pictures/Proyecto/ProyectoP2_Helpdesk (main)
$ git push -u origin main
Enumerating objects: 86, done.
Counting objects: 100% (86/86), done.
Delta compression using up to 4 threads
Compressing objects: 100% (64/64), done.
Writing objects: 100% (85/85), 269.02 KiB | 4.98 MiB/s, done.
Total 85 (delta 4), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (4/4), done.
To https://github.com/RicharFl/ProyectoP2_Helpdesk.git
3de98c8..76bdbfd main -> main
branch 'main' set up to track 'origin/main'.

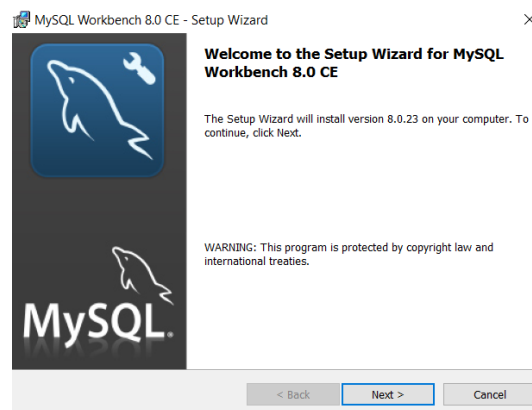
```

4. Viéndose reflejado en el sitio web.

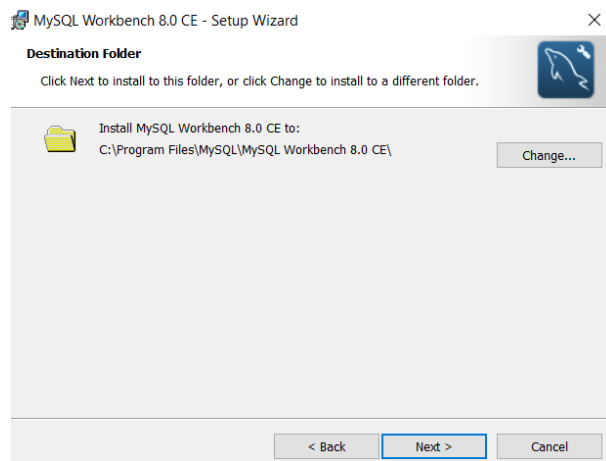


## Instalación de MYSQL Workbench

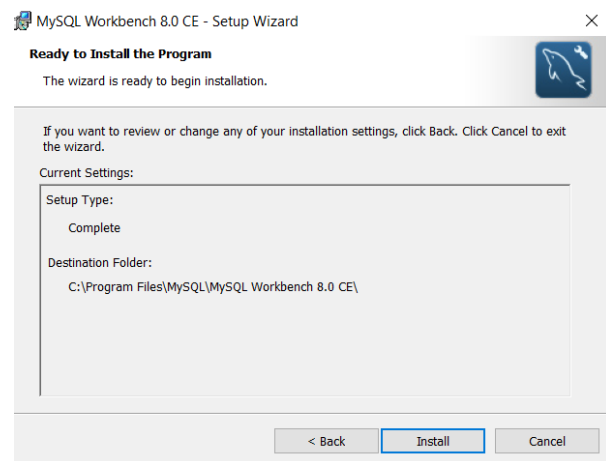
1. Ingresar a la página de MySQL Workbench en AcademicSoftware <https://dev.mysql.com/downloads/file/?id=510038> y haz clic en el botón 'Descargar MySQL Workbench' para descargar el archivo de instalación.
2. Abre el archivo de instalación MSI de tu carpeta de descargas y haz clic en Next para iniciar la instalación.



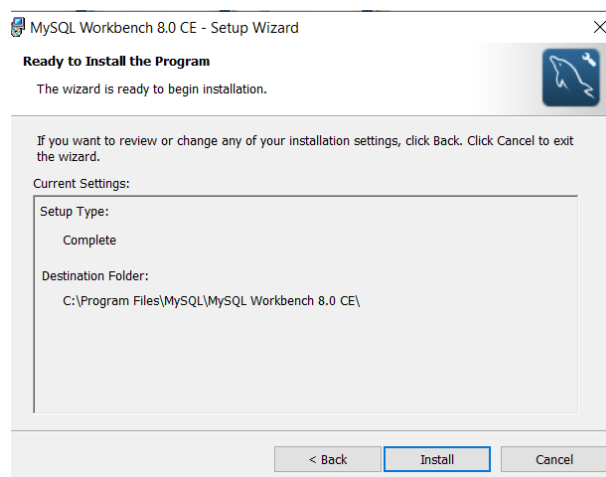
3. Haz clic en Next para instalar el software en la ubicación predeterminada. También puedes cambiar la carpeta de destino.



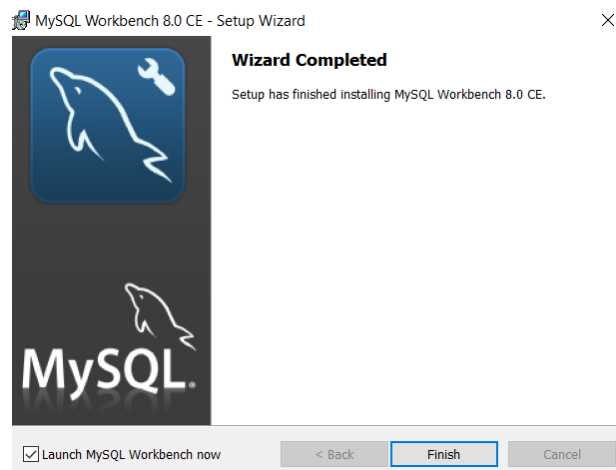
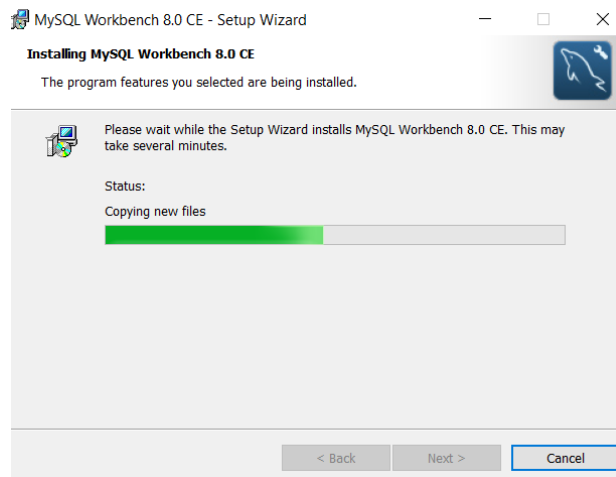
4. Haz clic en Next para instalar todas las funcionalidades del programa



5. Haz clic en Install para iniciar la instalación.



6. Una vez finalizada la instalación, marca la casilla para lanzar el programa y haz clic en Finish.



Despliegue de base de datos Preproducción.

En la siguiente sección se describirá el despliegue del scrip de la base de datos asociada al proyecto, que por convención llamaremos **"Helpdes\_QA\_preP"**.

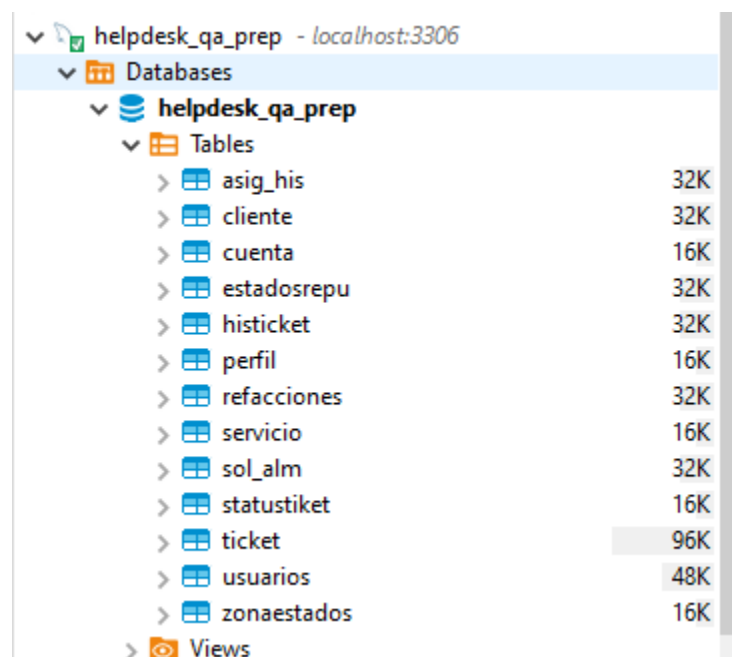
1. El despliegue tiene como objetivo los siguientes computos.
  - a. Implementación de base de datos desde 0.
  - b. Creación de tablas correctamente
2. Pruebas de creación de base de datos.
  - a. Generación de diagrama de arquitectura de la base de datos **"Helpdes\_QA\_preP"**

Para cumplir el alcance del punto 1, se realizó el despliegue por medio de una scrip, el cual se encuentra en el repositorio de git

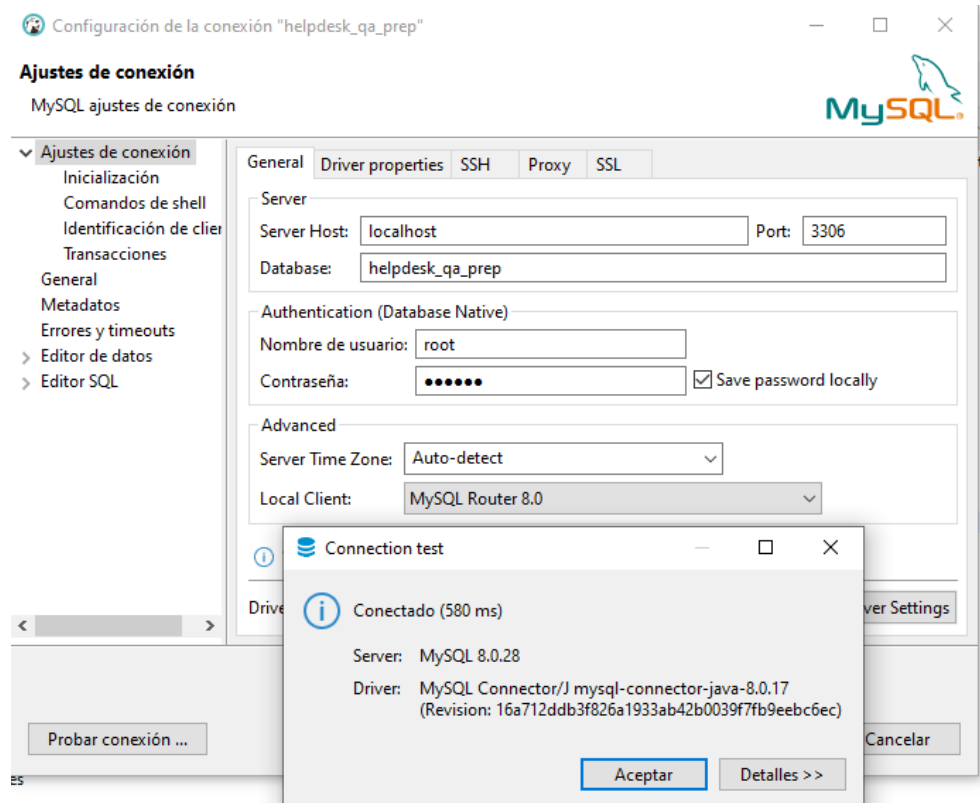
["https://github.com/RicharFI/ProyectoP2\\_Helpdesk/tree/main/BD\\_preP"](https://github.com/RicharFI/ProyectoP2_Helpdesk/tree/main/BD_preP), que tiene como salida por consola lo siguiente.

✓	59	14:00:16	create database Helpdesk_QA_preP	1 row(s) affected	0.140 sec
✓	60	14:00:17	use Helpdesk_QA_preP	0 row(s) affected	0.000 sec

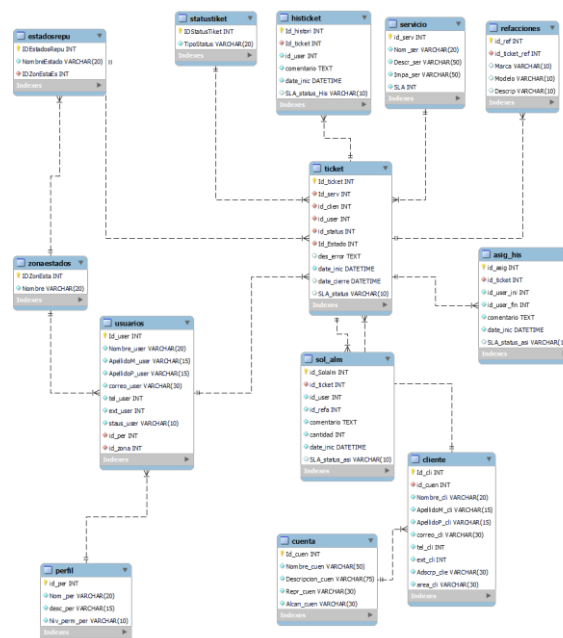
De este modo siguiendo con el despliegue se creará con el mismo scrip las tablas y relaciones que harán parte de la base de datos, esto se puede ver la siguiente imagen.



Para cumplir con el arcense del punto 2, referente al correcto funcionamiento y despliegue de la BD, se documenta la siguiente evidencia, donde se muestra que la conexión a la base de datos se puede dar por el puerto 3306 y la IP 127.0.0.1 (localhost).



De igual manera se genero el diagrama la arquitectura, demostrando el despliegue se realizo correctamente



Continuado con el test de pruebas

## Creación de usuario a nivel BD

Se procederá a la creación de usuarios a nivel base de datos los cuales tendrán permisos asociados a su rol dentro del sistema.

Creación y asignación de permisos a usuario:

### Administrador

El usuario administrador tendrá asignado todos los privilegios para el manejo de la base de datos, este usuario solo será utilizado por al administrador del sistema.

Name	admin_heldes
Password	Usuario_admin_2022Q@
Server Host	Localhost
Port	3306
SSL	allowPublicKeyRetrieval

### SoporteN2

El usuario soporte tendrá asignado los privilegios asignados a el flujo de un ticket.

Name	Soporte_N2
Password	Usuario_Soporte_2022Q@
Server Host	Localhost
Port	3306
SSL	allowPublicKeyRetrieval

La siguiente tabla describe los privilegios asignados por tablas a nivel BD, tiene lo

Tabla	UPDATE	DELETE	INSERT
helpdesk_qa_prep.usuarios	Ok	N/A	OK
helpdesk_qa_prep.sol_alm	Ok	N/A	OK
helpdesk_qa_prep.ticket	OK	N/A	OK
helpdesk_qa_prep.cliente	OK	OK	OK

### Almacén

El usuario soporte tendrá asignado los privilegios asignados a el flujo de una solicitud de ticket.

Name	Almacen_N1
Password	Usuario_almacen_2022Q@
Server Host	Localhost

Port	3306
SSL	allowPublicKeyRetrieval

Tabla	UPDATE	DELETE	INSERT
helpdesk_qa_prep.usuarios	Ok	N/A	OK
helpdesk_qa_prep.sol_alm	Ok	N/A	OK
helpdesk_qa_prep.ticket	OK	N/A	OK

Para la generación de los permisos se desplegará en el gestor de base de datos, el siguiente script “user\_per\_prep” que se encuentra en el repositorio en la siguiente URL [https://github.com/RicharFI/ProyectoP2\\_Heldesk/tree/main/BD\\_prep](https://github.com/RicharFI/ProyectoP2_Heldesk/tree/main/BD_prep)

Teniendo como salida de ejecución las siguientes imágenes, donde se observará los privilegios dados a los usuarios

```
Grants for almacen_n1@localhost
GRANT USAGE ON *.* TO `almacen_n1`@`localhost`
GRANT SELECT, UPDATE ON `helpdesk_qa_prep`.`sol_alm` TO `almacen_n1`@`localhost`
GRANT SELECT, UPDATE ON `helpdesk_qa_prep`.`ticket` TO `almacen_n1`@`localhost`
GRANT SELECT, UPDATE ON `helpdesk_qa_prep`.`usuarios` TO `almacen_n1`@`localhost`
```

```
Grants for soporte_n2@localhost
GRANT USAGE ON *.* TO `soporte_n2`@`localhost`
GRANT SELECT, UPDATE, DELETE ON `helpdesk_qa_prep`.`cliente` TO `soporte_n2`@`localhost`
GRANT SELECT, UPDATE ON `helpdesk_qa_prep`.`sol_alm` TO `soporte_n2`@`localhost`
GRANT SELECT, UPDATE ON `helpdesk_qa_prep`.`ticket` TO `soporte_n2`@`localhost`
GRANT SELECT, UPDATE ON `helpdesk_qa_prep`.`usuarios` TO `soporte_n2`@`localhost`
```

```
Grants for admin_heldes@localhost
GRANT USAGE ON *.* TO `admin_heldes`@`localhost`
GRANT ALL PRIVILEGES ON `helpdesk_qa_prep`.* TO `admin_heldes`@`localhost`
```

## Backend APIS REST

Los procesos para desarrollar en la sección de Backed estarán referidos al desarrollo de las implementaciones necesarias descritas al inicio de dicho documento.



Por lo anterior, la información se encontrará desglosada por APIS REST, en el desarrollo estará descrito por la siguiente estructura MVC

- Modelos
- Servicios
- Servicios IMPL
- Controladores
- Repositorios

Siguiendo el diseño antes misionado se describen los siguientes APIS REST así como sus End Point

## API Almacén

### 1. Modelo

```
@Table name = "Almacen", schema = "helpdesk_qa_prep")
public class Almacen implements Serializable {
-----Codigo-----
}
```

### 2. Service

```
public interface AlmacenService {
{
    Almacen AgregarProducto (Almacen almacen);

    Almacen ActulizarStockProductos (Almacen almacen);

    Set<Almacen> ListaProductosAlmacen ();

    void EliminarProductos (Long Id_almacen);

    Almacen ObtenerProductoAlmacen (Long Id_almacen);
}
```

### 3. Service IMPL

Desarrollo de cada uno de los service con las reglas de negocio, lo anterior se encuentra descrito en el repositorio antes mencionado.

#### 4. Controller

```
@RestController
@RequestMapping("/IPN/helpdesk/Almacen/")
@CrossOrigin("*")
public class AlmacenController {

    @Autowired
    private AlmacenService almacenService;

    @PostMapping("/") // PARA GUARDAR UN PRODUCTO
    public ResponseEntity<Almacen> AgregarProductoAlmacen(@RequestBody Almacen almacen) {
        Almacen agregarProductoAlmacen = almacenService.AgregarProducto(almacen);
        return ResponseEntity.ok(agregarProductoAlmacen);
    }

    @GetMapping("/")
    public ResponseEntity<?> ListarProductosAlmacen ()
    {
        return ResponseEntity.ok(almacenService.ListaProductosAlmacen());
    }

    @GetMapping("/{id_almacen}")

    public Almacen ListaProductosAlmacenPorId(@PathVariable("id_almacen") Long Id_almacen) {
        return almacenService.ObtenerProductoAlmacen(Id_almacen);
    }

    @PutMapping("/")
    public Almacen ActulizarProductoAlmacen(@RequestBody Almacen almacen) {
        return almacenService.ActulizarStockProductos(almacen);
    }

    @DeleteMapping("/{id_almacen}")
    public void EliminarProductoAlmacen (@PathVariable("id_almacen") Long Id_almacen)
    {
        almacenService.EliminarProductos(Id_almacen);
    }
}
```

#### 5. Repositorio

```
1 package com.ipn.helpdesk.repositories;
2
3 import org.springframework.data.jpa.repository.JpaRepository;
4
5 @Repository
6 public interface AlmacenRepository extends JpaRepository<Almacen, Long>
7 {
8 }
9 }
```

#### API Cliente

El código del Desarrollo de Backed se encuentra en el Repositorio siguiente:

[https://github.com/RicharFI/ProyectoP2\\_Heldesdesk.git](https://github.com/RicharFI/ProyectoP2_Heldesdesk.git)

#### API Cuenta

El código del Desarrollo de Backed se encuentra en el Repositorio siguiente:

[https://github.com/RicharFI/ProyectoP2\\_Heldesdesk.git](https://github.com/RicharFI/ProyectoP2_Heldesdesk.git)

### API Estados Republica

El código del Desarrollo de Backed se encuentra en el Repositorio siguiente:

[https://github.com/RicharFI/ProyectoP2\\_Heldesk.git](https://github.com/RicharFI/ProyectoP2_Heldesk.git)

### API Historia de Asignación

El código del Desarrollo de Backed se encuentra en el Repositorio siguiente:

[https://github.com/RicharFI/ProyectoP2\\_Heldesk.git](https://github.com/RicharFI/ProyectoP2_Heldesk.git)

### API Hstoria de Ticket

El código del Desarrollo de Backed se encuentra en el Repositorio siguiente:

[https://github.com/RicharFI/ProyectoP2\\_Heldesk.git](https://github.com/RicharFI/ProyectoP2_Heldesk.git)

### API Perfil

El código del Desarrollo de Backed se encuentra en el Repositorio siguiente:

[https://github.com/RicharFI/ProyectoP2\\_Heldesk.git](https://github.com/RicharFI/ProyectoP2_Heldesk.git)

### API Servicios

El código del Desarrollo de Backed se encuentra en el Repositorio siguiente:

[https://github.com/RicharFI/ProyectoP2\\_Heldesk.git](https://github.com/RicharFI/ProyectoP2_Heldesk.git)

#### API SLA Tikect

El código del Desarrollo de Backed se encuentra en el Repositorio siguiente:

[https://github.com/RicharFI/ProyectoP2\\_Heldesk.git](https://github.com/RicharFI/ProyectoP2_Heldesk.git)

#### API Solicitud a Almacen

El código del Desarrollo de Backed se encuentra en el Repositorio siguiente:

[https://github.com/RicharFI/ProyectoP2\\_Heldesk.git](https://github.com/RicharFI/ProyectoP2_Heldesk.git)

#### API Estatus Ticket

El código del Desarrollo de Backed se encuentra en el Repositorio siguiente:

[https://github.com/RicharFI/ProyectoP2\\_Heldesk.git](https://github.com/RicharFI/ProyectoP2_Heldesk.git)

#### API Ticket

El código del Desarrollo de Backed se encuentra en el Repositorio siguiente:

[https://github.com/RicharFI/ProyectoP2\\_Heldesk.git](https://github.com/RicharFI/ProyectoP2_Heldesk.git)

#### API Usuario

El código del Desarrollo de Backed se encuentra en el Repositorio siguiente:

[https://github.com/RicharFI/ProyectoP2\\_Heldesk.git](https://github.com/RicharFI/ProyectoP2_Heldesk.git)

#### API Zona Estado

El código del Desarrollo de Backed se encuentra en el Repositorio siguiente:

[https://github.com/RicharFI/ProyectoP2\\_Heldesk.git](https://github.com/RicharFI/ProyectoP2_Heldesk.git)

