

GA7-220501096-AA3-EV01 Codificación de módulos del software Stand alone, web y móvil de acuerdo al proyecto a desarrollar.

Mayra Alejandra Montalvo Herrera

Cristina Isabel Reales Echeverri

Análisis y Desarrollo de Software

Ficha 2879657 y 2879658

Maritza Calvache Chicangana

2025

**Introducción**

El presente documento aborda la conexión entre la base de datos prueba y el framework Spring utilizando el entorno de desarrollo NetBeans, con el lenguaje de programación Java. Se explican los pasos necesarios para establecer dicha conexión y se detallan las operaciones básicas que se pueden realizar sobre la base de datos, tales como la creación, modificación, eliminación y consulta de registros.

El enfoque principal está orientado al manejo de datos de aprendices, permitiendo una gestión eficiente y estructurada mediante el uso de tecnologías modernas en el desarrollo de aplicaciones Java.

Tabla de contenido

[PASOS A REALIZAR: 3](#_Toc197801797)

[Se crean los 4 paquetes solicitados 10](#_Toc197801798)

[Se crea el paquete repository. 12](#_Toc197801799)

[**Crear la configuración de los paquetes** 13](#_Toc197801800)

[**Aprendiz.java** 14](#_Toc197801801)

[**Agregamos el Paquete repositorio** 15](#_Toc197801802)

[**Insertamos los métodos para que permita Crear – Leer – Actualizar – Eliminar de la base de datos** 20](#_Toc197801803)

[Creamos la base de datos prueba 21](#_Toc197801804)

[**Codificación del paquete Controller** 23](#_Toc197801805)

# PASOS A REALIZAR:

Descargar spring

Se crea proyect Maven

Language Java

Spring boot 3.4.4

Group com.Proysena

Artifact ejemplo

Name ejemplo

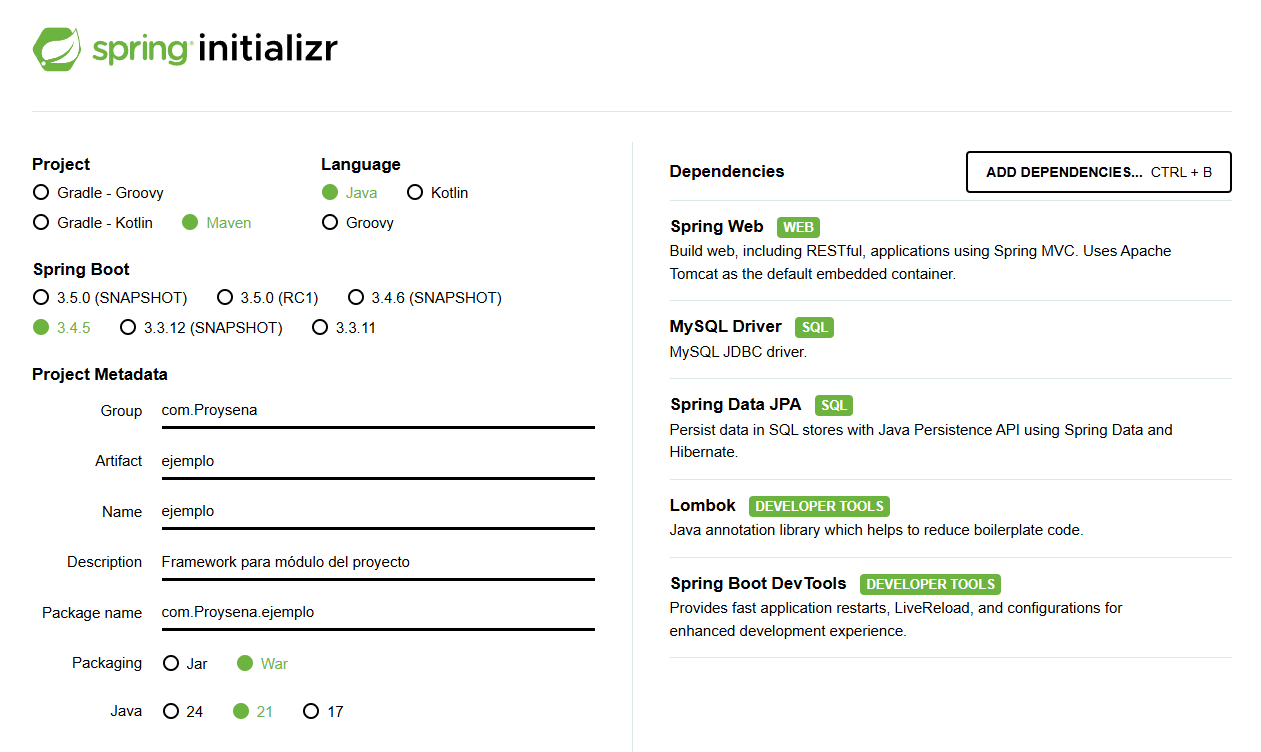
Descripción Framework para módulo del proyecto

Package name com.Proysena.demo

Packaging JWar

JAVA 22



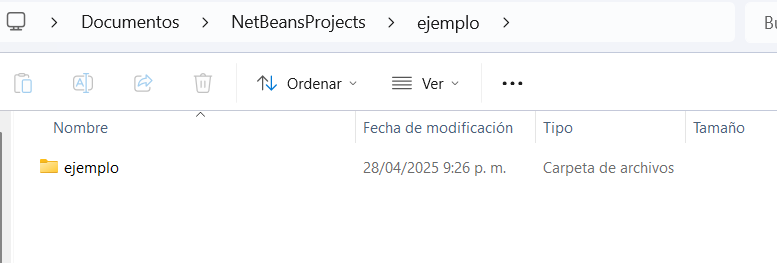


Oprimir generate

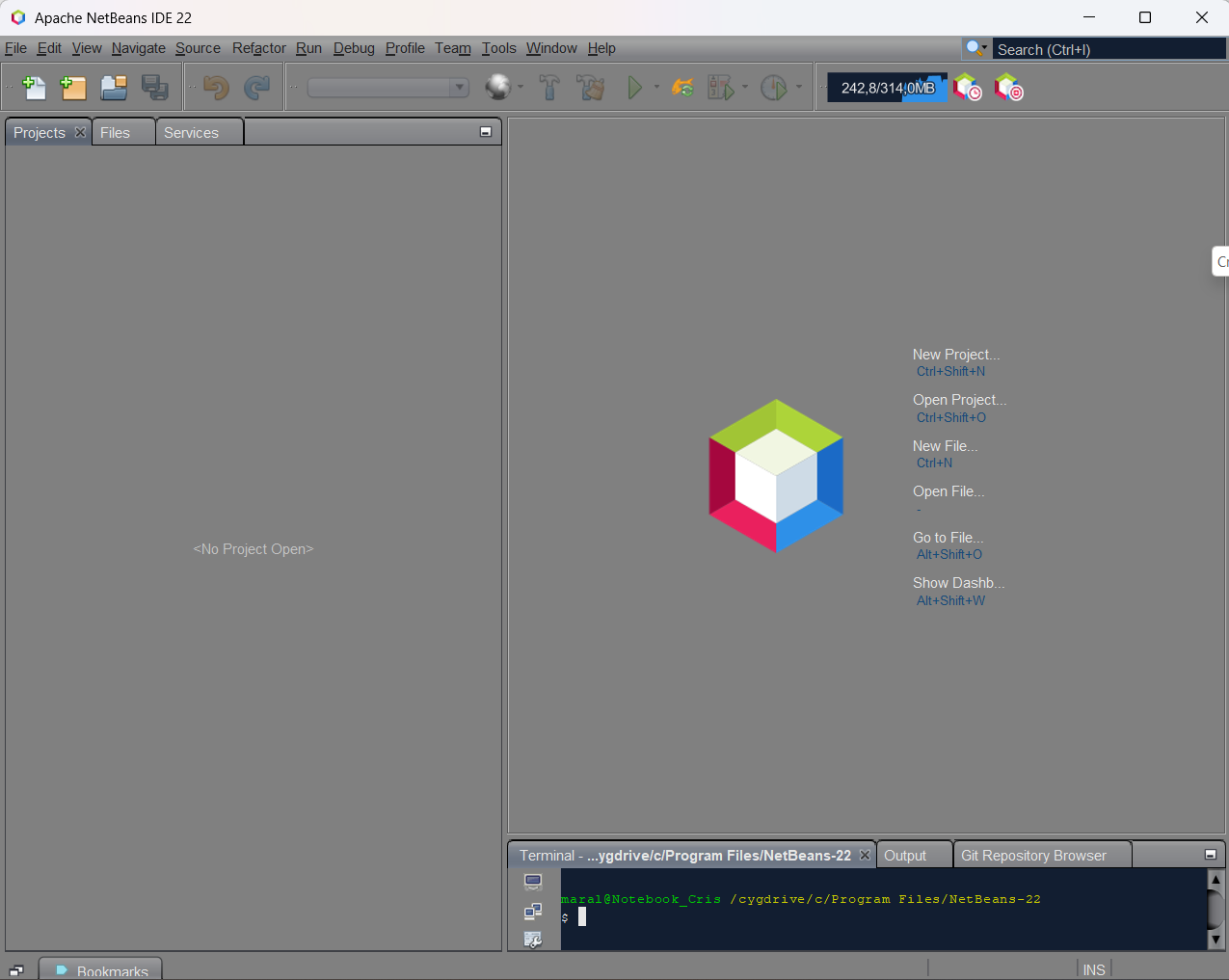
Se descarga un zip llamado ejemplo

Lo descomprimimos, lo cual crea la carpeta ejemplo

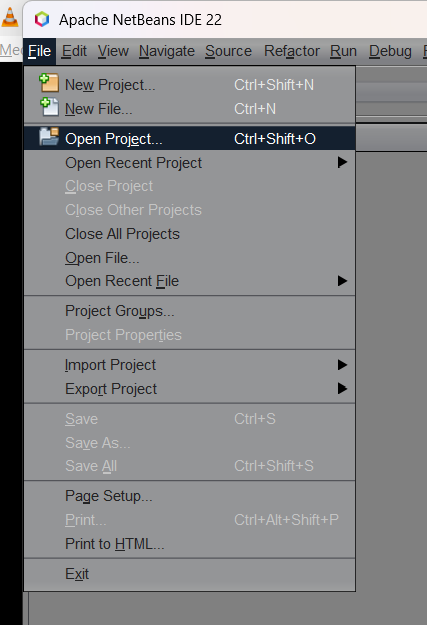
Movemos la carpeta a Mis documentos / Net Beans Proyects

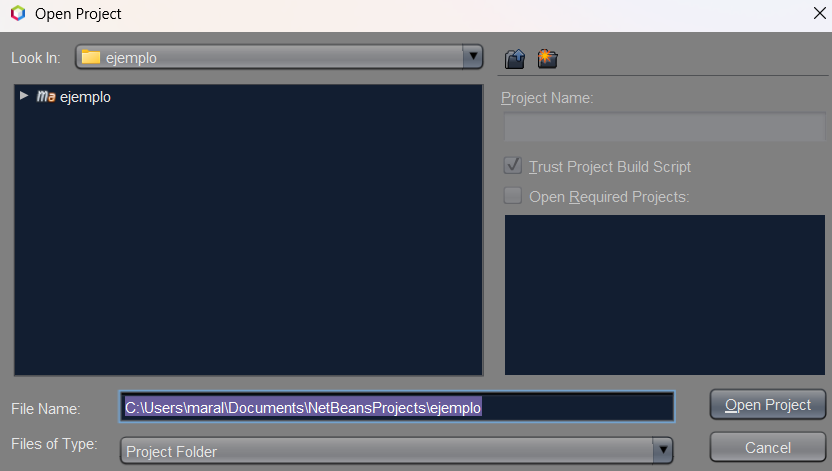


Abrimos Net beans

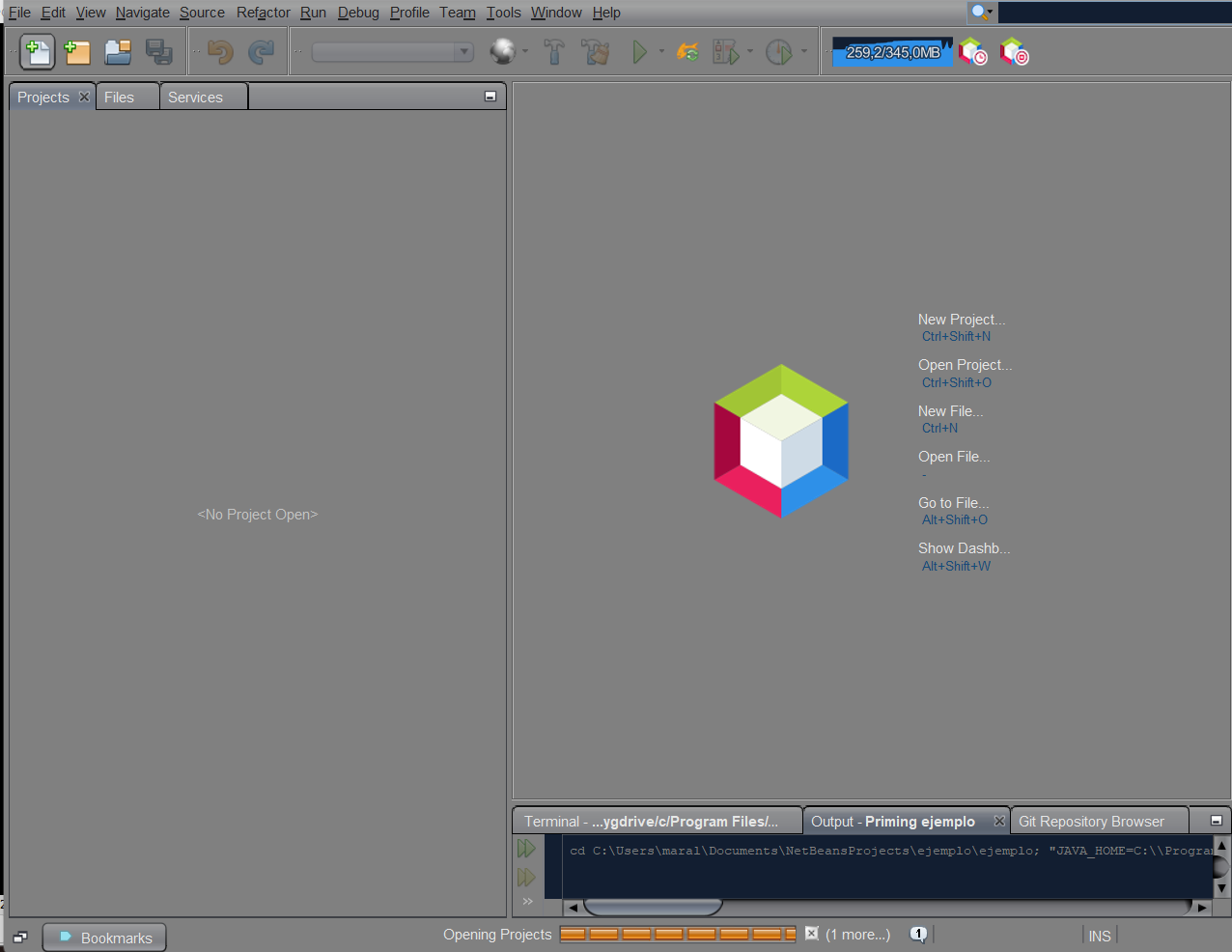


Abrimos el proyecto ejemplo



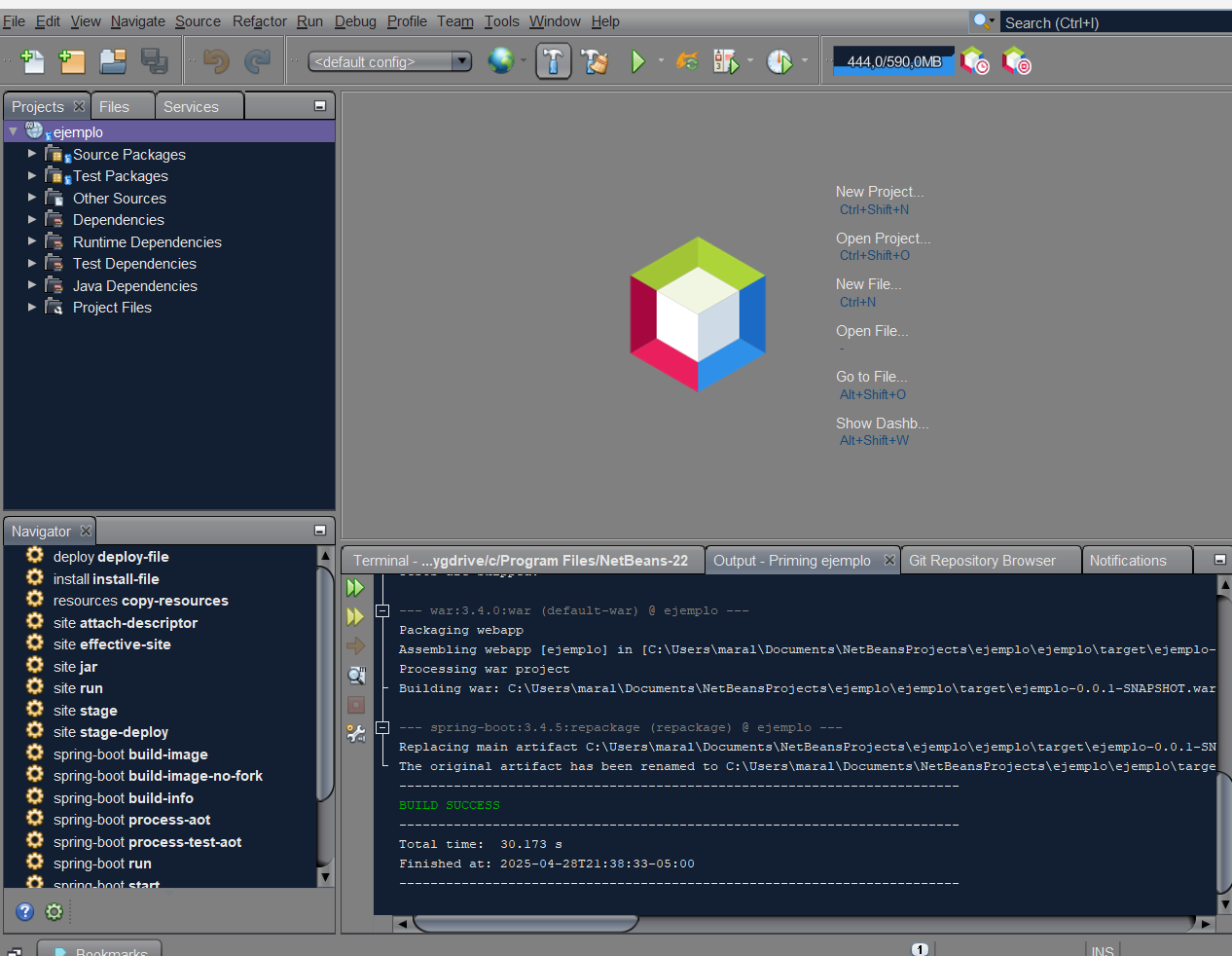


Seleccionamos 

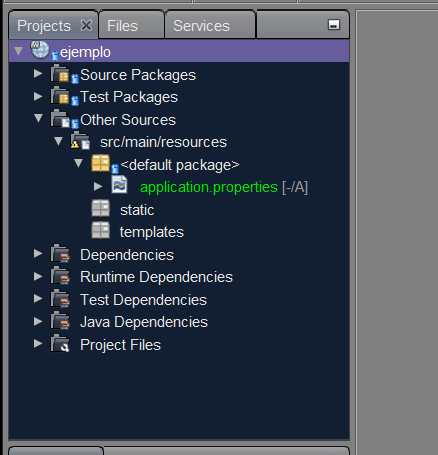


Abrimos Tomcat

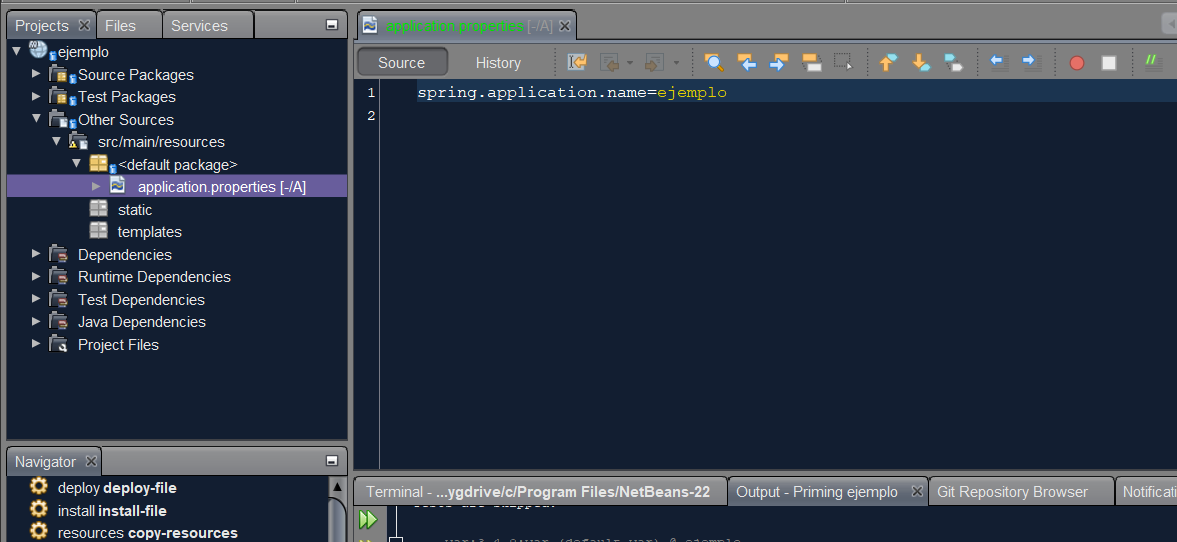
Arroja el mensaje BUILD SUCCESS y se crea el proyecto en la barra izquierda

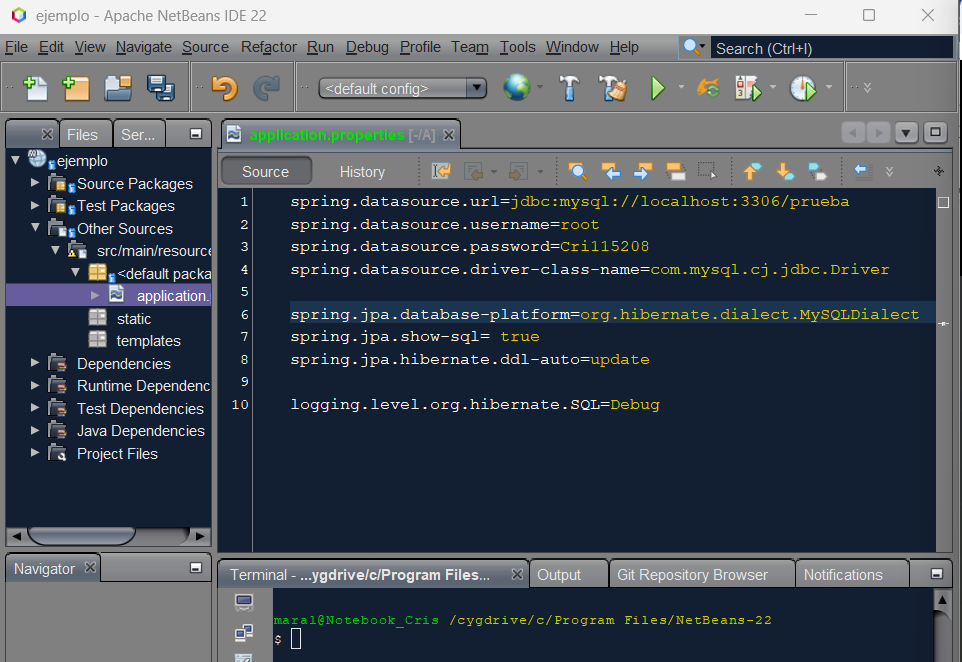


Doble clic para abrir el archivo



Esto abre la ventana aplication.properties





Creamos la conexión con Hibernate a la base de datos

Se crean los 4 paquetes solicitados:

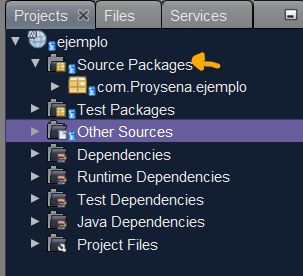
Model: configuración de la base de datos, es donde se crean todas las tablas.

Repository: que se guarde en repositorio

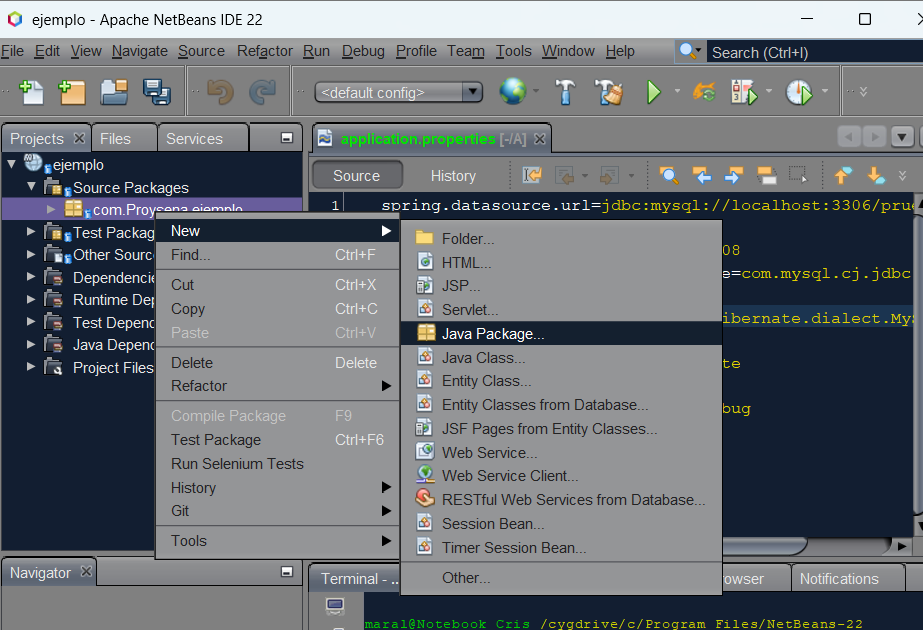
Service: como el Tomcat que ayuda a que podamos visualizar el proyecto en web.

Controller: Controla la comunicación entre los archivos que creamos.

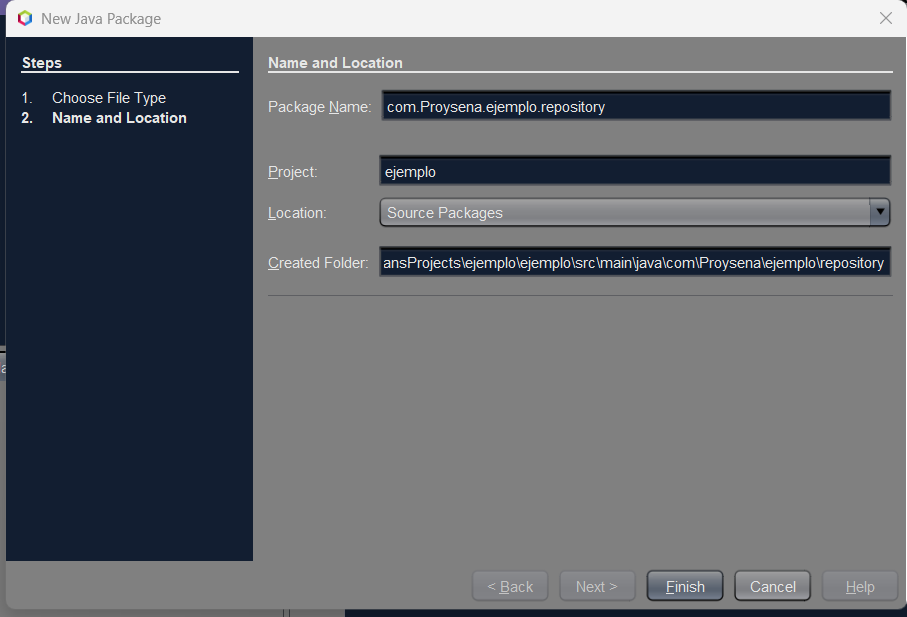
Se crean los paquetes en el Source Packages



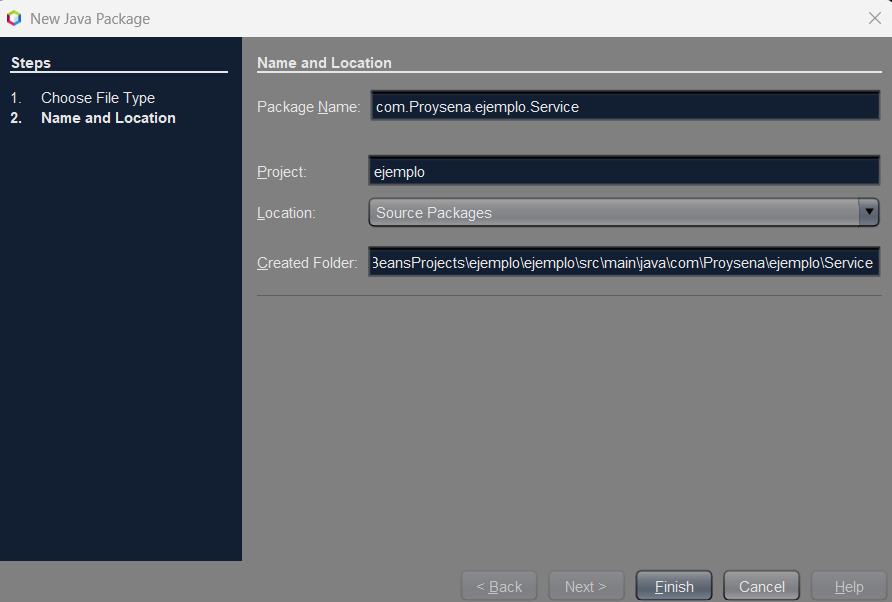
En Source Packages / com.Proysena.ejemplo damos clic derecho New/ Java Package…



Colocamos al final del nombre la palaba Model para crear el primer paquete solicitado. Se crea este paquete ya que es el que hará de tabla en la base de datos.



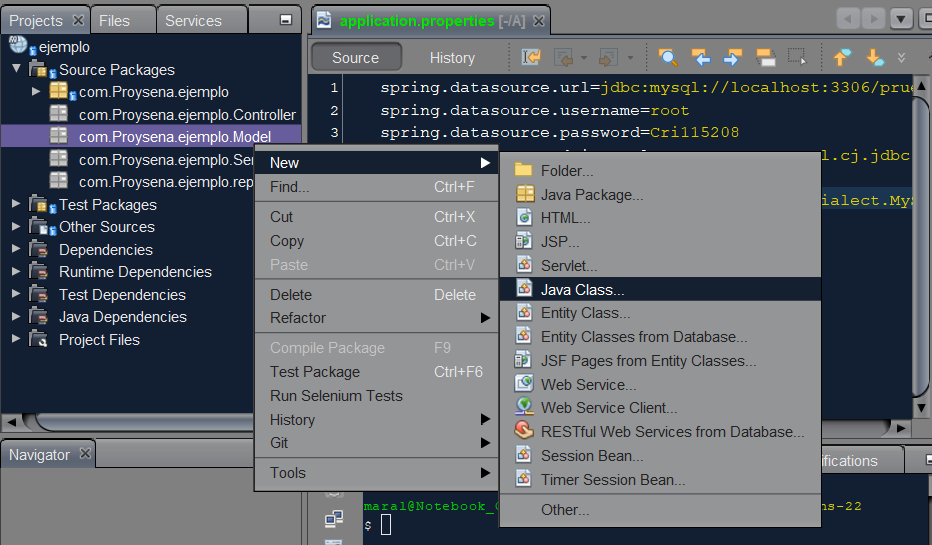
# Se crea el paquete repository.

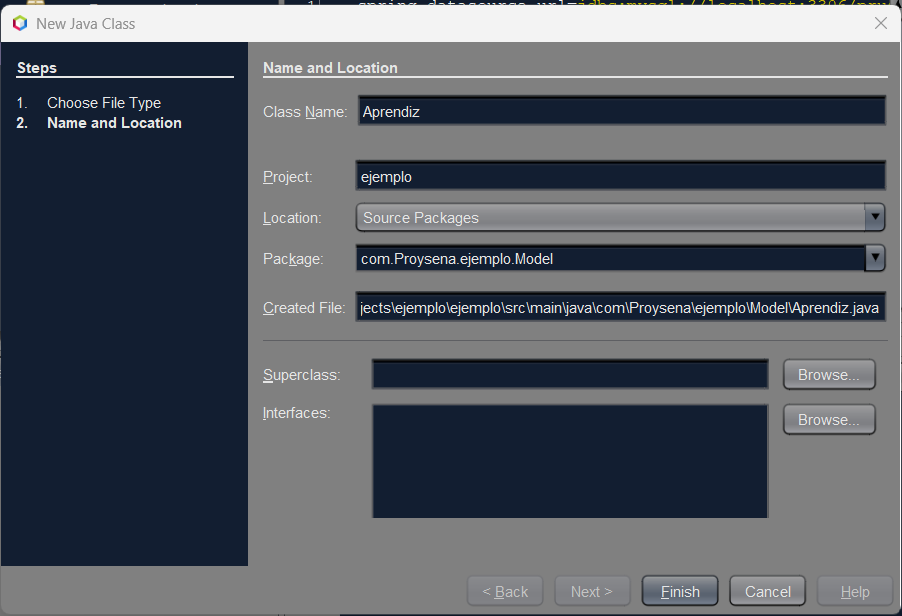


Se crea el paquete Service

# **Crear la configuración de los paquetes**

En el paquete Model se crea una Java Class como se muestra acontinuación:

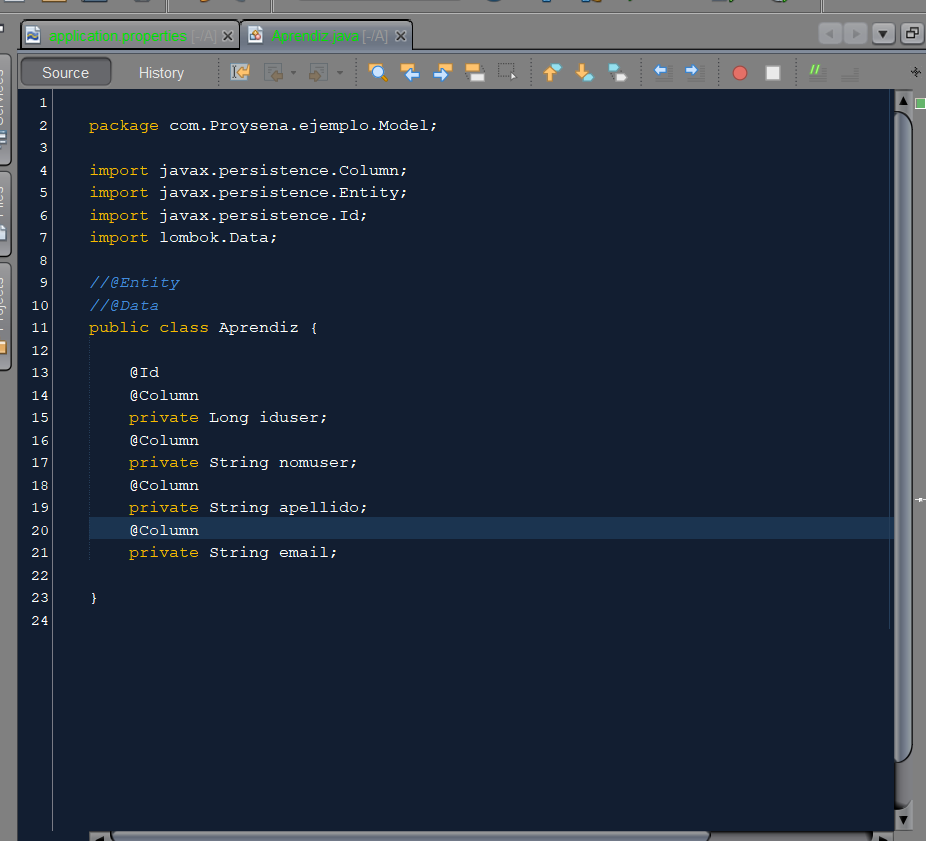




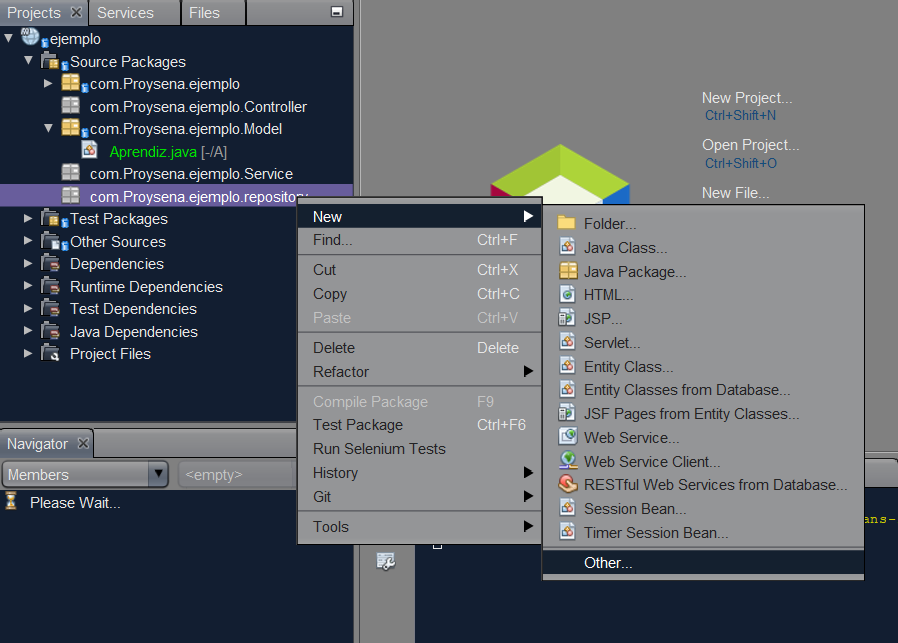
Se crea la clase que al cambiarle el nombre queda como Aprendiz, la cual será la clase para administrar la tabla Aprendiz dentro de la tabla prueba. Colocamos Finish.

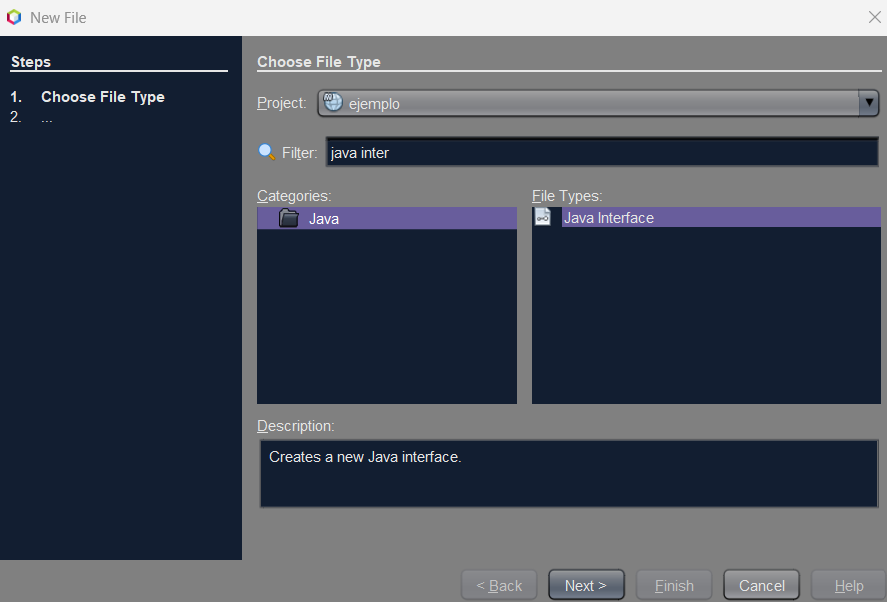
# **Aprendiz.java**

A la clase vamos a insertar el código de las opciones para insertar los datos a la tabla llamada Aprendiz.

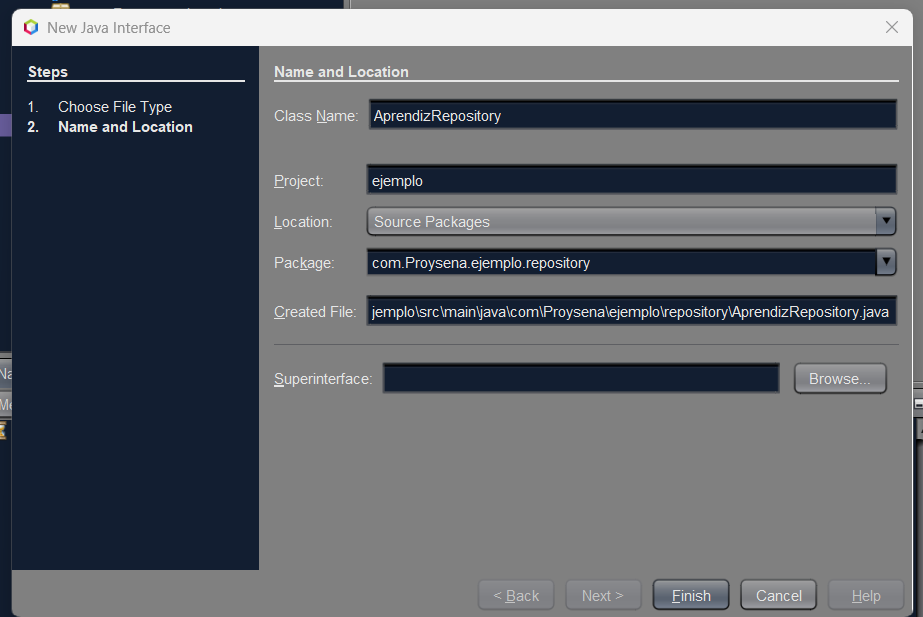


**Agregamos el Paquete repositorio el cual creará una interfaz de comunicación.**

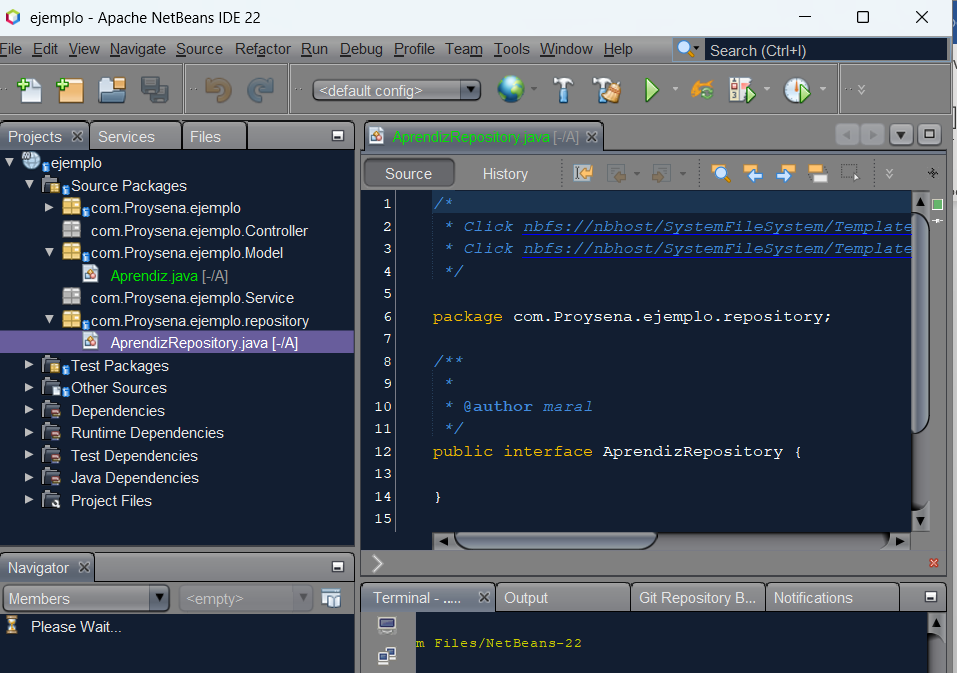
Nos ubicamos en la carpeta Repository /New / Other / Java Interface

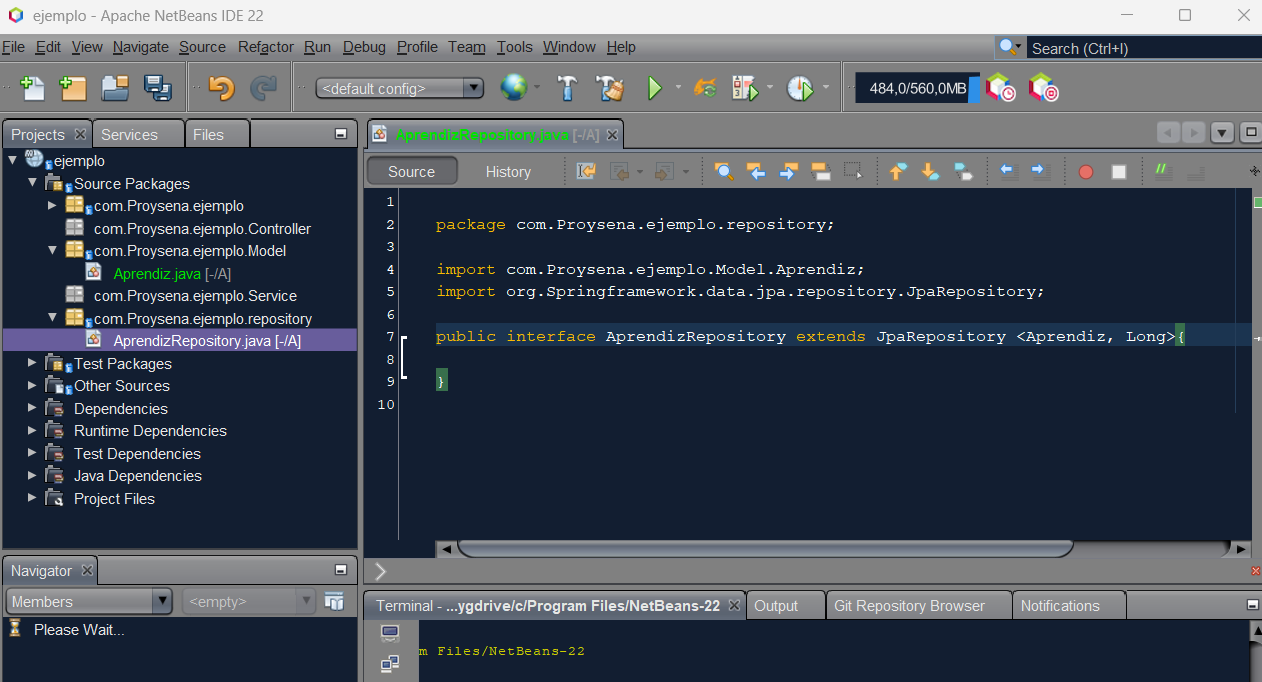


Le damos clic en Next, y sale otro cuadro donde editamos el nombre:



Se coloca el nombre AprendizRepository / Finish

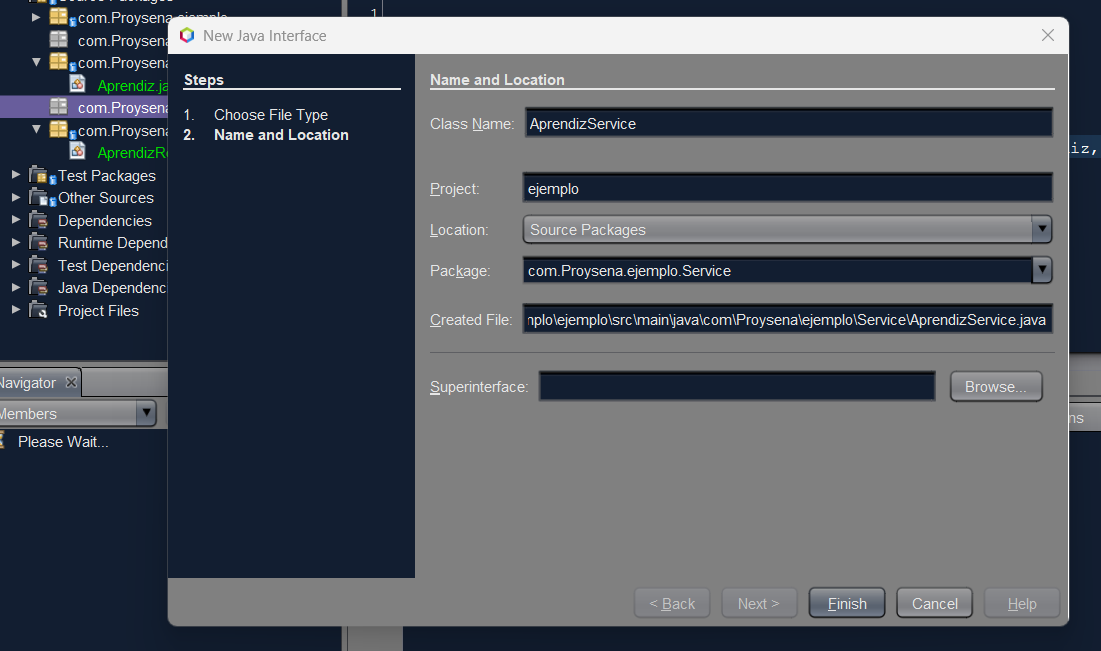
Esto crea AprendizRepository.java.



Las anteriores líneas de código establecen la comunicación tanto al paquete Model como al Repository.

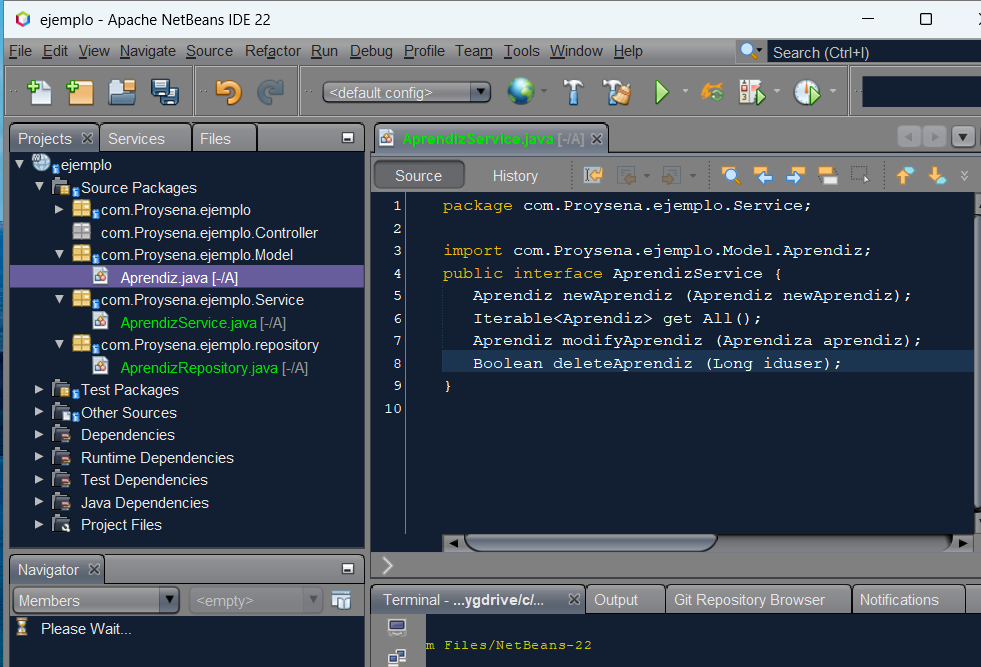
Hacemos el mismo proceso con el archivo com.Proysena.Service

La interface que se crea tiene como nombre en este caso AprendizService y se coloca Finish



Lo anterior crea el archivo AprendizService.java

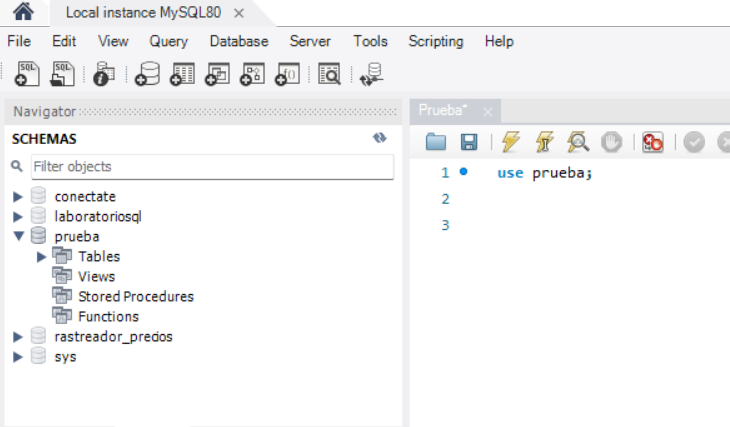
En la que importaremos del Proyecto la capa del Modelo y al Aprendiz

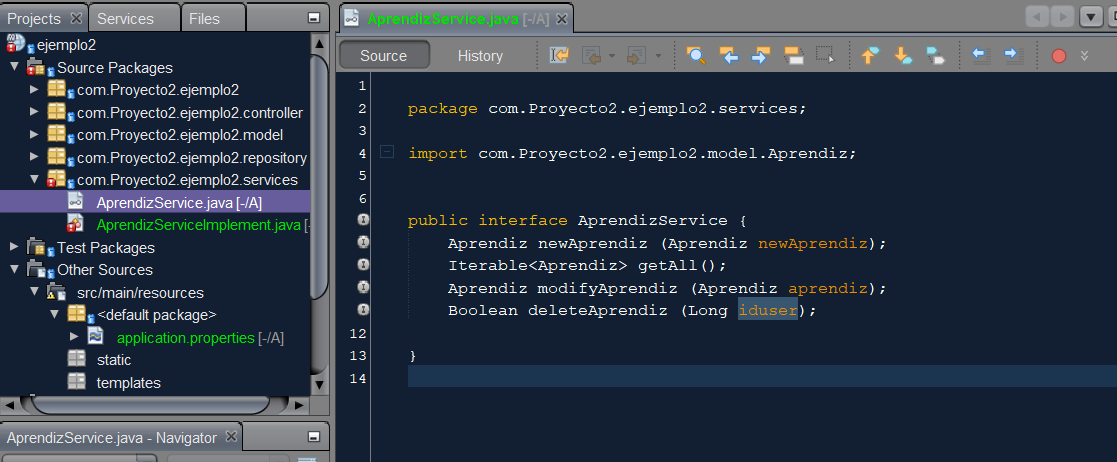


## **Insertamos los métodos para que permita Crear – Leer – Actualizar – Eliminar de la base de datos**

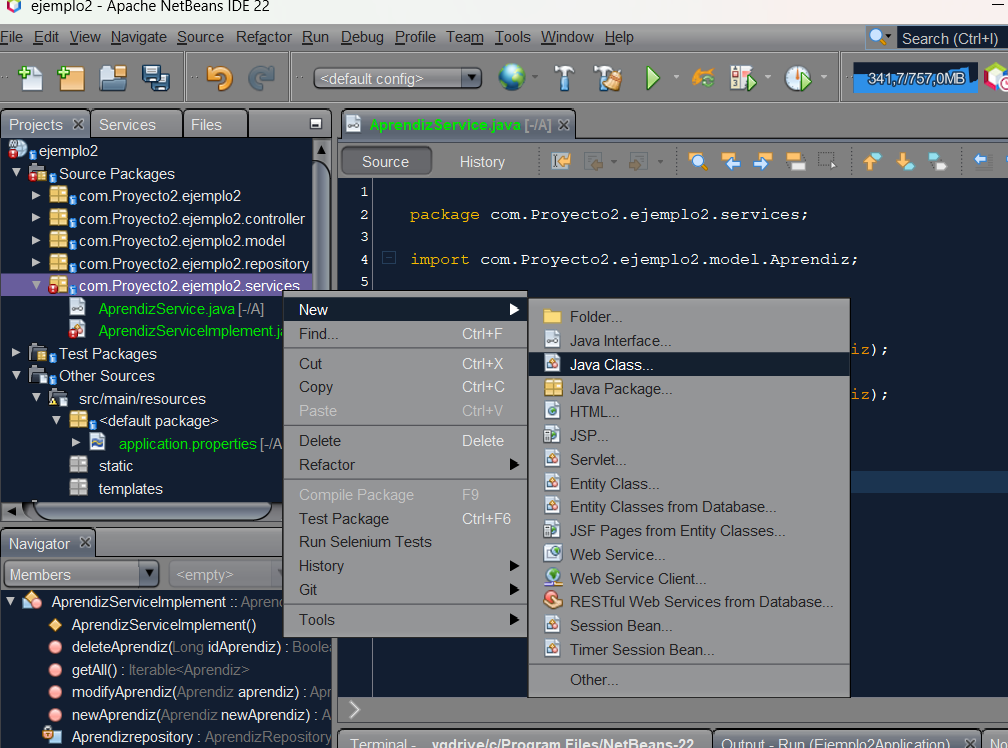
Creamos la clase dentro de servicios para hacer la implementación del funcionamiento

# Creamos la base de datos prueba



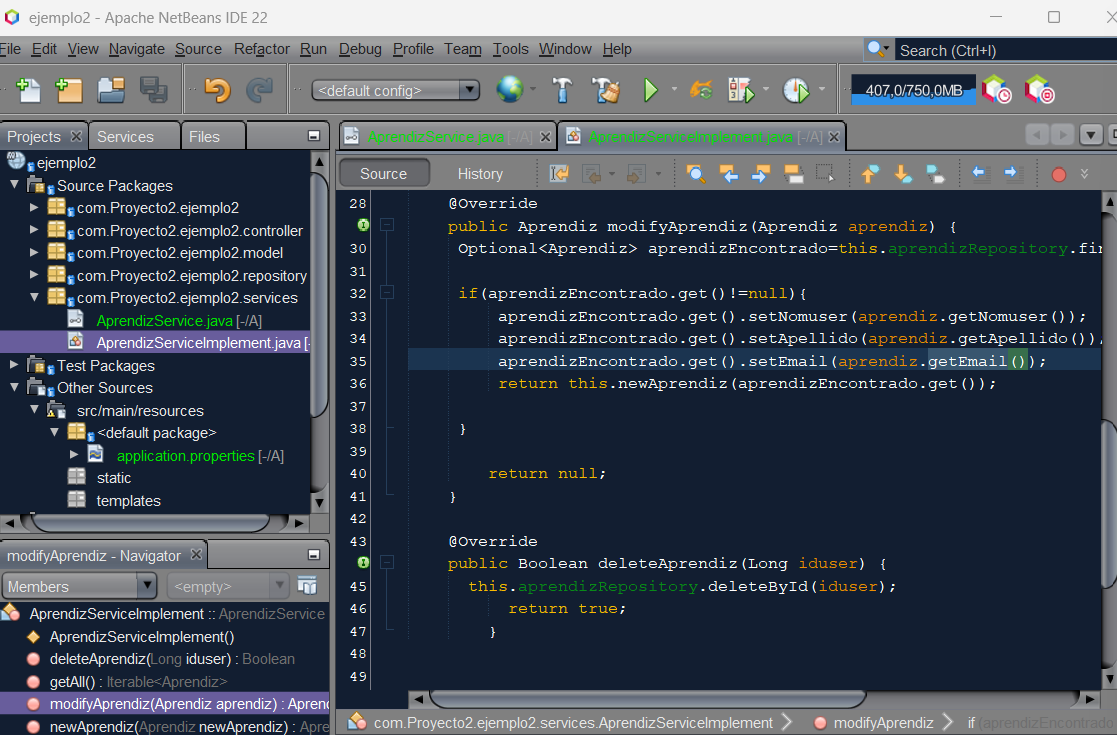


Se crea en Service, El paquete AprendizService.java



Se crea un New / Java Class / AprendizServiceImplement

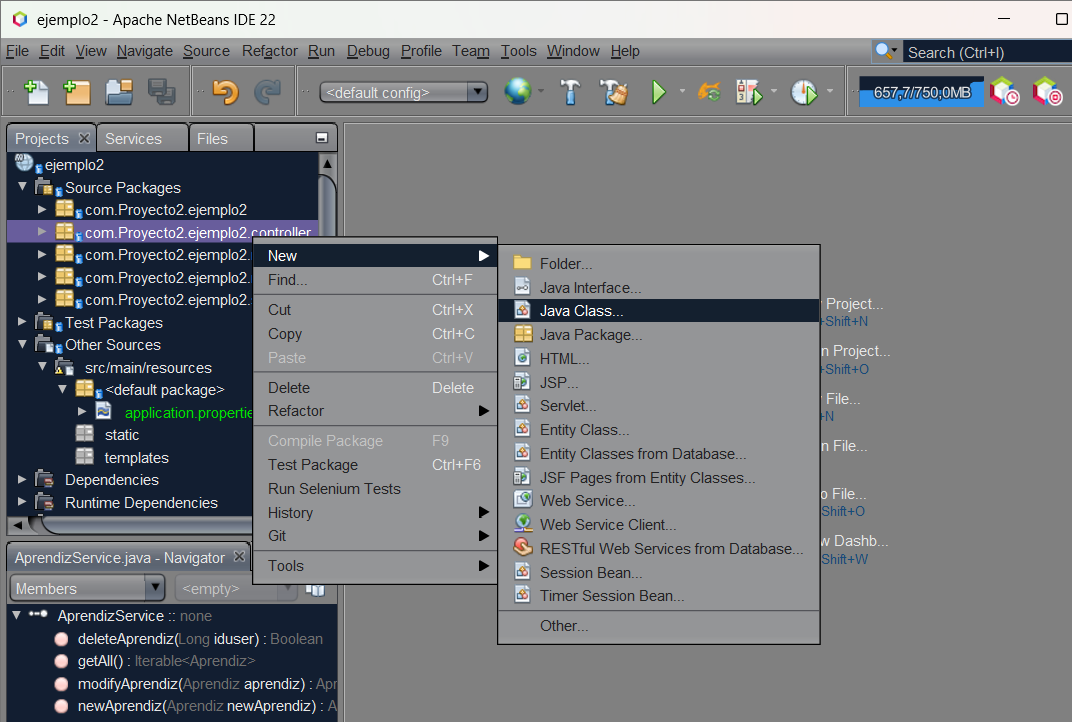
En el cual se van a implementar los métodos abstractos que se codificaron en AprendizService.java



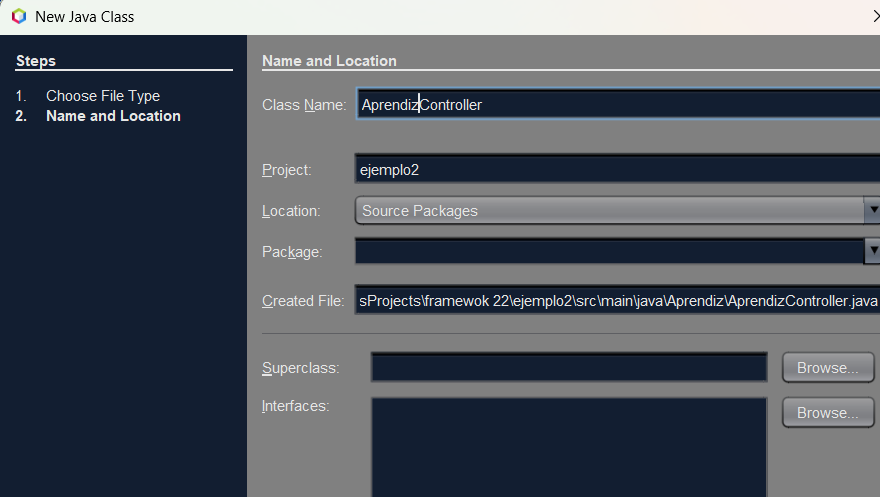
En el cual se codifica la creación, visualización, modificación y borrado del Aprendiz.

# **Codificación del paquete Controller**

Clic derecho en com.Proyecto2.ejemplo.controller / New / Java Class /



Se le coloca el nombre de AprendizController



El Spring va a trabajar con un “mapeo de información” con el cual va haciendo la captura de lo que necesitemos.

**@RestController**

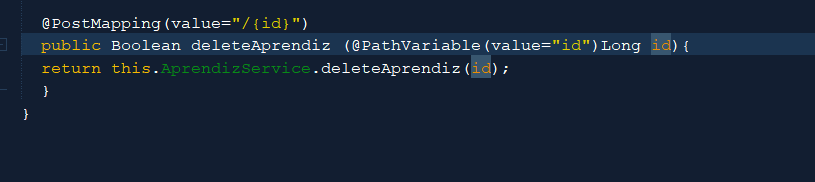
**@RequestMapping** hace el enlace con el entorno gráfico.

public Aprendiz newAprendiz(RequestBody): sirve para crear un nuevo aprendiz en un entorno Html

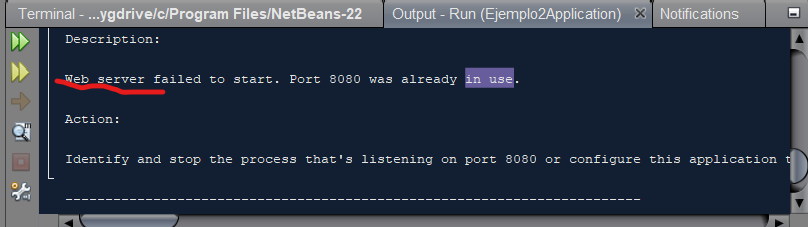
**@GetMapping** Es usado para mostrar la información.

**@PostMapping** (“/modificar”)

**@PostMapping** (value=“/{id}”) sirve para con solo el ide borrar los datos del aprendiz.



Se prueba con Clic derecho en Ejemplo2Application.java, se coloca Run File y es probable que salga un error así:

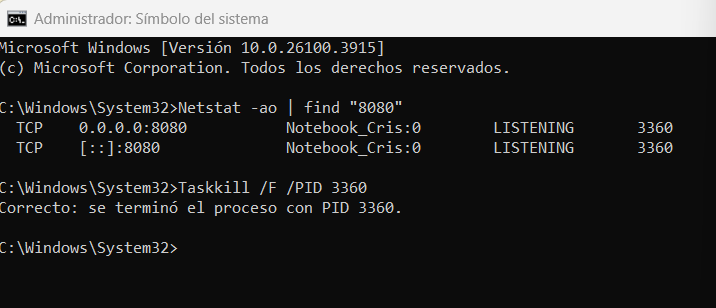
Lo que quiere decir que el puerto 8080 está en uso y para poder correr el programa se debe cerrar el aplicativo que lo esté utilizando, para eso iniciamos CDM en modo administrador, colocamos la siguiente sentencia:

**Netstat -ao | find "8080"**

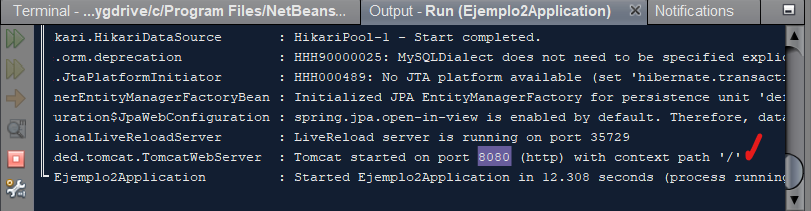
La cual sirve para buscar el programa que use el puerto 8080,

Después de LISTENING va a dar un número (en este caso fue 3360) el cual debe colocarse al final de la siguiente sentencia:

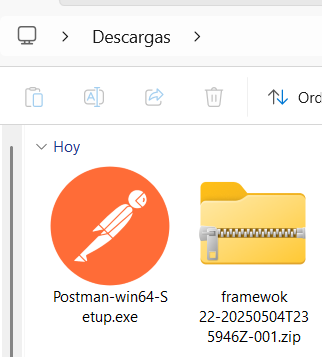
**Taskkill /F /PID 3360**



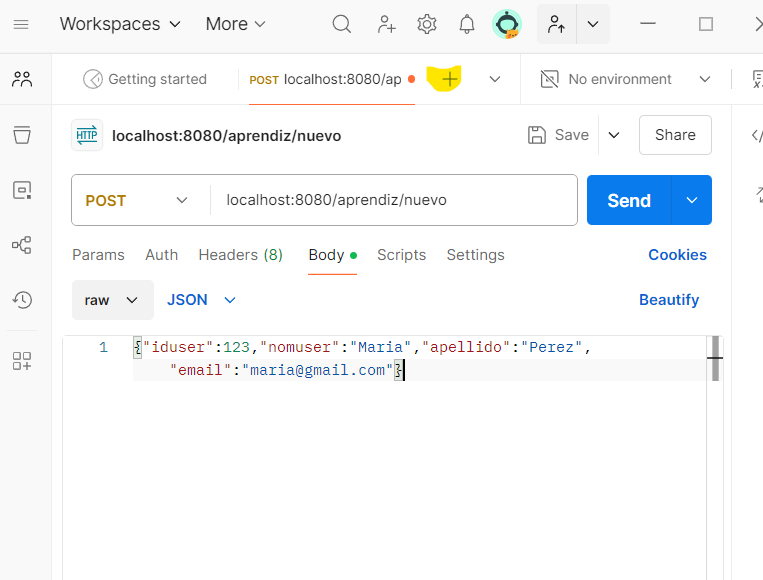
Al colocar Run File de nuevo ya permite la conexión al puerto 8080:



Se descarga Postman para Windows

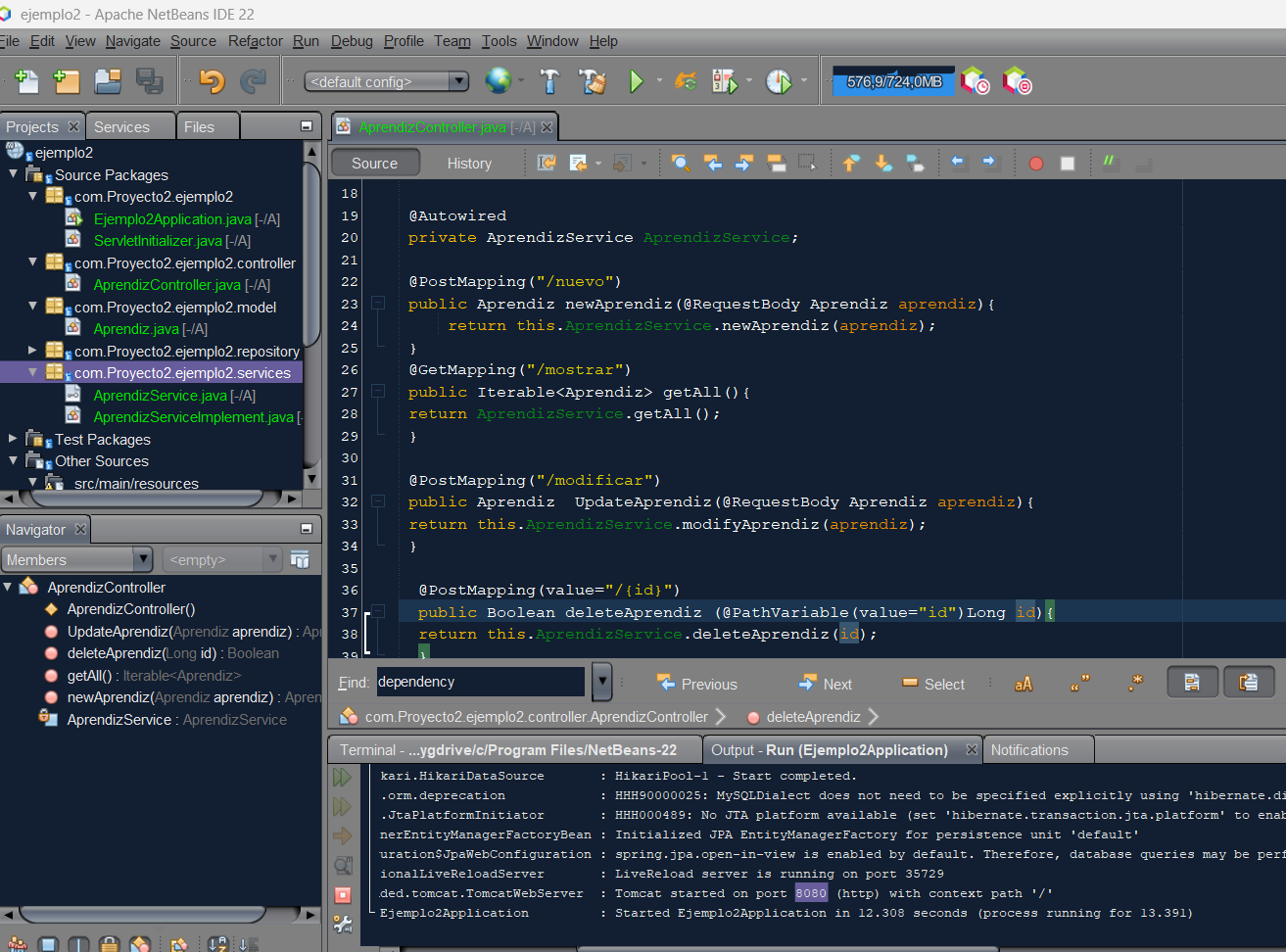


Se instala, va a pedir cómo queremos iniciar sesión, colocamos que con Google.

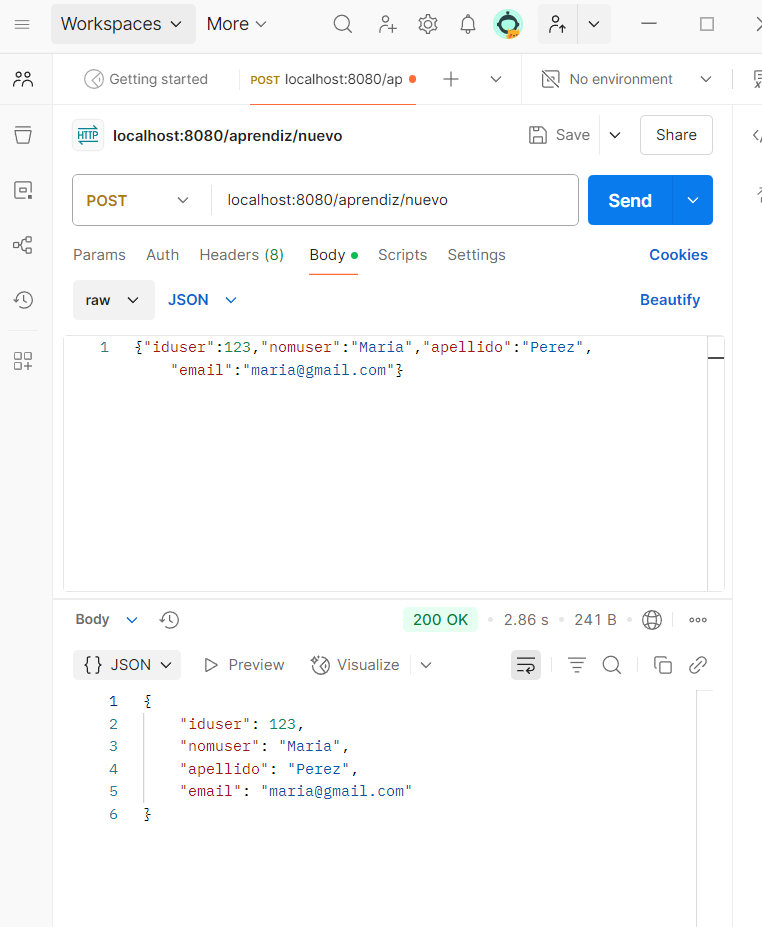


Se crea una nueva pestaña, se coloca de tipo POST localhost:8080/aprendiz/nuevo

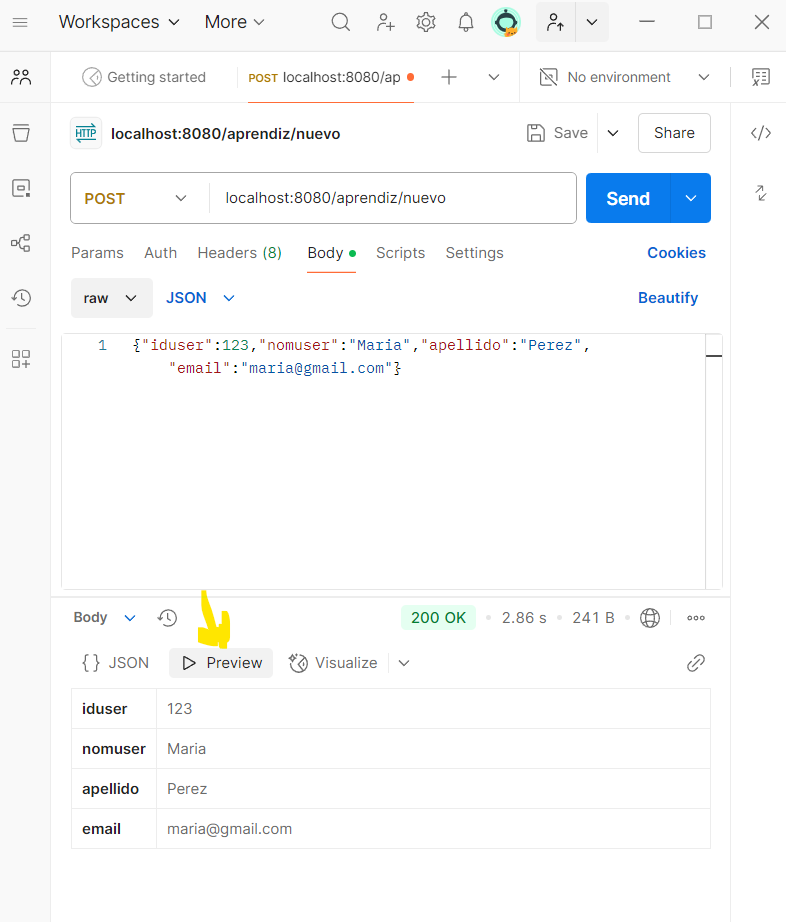
Y copiamos los datos para ingresar un nuevo aprendiz, teniendo en cuenta la ventana “AprendizController.java” ubicada dentro de “com.Proyecto2.ejemplo2.services “



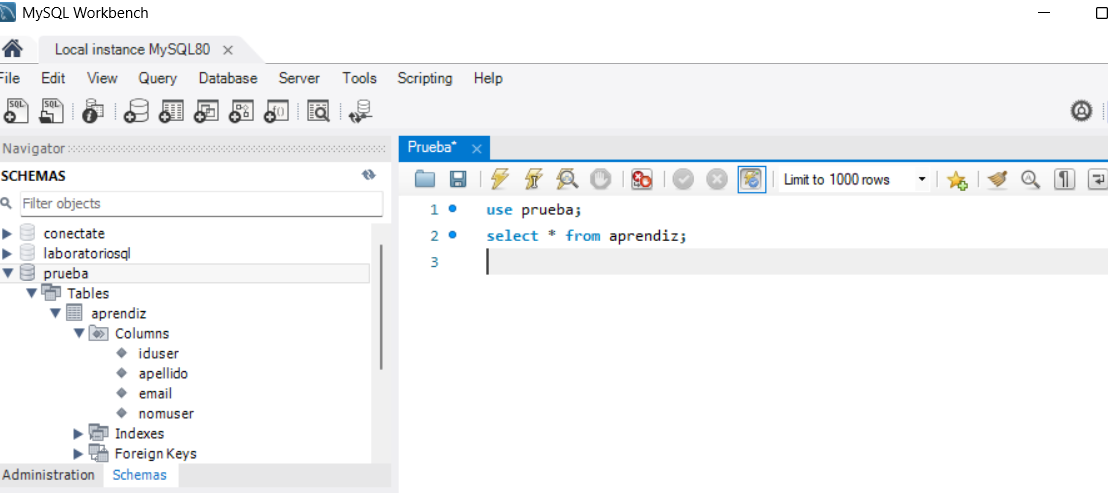
Si se está haciendo la conexión se oprime el botón SEND



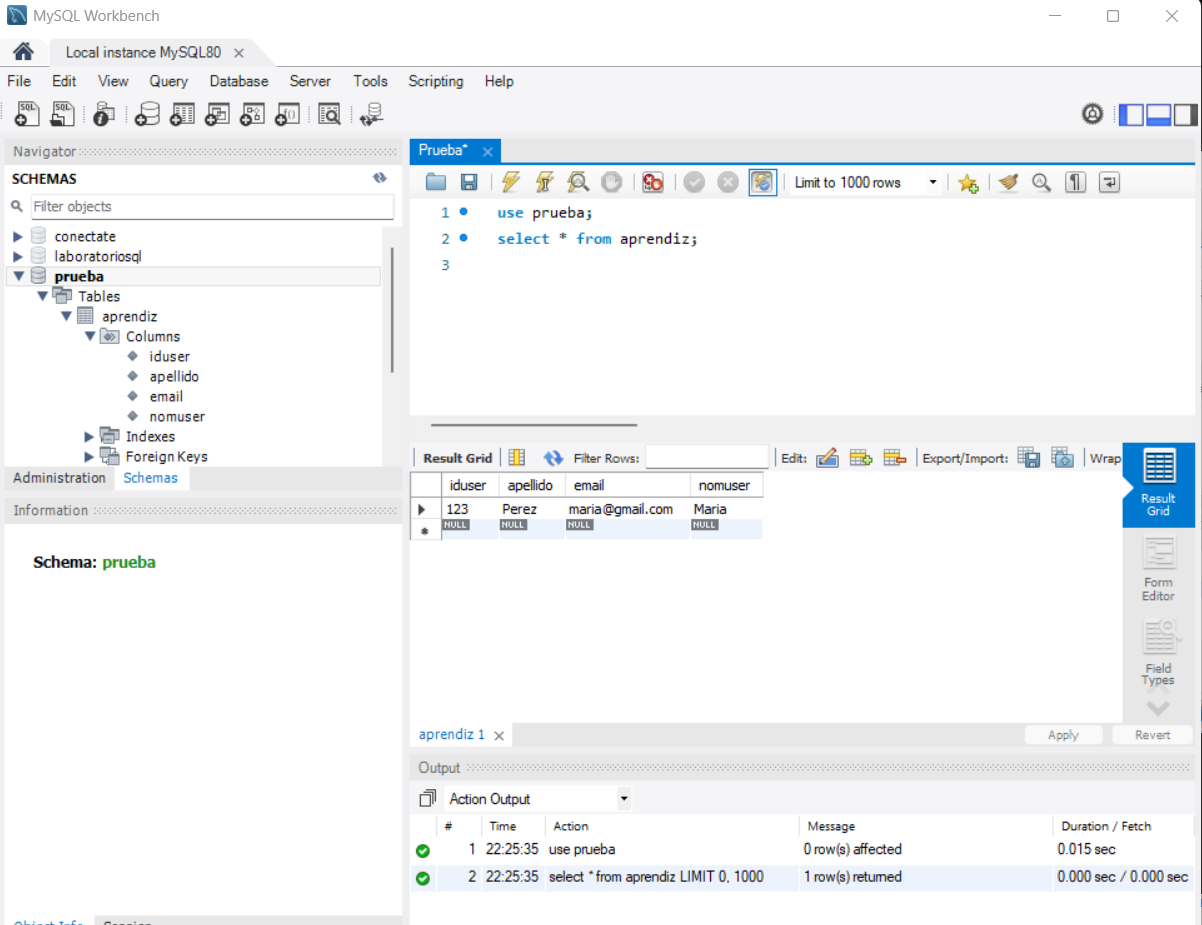
Se oprime Preview para tener una vista previa de los resultados.

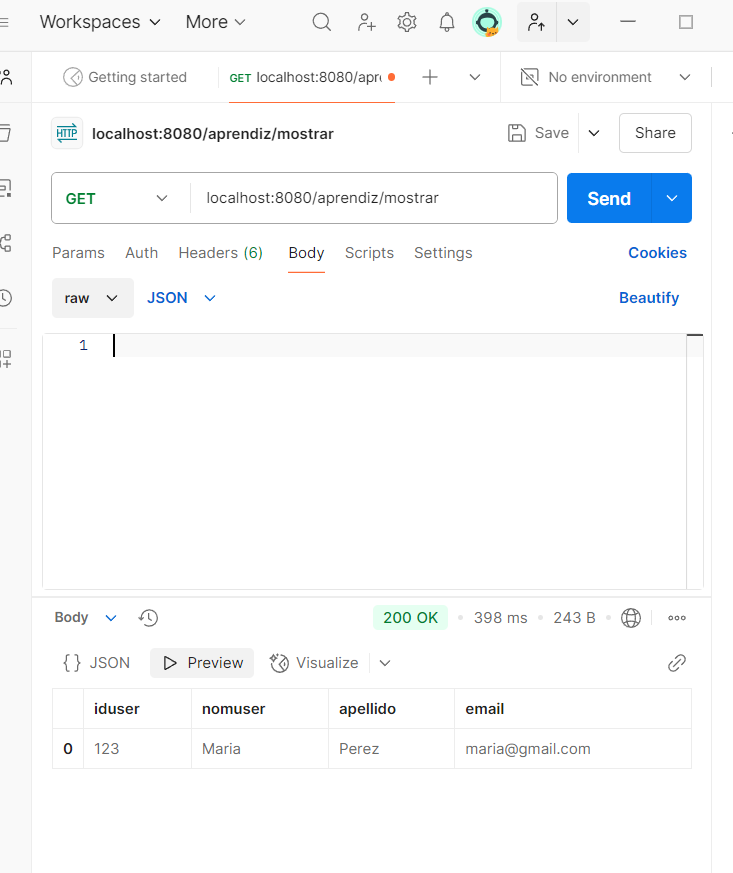


En Workbench colocamos las siguientes sentencias para que me muestre todo lo que haya en la base de datos prueba tabla aprendiz

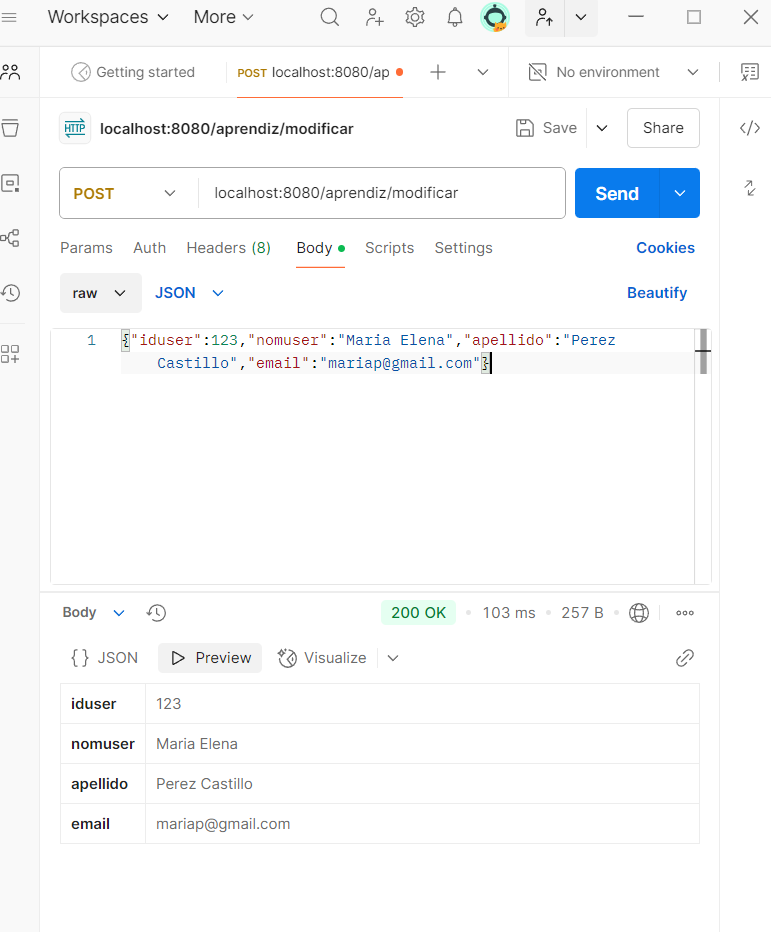


Lo cual da como resultado lo siguiente:





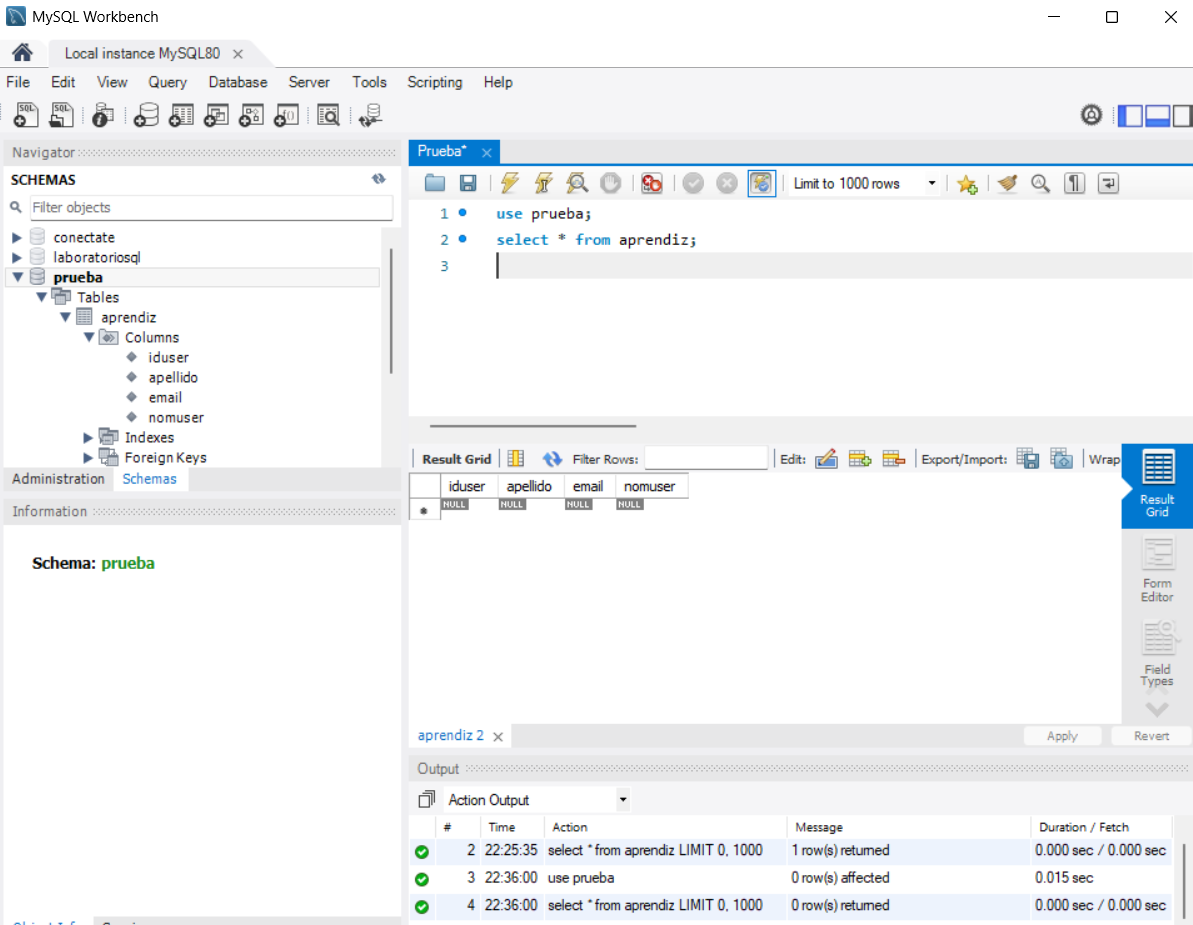
Se realiza la prueba con método GET y modificando el vinculo para al final colocar mostrar, lo que muestra la información en base de datos.

Se utiliza el método POST para modificar datos en la base de datos, como se muestra a continuación: 

Para el proceso de eliminar colocamos el id, el cual en el ejemplo es 123, con el método POST localhost:8080/aprendiz/123



Esto me arroja el valor booleano de true. Lo que quiere decir que borra el registro.



En workbench revisamos que efectivamente el registro se borró de la base de datos.

Enlace al repositorio GIT hub.

<https://github.com/Cirales/StandAlone/commits/main/>