

INFORME DE INVESTIGACIÓN/TRABAJO AUTÓNOMO

1. Datos Generales:

Carrera:	Tecnología Superior en Big Data
Período académico:	Mayo – Octubre 2022
Asignatura:	Tendencias Actuales de Programación
Unidad N°:	2
Tema:	API REST Full
Ciclo-Paralelo:	M5B
Estudiante:	Mayra Peñañiel
Docente:	Ing. Diego Cale
Fecha:	01/08/2022

2. Contenido:

2.1 Introducción:

Spring Boot es una tecnología que nos permite crear aplicaciones autocontenidas, con esto nos podemos olvidar de la arquitectura y enfocarnos únicamente en desarrollo, delegando a Spring Boot labores como configuración de dependencias, desplegar nuestro servicio o aplicación a un servidor de aplicaciones y enfocarnos únicamente en crear nuestro código.

Para esto Spring Boot utiliza internamente un servidor de aplicaciones embebido, por defecto utiliza Tomcat, pero también lo podemos hacer con jetty o undertow.

2.2 Objetivo:

Crear una API REST-Full que implemente el servicio CRUD con la base de datos MySQL con la ayuda de springboot.

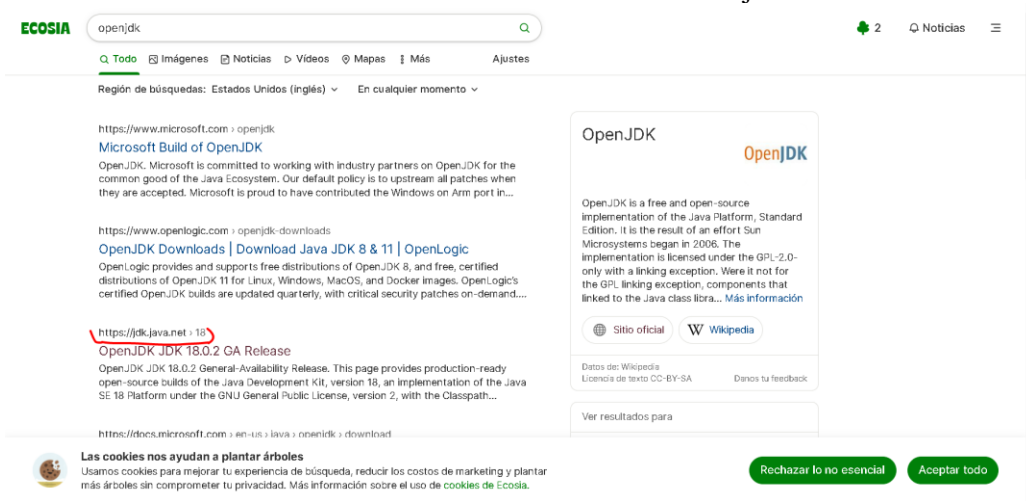
2.3 Materiales, herramientas, equipos y software:

Computador personal, IDE NetBeans, Lenguaje de Programación Java, Springboot Tools, Workbench MySQL.

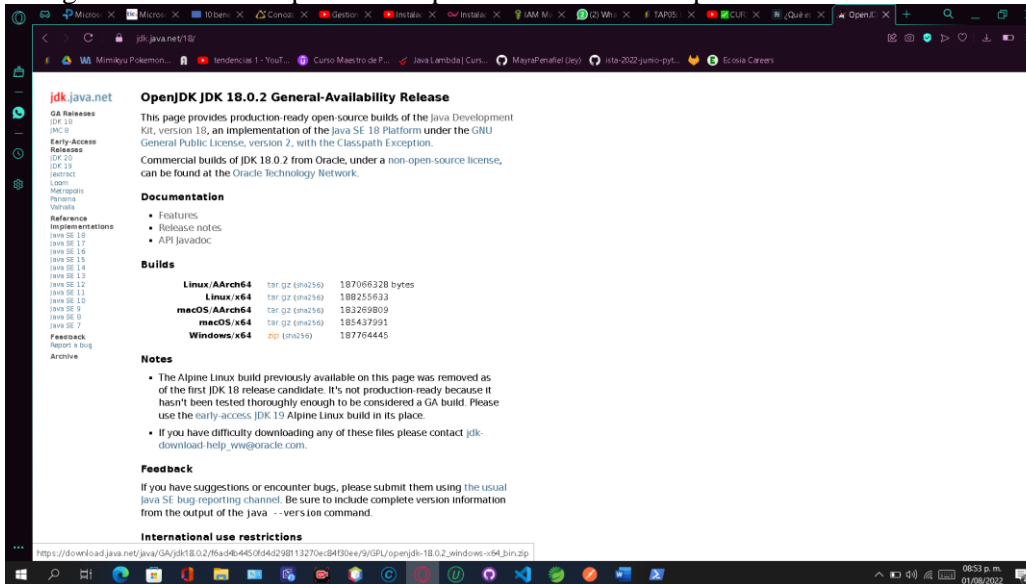
2.4 Procedimiento

- **Descargar e instalar el openjdk:**

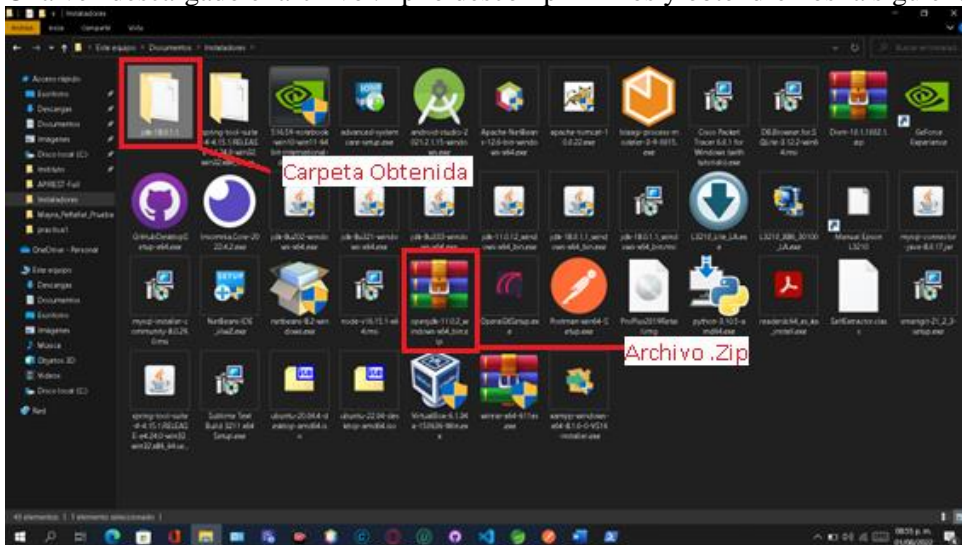
Buscar en su buscador de preferencia openjdk y descargar openjdk la versión mas reciente y estable posible en este caso se uso la 18.0.1.1 entramos al enlace con dominio java.net :



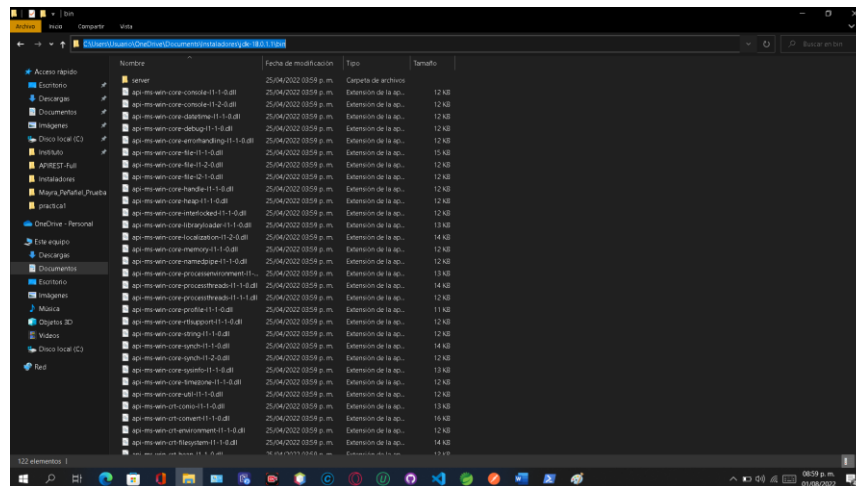
Escogemos el archivo zip adecuado para nuestro sistema operativo en este caso windows:



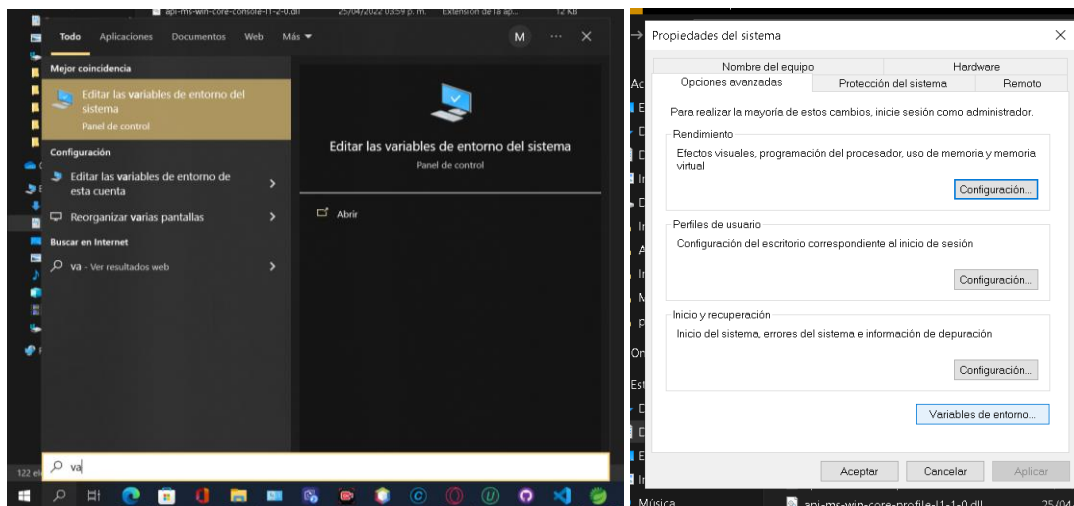
Una vez descargado el archivo .zip lo descomprimos y obtendremos la siguiente carpeta:



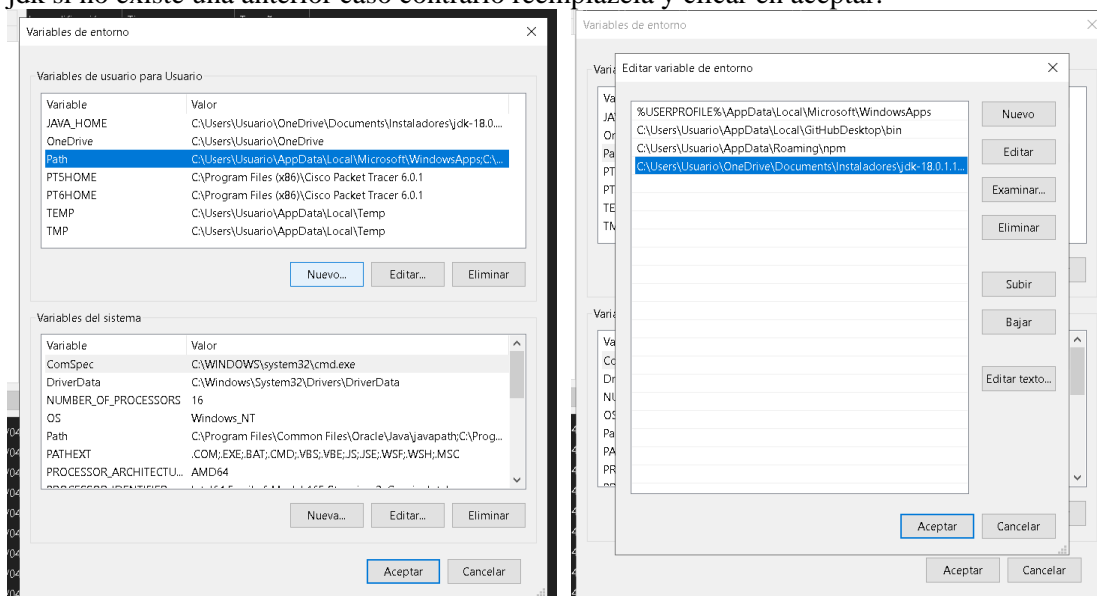
Entrar a la carpeta descomprimida y luego a la carpeta bin y luego copiar la dirección del enlace



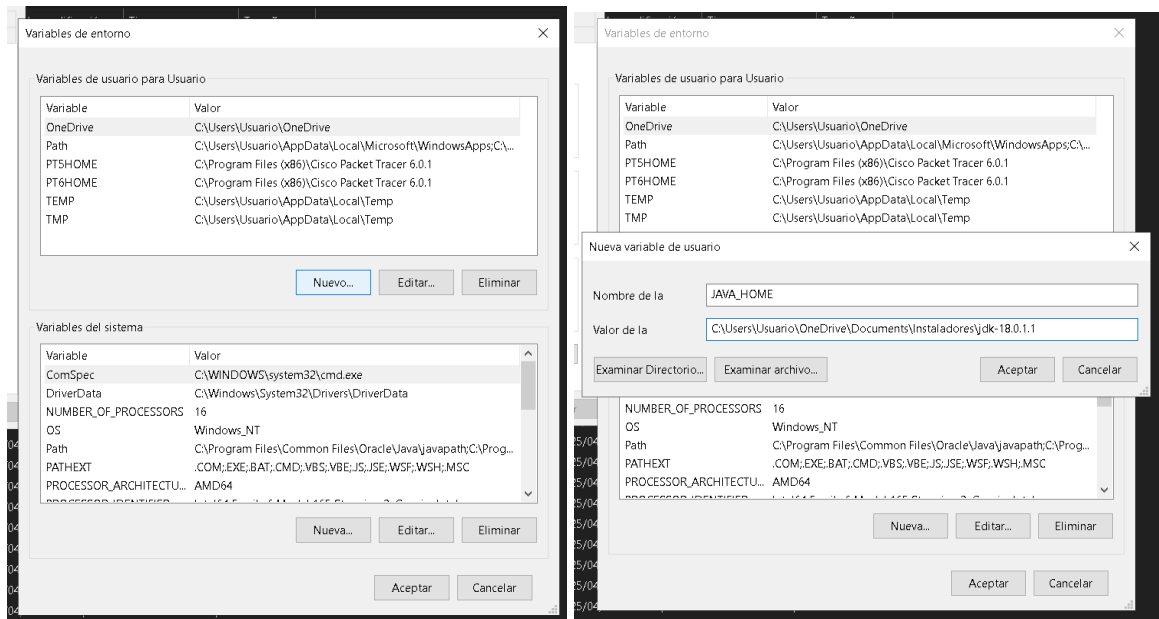
Buscar en variables de entorno del sistema:



Una vez aparezca la siguiente pantalla seleccionar Path y pegar el enlace que copiamos anteriormente del jdk si no existe una anterior caso contrario reemplazela y clicar en aceptar:



Luego hacemos click en el botón nuevo Y creamos la Variable “JAVA_HOME” y en valor de a variable gamos el mismo enlace pero sin “/bin” es decir solo hasta la versión del jdk en este caso 18.0.1.1 y damos click al botón aceptar hasta cerrar la pantalla:



Para Comprobar la versión de Java ejecutamos la sentencia “java -version” y “javac -version” en la consola de Windows PowerShell ambas versiones deben coincidir.

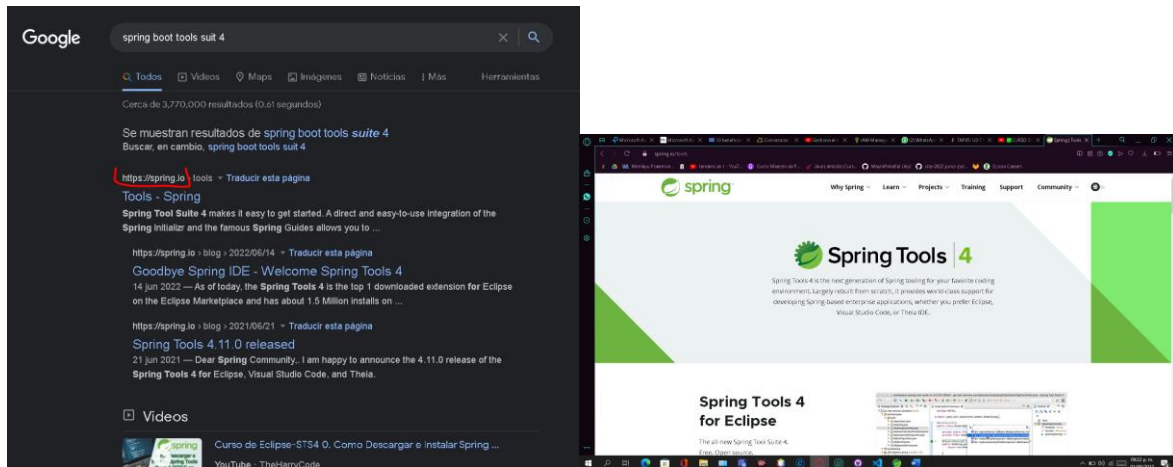
```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Prueba la nueva tecnología PowerShell multiplataforma https://aka.ms/pscore6

PS C:\Users\Usuario> java -version
java version "18.0.1.1" 2022-04-22
Java(TM) SE Runtime Environment (build 18.0.1.1+2-6)
Java HotSpot(TM) 64-Bit Server VM (build 18.0.1.1+2-6, mixed mode, sharing)
PS C:\Users\Usuario> javac -version
javac 18.0.1.1
PS C:\Users\Usuario>
```

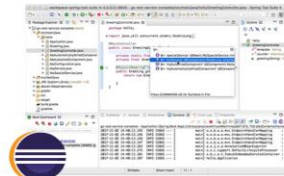
- **Descargar e instalar SpringBoot Tools Suit 4:**

Escribir en su buscador Springboot Tools Suit 4 y entrar en el dominio “spring.io” y seleccionamos Spring tools 4 for Eclipse de acuerdo a nuestro sistema operativo en este caso Windows y se descargara un archivo .jar:



Spring Tools 4 for Eclipse

The all-new Spring Tool Suite 4.
Free. Open source.



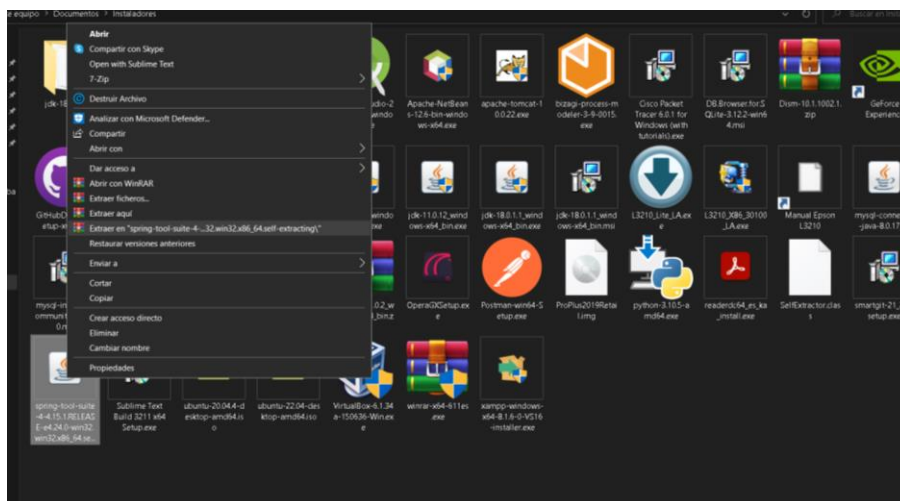
4.15.1 - LINUX X86_64

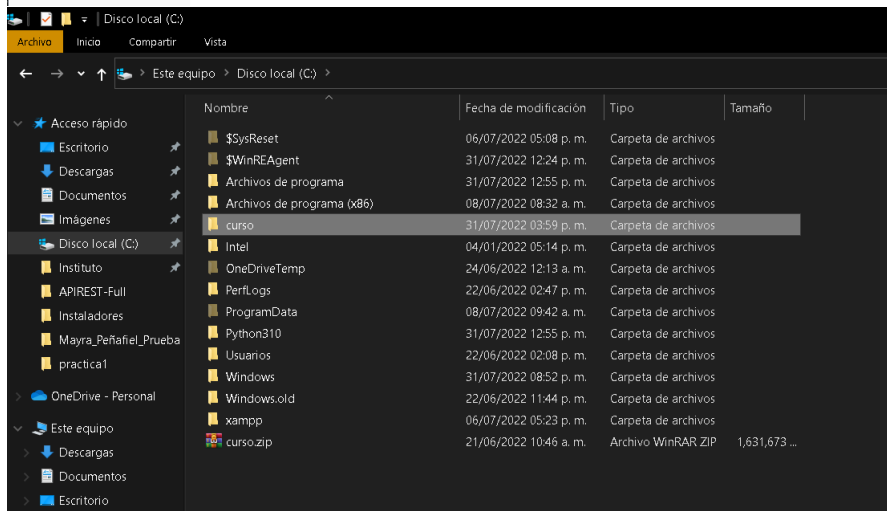
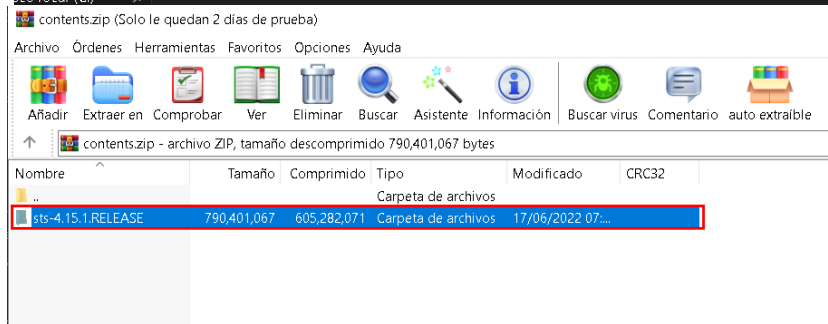
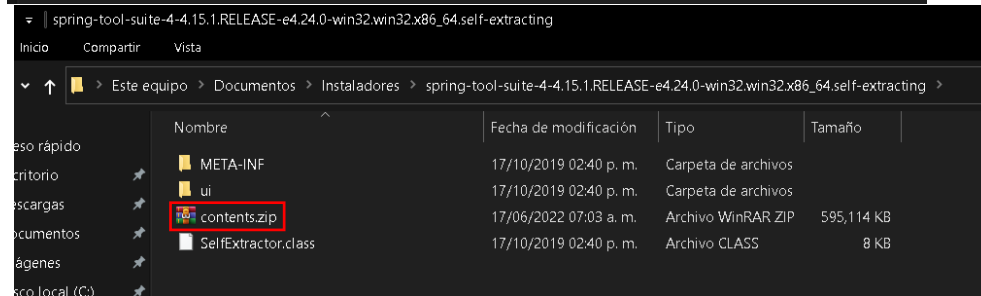
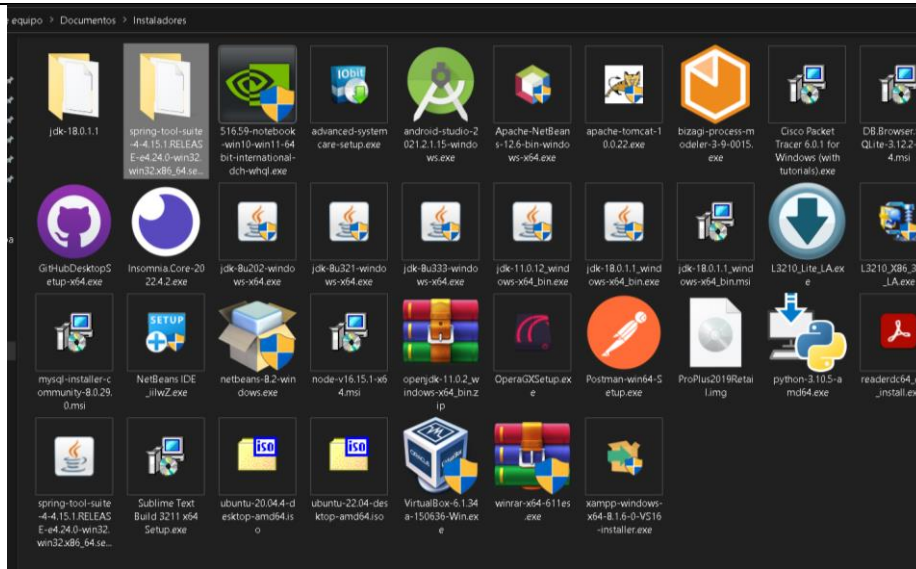
4.15.1 - MACOS X86_64

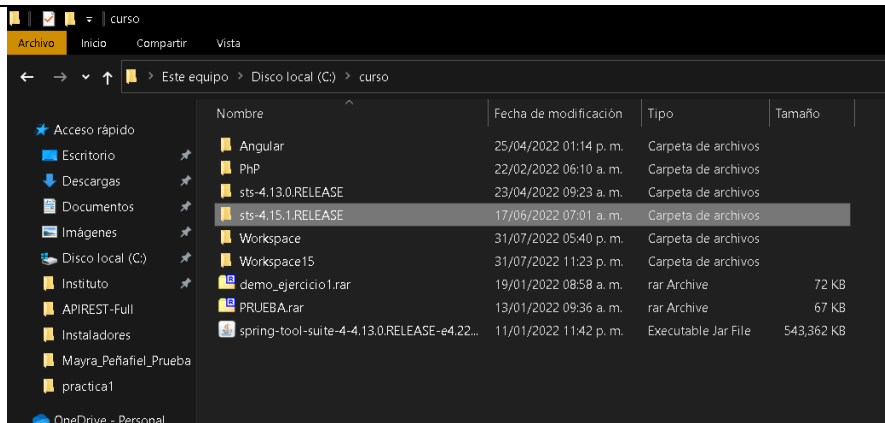
4.15.1 - MACOS ARM_64

4.15.1 - WINDOWS X86_64

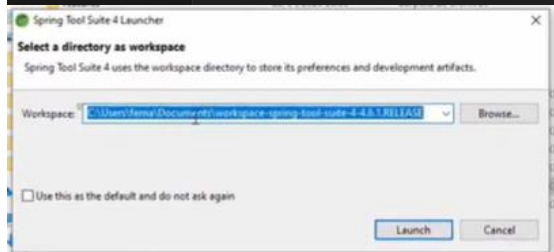
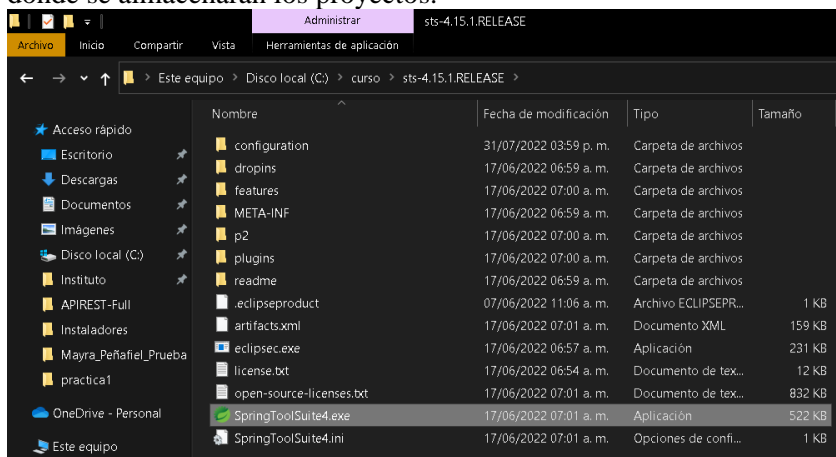
Descomprimos el archivo .jar y dentro de esta carpeta encontraremos otra carpeta comprimida llamada contents, dentro de esta encontramos la carpeta “sts-4.15.1.RELEASE” arrastramos esta carpeta al lugar donde crearemos nuestro proyecto en este caso he creado una carpeta en el disco local C llamada “curso”:



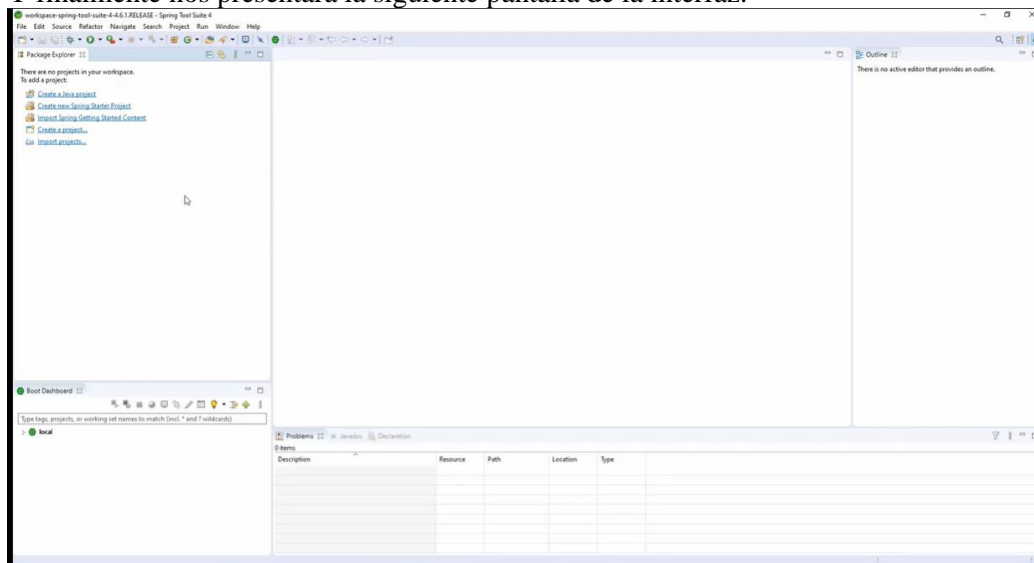




Dentro de la carpeta curso creamos otra carpeta donde se almacenaran todo los proyecto quevayamos a crear en spring en este caso se llama “Workspace 15”, luego abriremos la carpeta que arrastramos y ejecutamos el archivo “SpringToolSuite4.exe” y se nos mostrara una pantalla donde nos pedirá la dirección de la carpeta donde se almacenaran los proyectos:

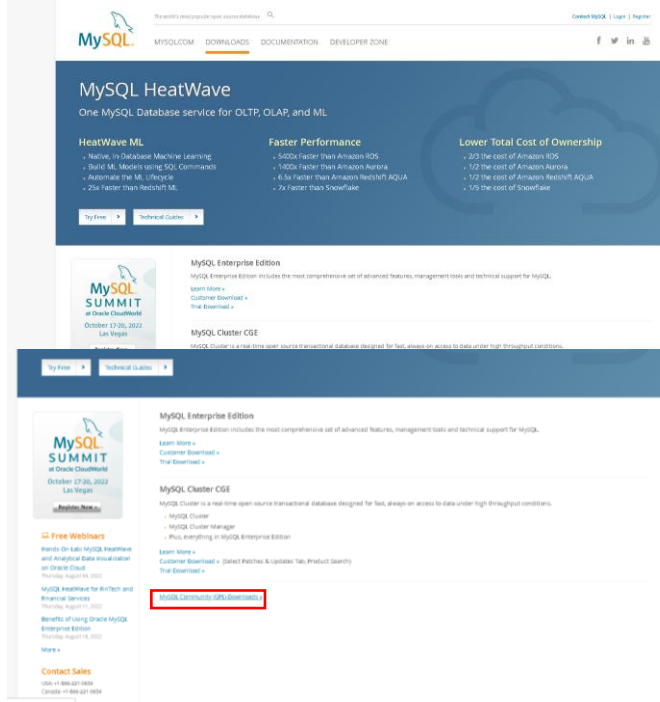
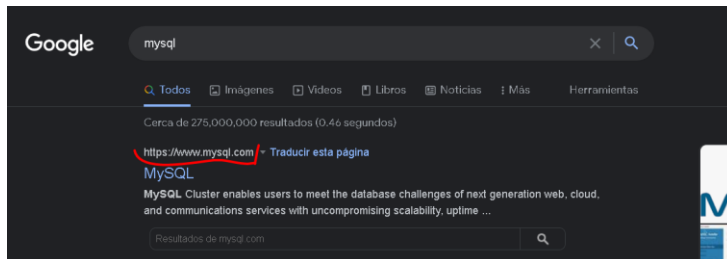


Y finalmente nos presentara la siguiente pantalla de la interfaz:



- Descarga e instalación de Workbench MySQL:

Escribimos en el buscador “MySQL” y entramos al enlace con el dominio mysql.com, luego le damos a dowlands, escogemos la opción MySQL Community (GPL) Downlands, escogemos MySQL community Server, vamos al instalador y finalmente descargamos la segunda opción del MSI :

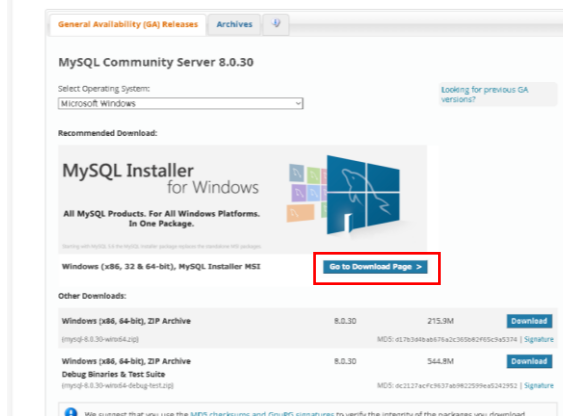


MySQL Community Downloads

- MySQL Yum Repository
- MySQL APT Repository
- MySQL SUSE Repository
- MySQL Community Server**
- MySQL Cluster
- MySQL Router
- MySQL Shell
- MySQL Operator
- MySQL Workbench
- MySQL Installer for Windows
- MySQL for Visual Studio
- C API (libmysqlclient)
- Connector/C++
- Connector/J
- Connector/NET
- Connector/Node.js
- Connector/ODBC
- Connector/Python
- MySQL Native Driver for PHP
- MySQL Benchmark Tool
- Time zone description tables
- Download Archives

MySQL Community Downloads

MySQL Community Server

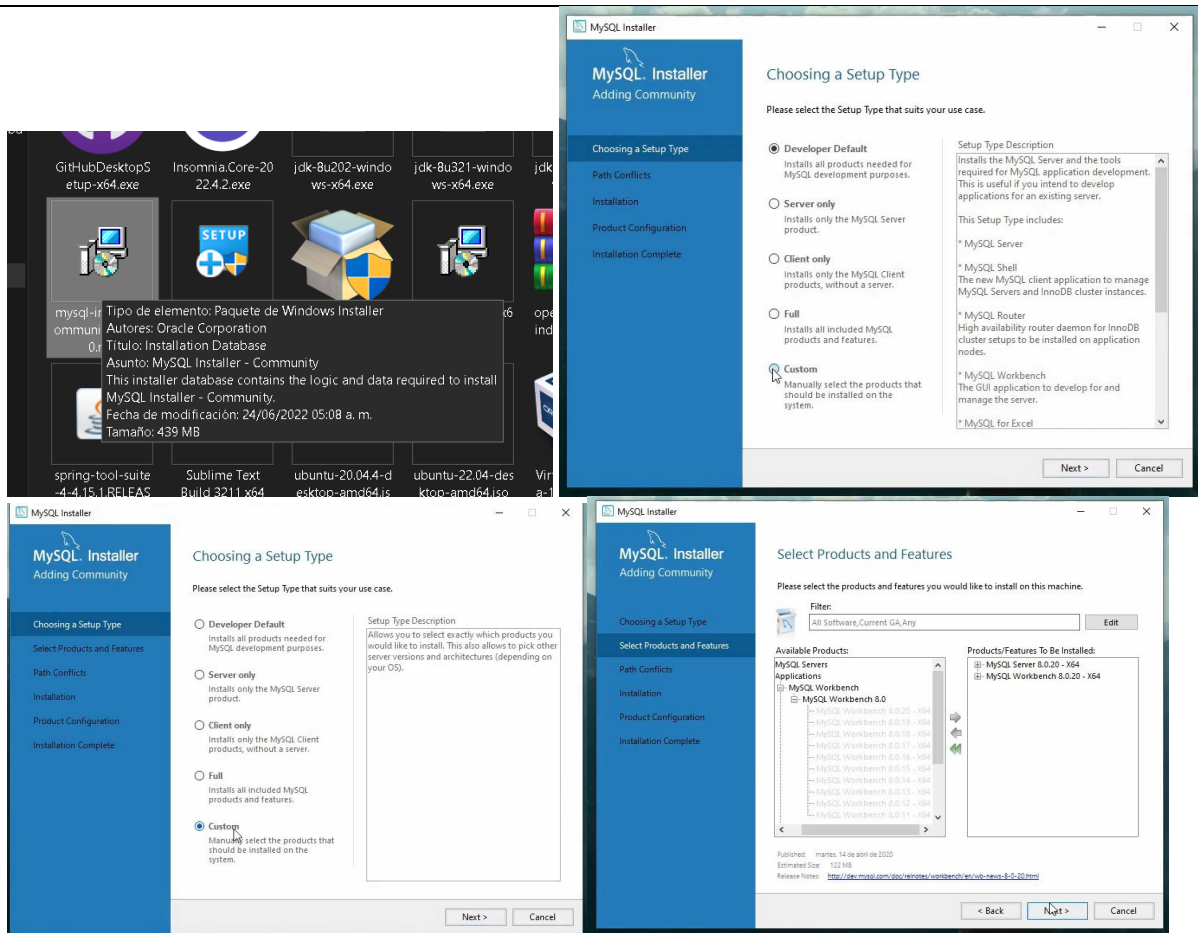


MySQL Community Downloads

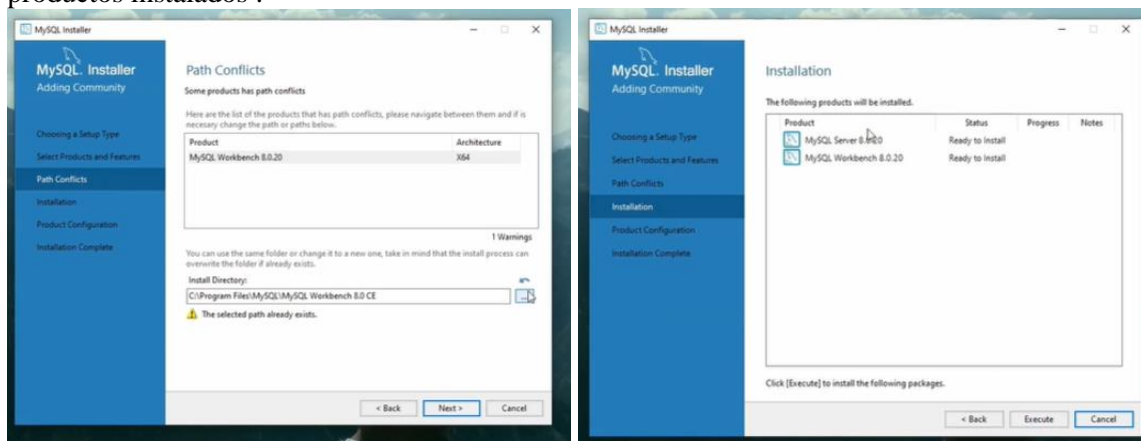
MySQL Installer

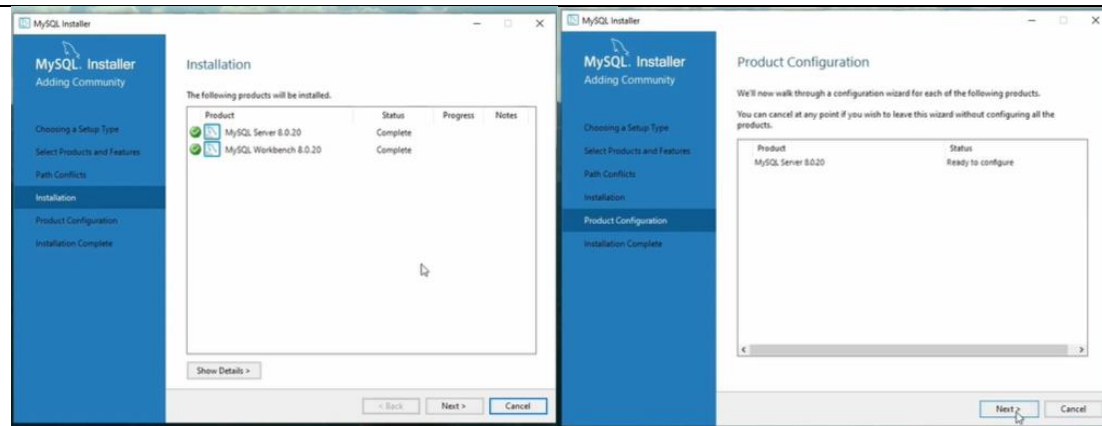


Se descargara un instalador o archivo msi, lo ejecutamos y damos click en ejecutar a lo que saldrá la siguiente pantalla, seleccionamos Custom para una instalación personalizada, seleccionamos los productos “MySQL Server (Version instalada)” y “MySQL Workbench (Version Instalada)”:

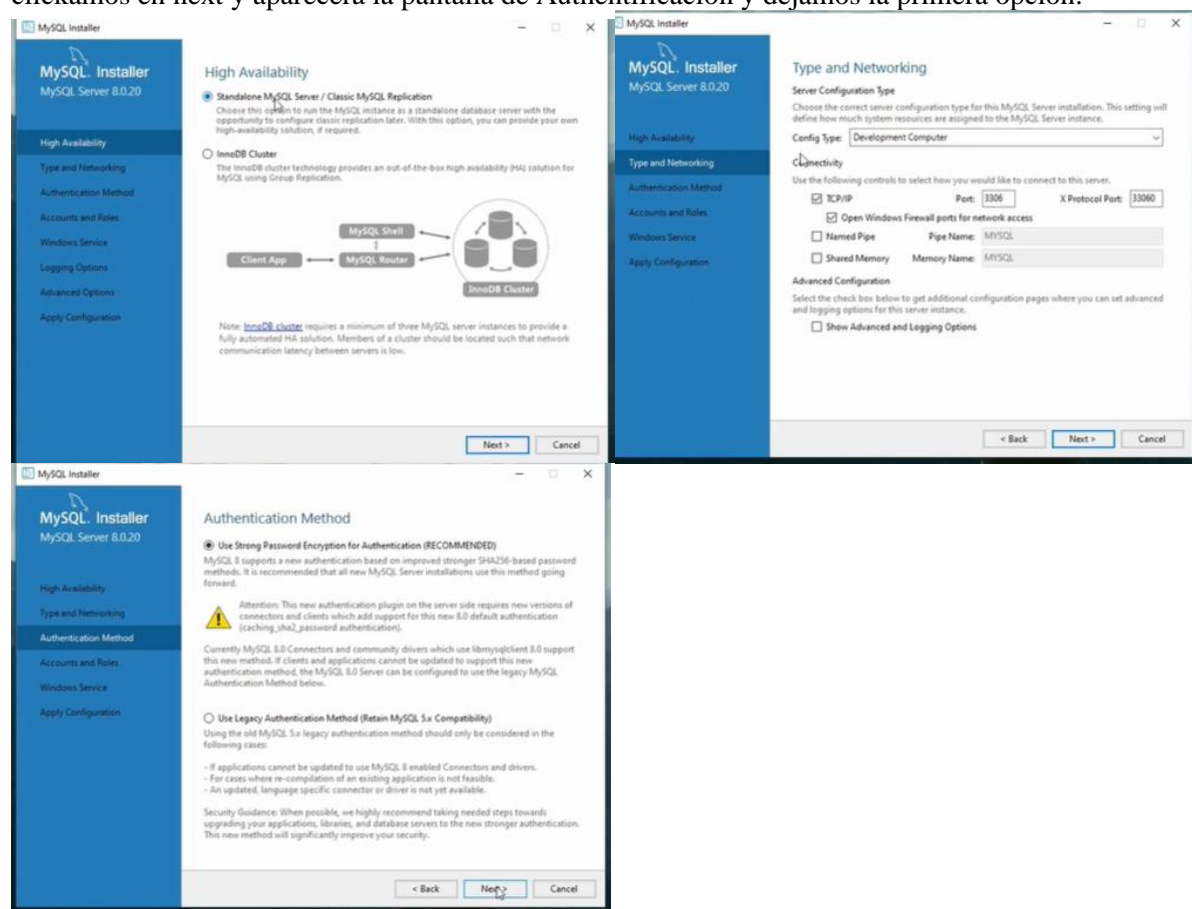


Seleccionamos la carpeta donde se instalara el programa, procedmos a instalar los productos que esogimos anteriormente, clickamos execute, una vez completado clickamos en next y empezaremos a configurar los productos instalados :

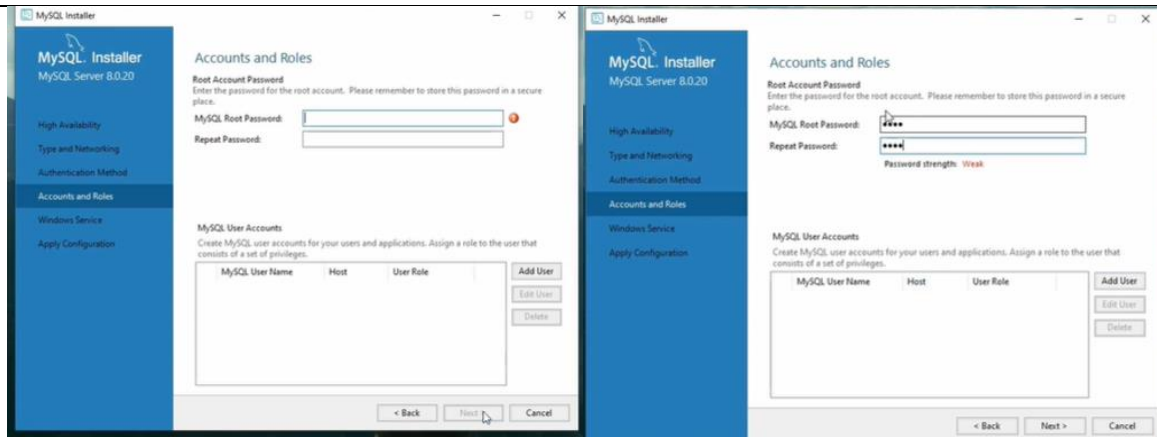




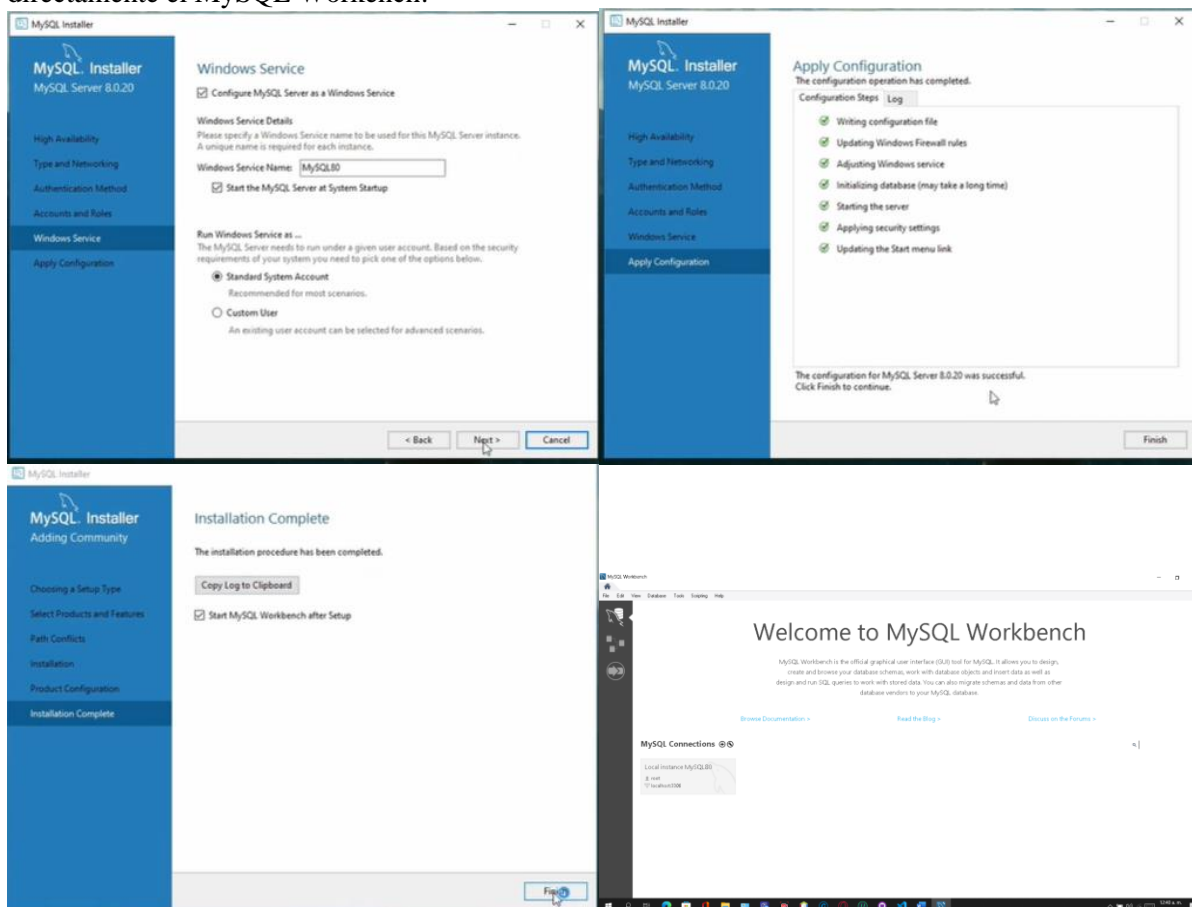
La siguiente pantalla suele ser opcional de High Availability pero de aparecer seleccionamos “Standalone MySQL Server / Classic MySQL Replication”, de lo contrario aparecerá la siguiente pantalla de Type and Networking donde escogemos “Development Computer” y dejamos las demás opciones por default, clickamos en next y aparecerá la pantalla de Authentication y dejamos la primera opción:



Pasamos a la Pantalla de Accounts and Roles donde podemos crear una contraseña para la cuenta root que viene predefinida, escoger si crear o no un nuevo usuario:

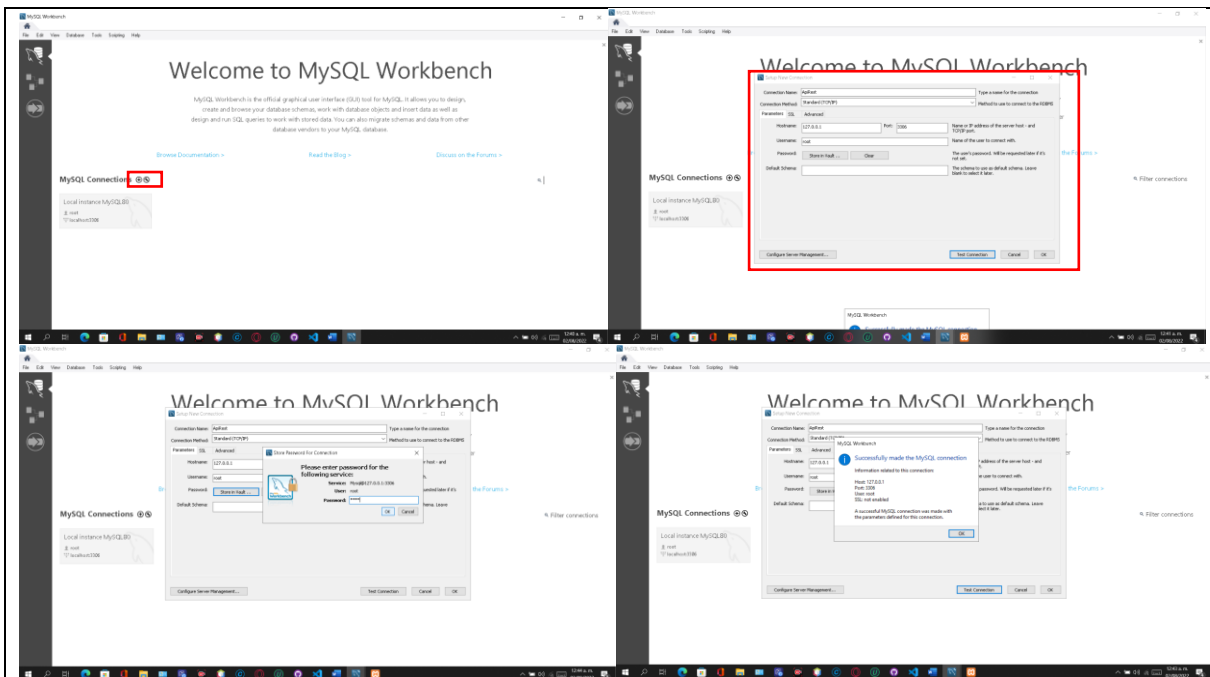


Damos click en next y entramos a la pantalla de Windows Service, dejamos las opciones predefinidas, aquí se puede definir el nombre del servicio del programa y como queremos que se ejecute el servicio finalmente pasamos a la última ventana de Apply Configuration y la ventana final donde nos permitirá ejecutar directamente el MySQL Workbench:

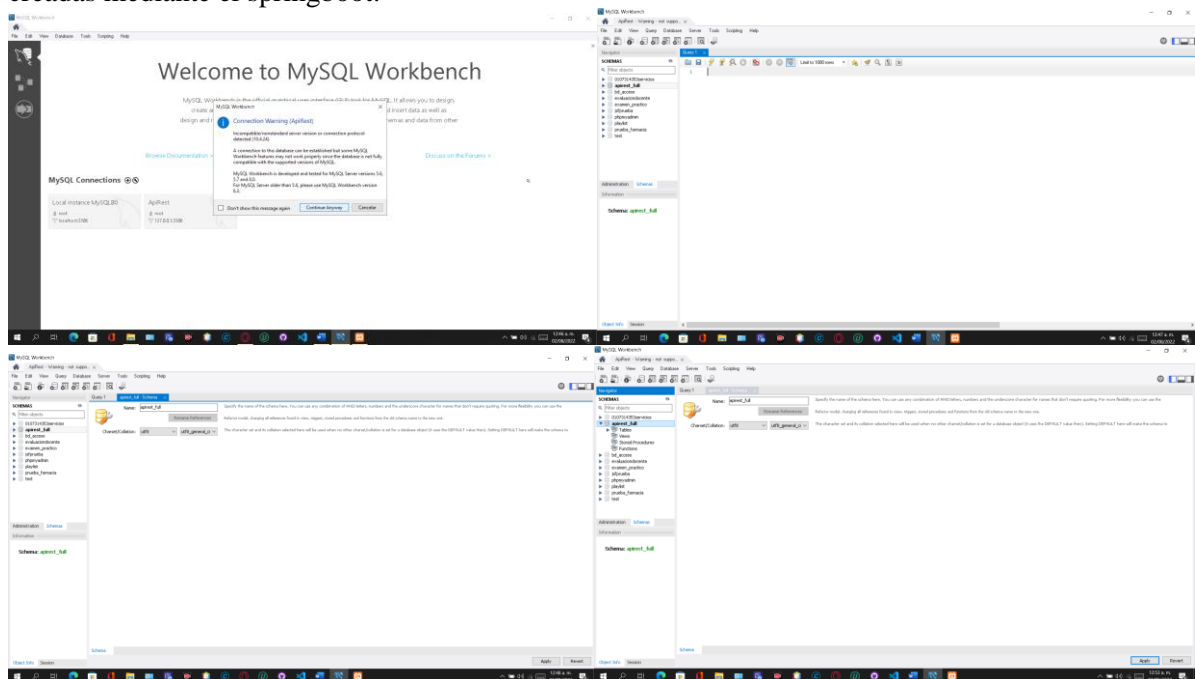


- Crear la Base de Datos:

Abrimos/ ejecutamos el programa de Workbench My SQL creamos una nueva conexión dando click al botón con un símbolo de mas "+", ponemos un nombre a la conexión, seleccionamos el usuario y ponemos la clave de preferencia que quede almacenada para ahorrar tiempo, dar click en el botón test Connection para comprobar que se haya conectado correctamente.:

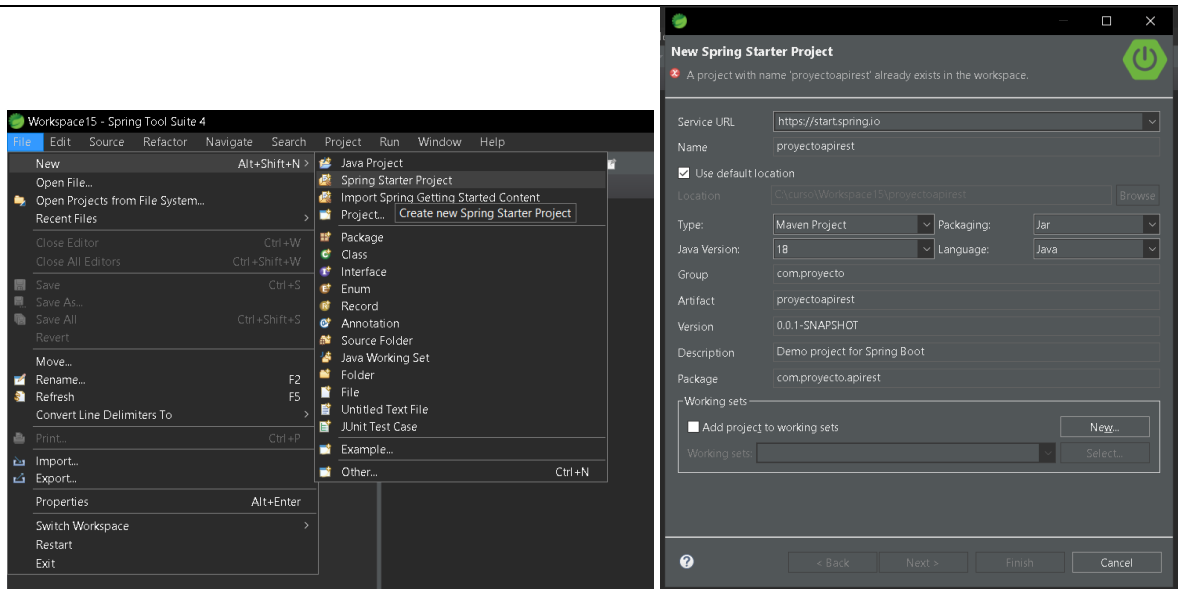


Abrimos la nueva conexion que creamos, creamos un nuevo Schema o Base de Datos ya sea por consulta query o con los botones de acceso directo del programa, seleccionamos como charset/Collation (utf8 / utf8_general_ci) en este caso se llamara “apirest_full” por el momento quedara vacia ya que las tablas serán creadas mediante el springboot:

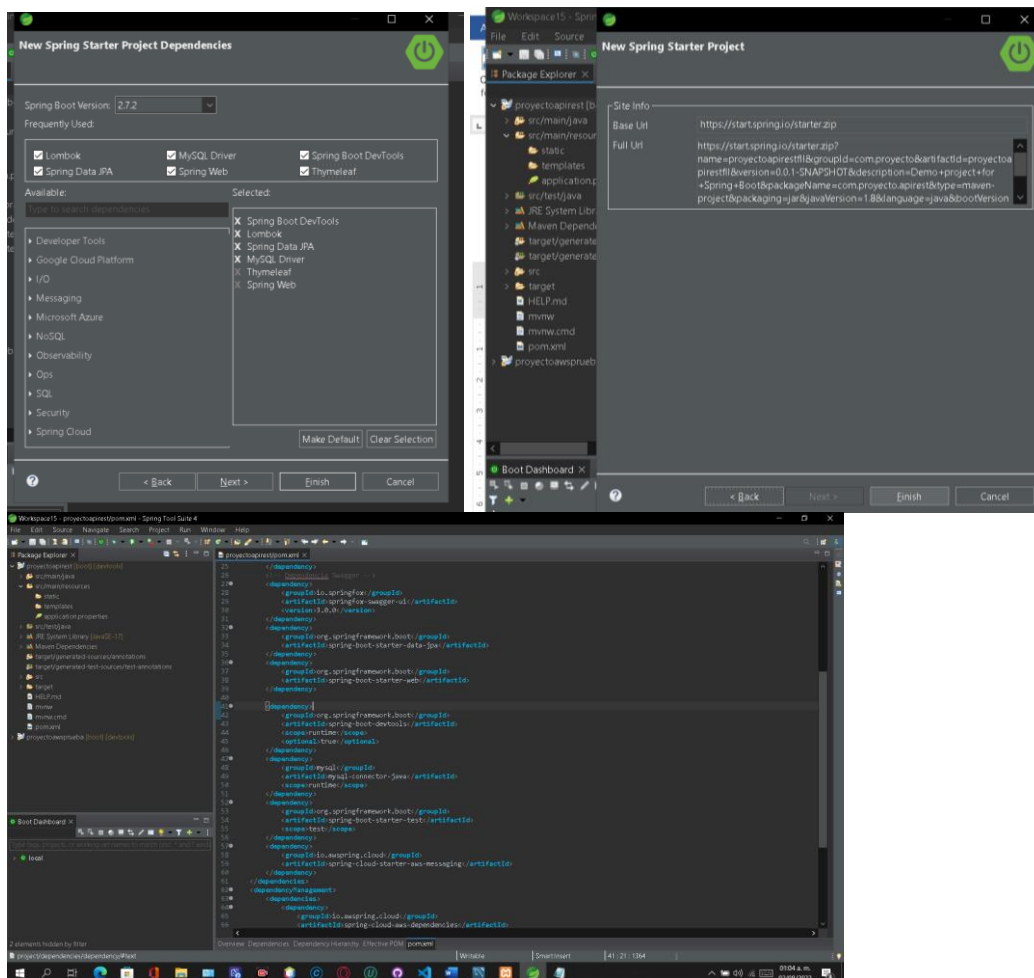


- Crear un Proyecto Api Rest Full:

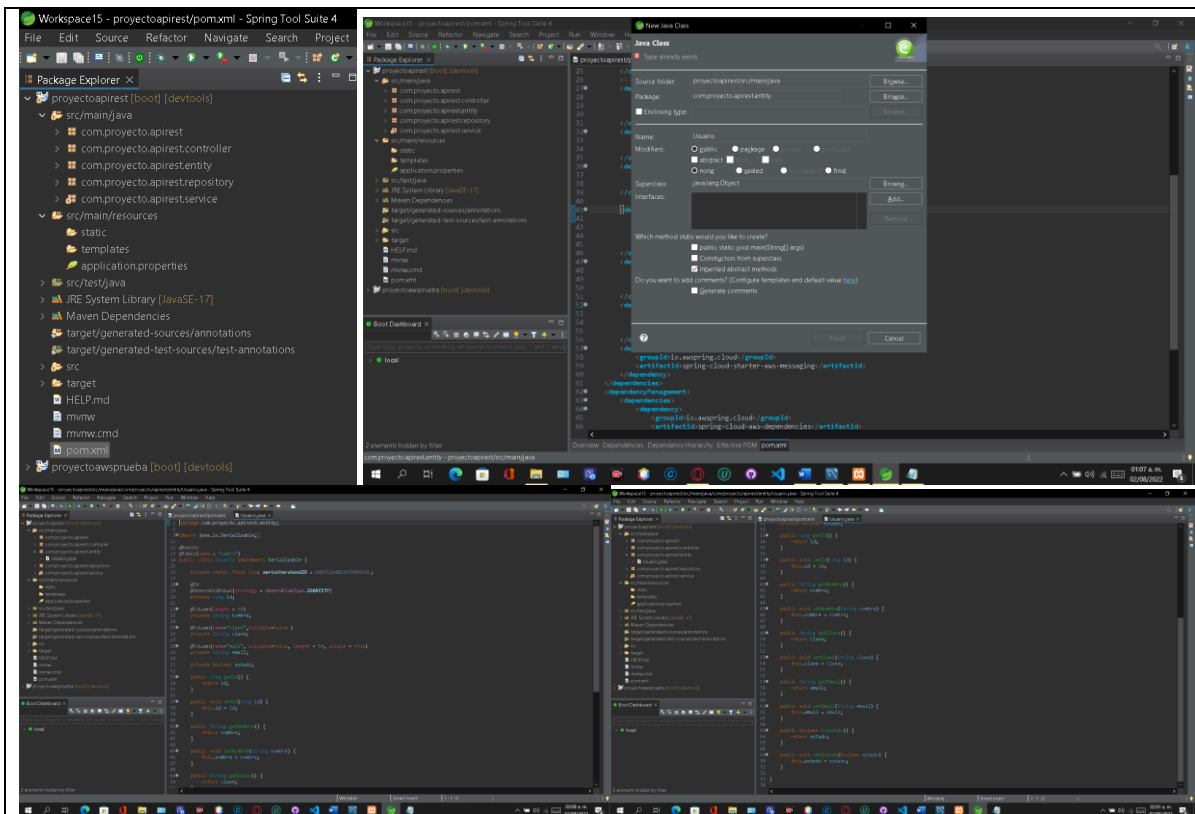
Damos clic a la opción Create new Spring Starter Project y nos aseguramos de llenar el nombre del proyecto, que sea tipo “Maven”, empaquetado “jar”, versión mas actual instalada del jdk y descripción general del proyecto:



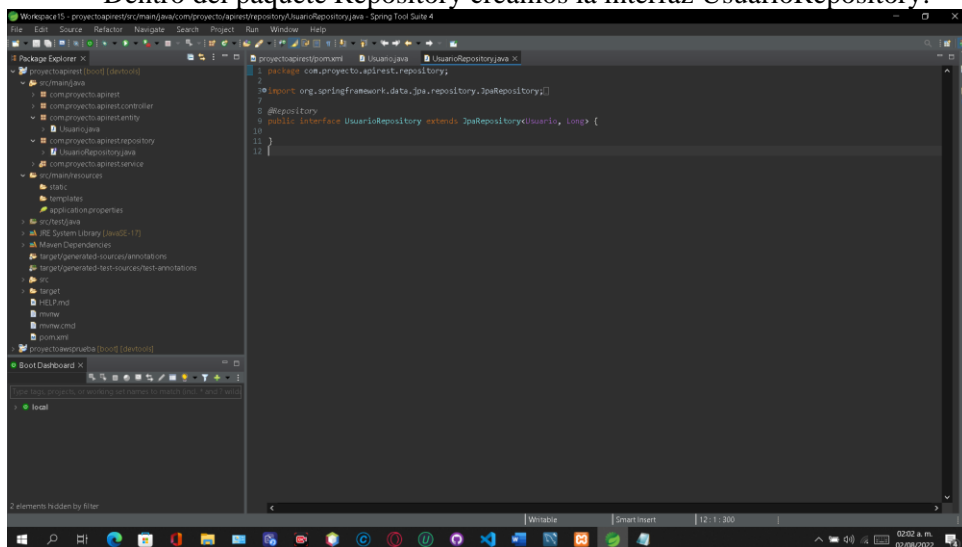
Procedemos a escoger las librerías y dependencias a utilizar y le damos a finalizar, una vez hecho esto podemos comprobar que si se realizó correctamente la información de cada dependencia:



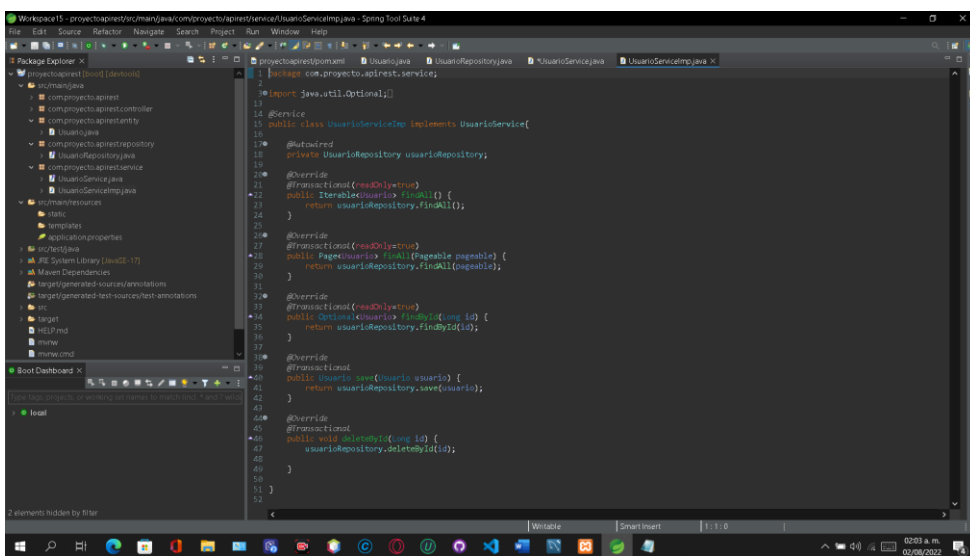
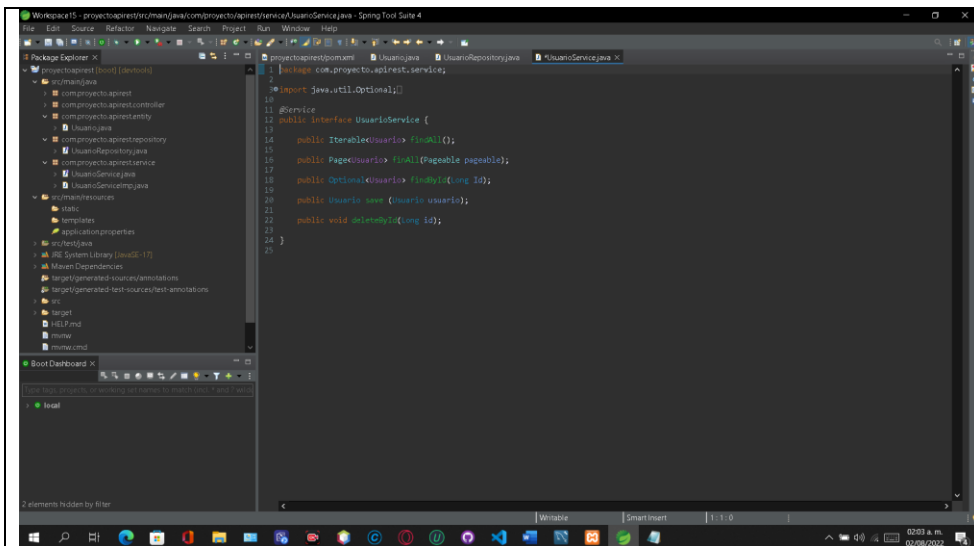
Creamos los paquetes de “Controller, Service, Model/Entity y Repository” y creamos una clase entity para usuario donde crearemos los atributos de la tabla usuario:



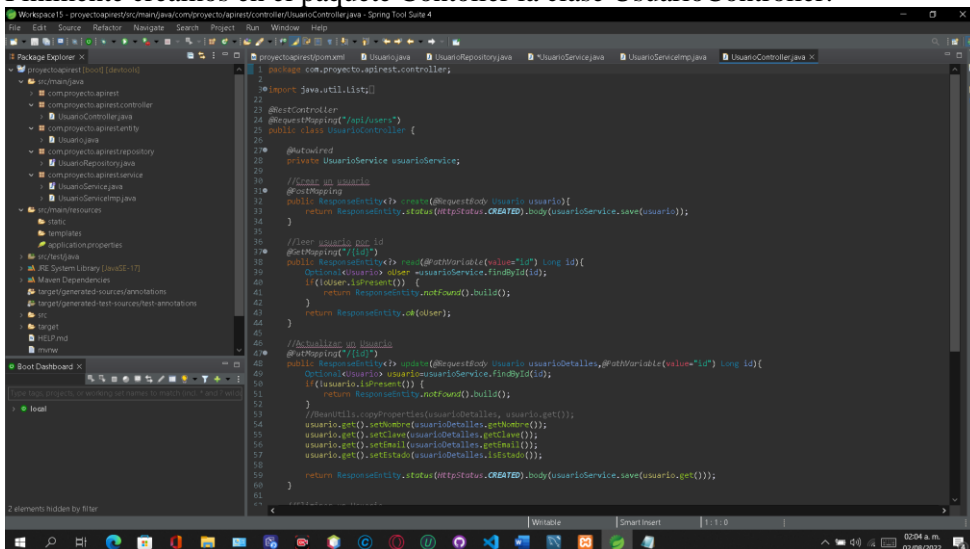
Dentro del paquete Repository creamos la interfaz UsuarioRepository:



Dentro del paquete Service se crea la interfaz de UsuarioService y luego la clase UsuarioServiceImp:



Finlmente creamos en el paquete Contoller la clase UsuarioController:



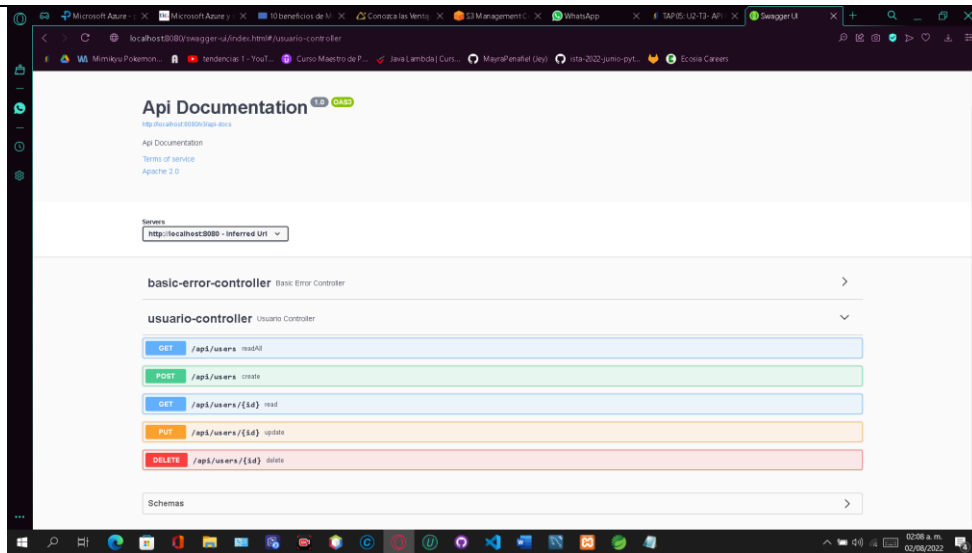
```
41 }
42 return ResponseEntity.ok(user);
43 }
44
45 //Actualizar un Usuario
46 @PostMapping("/{id}")
47 public ResponseEntity<?> update(@RequestBody Usuario usuarioDetalles, @PathVariable("id") Long id){
48     return usuarioService.findById(id);
49     if(usuario.isPresent()) {
50         return ResponseEntity.notFound().build();
51     }
52 }
53 //Buscar todos los usuarios
54 @GetMapping
55 public List<Usuario> readAll(){
56     List<Usuario> usuarios = usuarioRepository.findAll();
57     return usuarios;
58 }
59
60 //Eliminar un Usuario
61 @DeleteMapping("/{id}")
62 public ResponseEntity<?> delete(@PathVariable("id") Long id){
63     if(usuarioService.findById(id).isPresent()) {
64         return ResponseEntity.notFound().build();
65     }
66     usuarioService.delete(id);
67     return ResponseEntity.ok().build();
68 }
69
70 //Crear un Usuario
71 @PostMapping
72 public ResponseEntity<?> create(@RequestBody Usuario usuarioDetalles){
73     List<Usuario> usuarios = usuarioRepository.findAll();
74     if(usuarios.contains(usuarioDetalles)){
75         return ResponseEntity.notFound().build();
76     }
77     return ResponseEntity.ok(usuarioService.save(usuarioDetalles));
78 }
79
80 //Eliminar un Usuario
81 @DeleteMapping("/{id}")
82 public ResponseEntity<?> delete(@PathVariable("id") Long id){
83     if(usuarioService.findById(id).isPresent()) {
84         return ResponseEntity.notFound().build();
85     }
86     usuarioService.delete(id);
87     return ResponseEntity.ok().build();
88 }
89
90 //Buscar todos los usuarios
91 @GetMapping
92 public List<Usuario> readAll(){
93     List<Usuario> usuarios = usuarioRepository.findAll();
94     return usuarios;
95 }
96
97 //Eliminar un Usuario
98 @DeleteMapping("/{id}")
99 public ResponseEntity<?> delete(@PathVariable("id") Long id){
100     if(usuarioService.findById(id).isPresent()) {
101         return ResponseEntity.notFound().build();
102     }
103     usuarioService.delete(id);
104     return ResponseEntity.ok().build();
105 }
```

Al correr el programa comprobamos que se crea la tabla mediante la conexión en application properties:

```
1 spring.datasource.url=jdbc:mysql://localhost:3306/apirest_full?useSSL=false
2 spring.datasource.username=root
3 spring.datasource.password=
4 spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver
5 spring.jpa.database-platform=org.hibernate.dialect.MySQL8Dialect
6 spring.jpa.show-sql=true
7 spring.jpa.hibernate.ddl-auto=create-drop
8 logging.level.org.hibernate.SQL=debug
9
10 spring.jpa.hibernate.naming-strategy=com.hibernate.naming.DefaultNamingStrategy
```

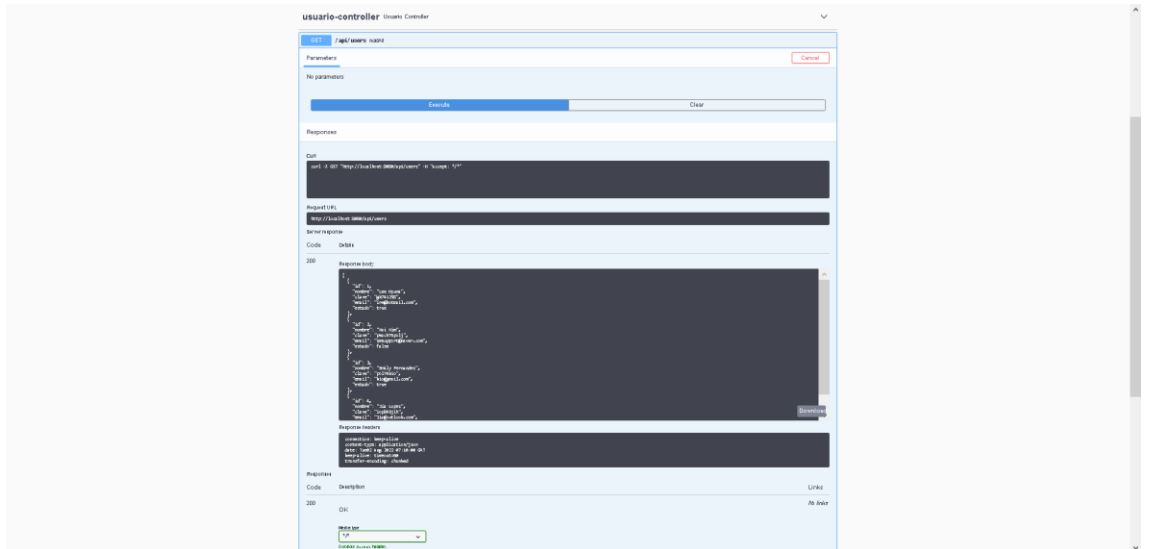
id	clave	mail	estado	nombre
1	g374250	reg@hotmail.com	1	Lee Rhona
2	peach74pnl	ansuaport@naver.com	0	Kia Kim
3	pot74ko	ko@gmail.com	1	Enay Fernandez
4	apirest9	happylol@outlook.com	0	Ja Lopez

Probamos en swagger los diferentes métodos creado en el controlador:

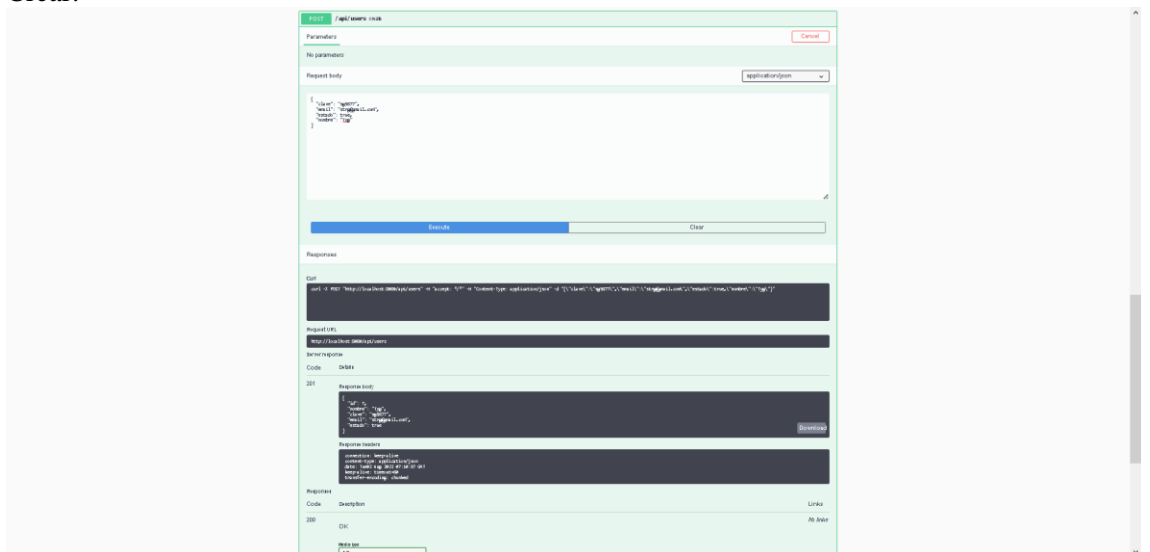


Metodos:

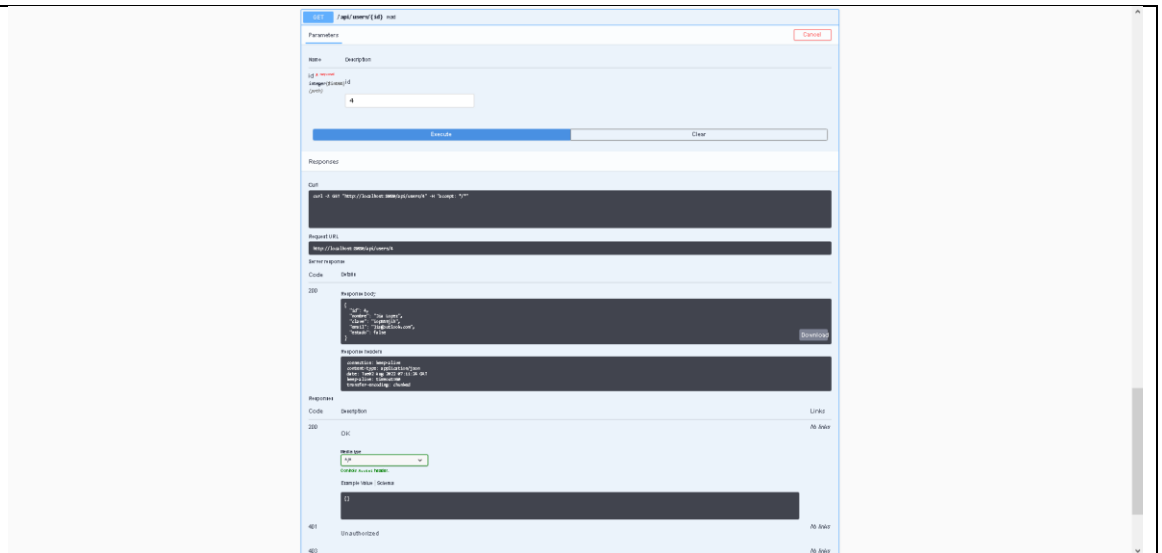
- Listar todos:



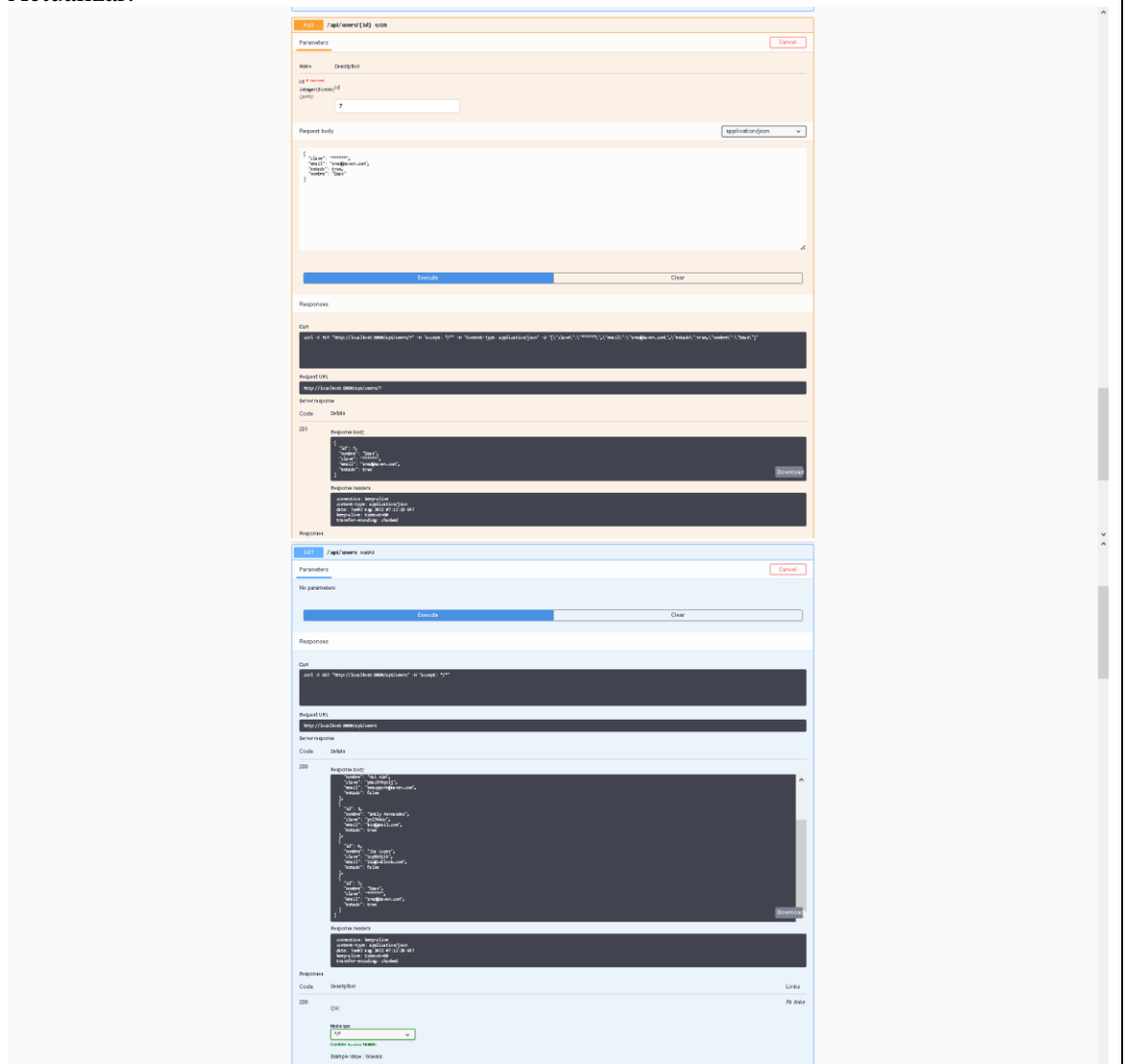
- Crear:



- Listar por id:



- Actualizar:



- Eliminar:

[illegible]

2.5 Conclusiones:


Se creo una API REST-Full que implemente el servicio CRUD con la base de datos MySQL con la ayuda de springboot de la tabla o entidad Usuario.

2.6 Bibliografía:

Descripción en norma APA

Fuentes, F. (2020). *CURSO SPRING BOOT + API REST + JPA + MySQL*. Youtube. <https://www.youtube.com/watch?v=ez6FNBdCUB0&list=PL-A7l3GTDnp1YkBwslsdzuJKF55cISdSD>

3 Firmas de Responsabilidad:

ESTUDIANTE	DOCENTE
	Nombre:
Firma	Firma
Fecha: 02/08/2022	Fecha: