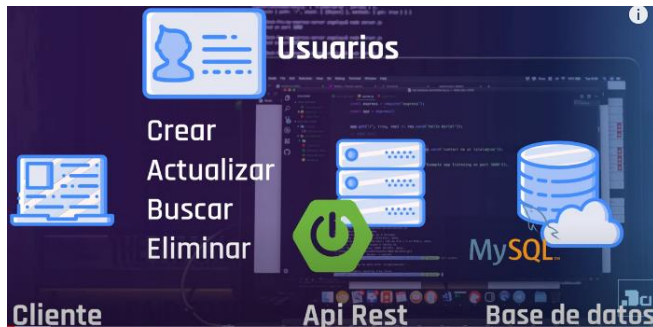


Desarrollar Api Rest con Java Spring Boot

Usamos inicializr:

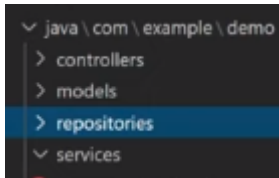
Iniciamos un proyecto con las dependencias MySQL que usa el conectorJ (nos conectamos a la base de datos) y JPA (nos va a ayudar a mapear los datos).

Creamos una carpeta principal que va a contener nuestros paquetes importantes del proyecto.



Estructura de archivos

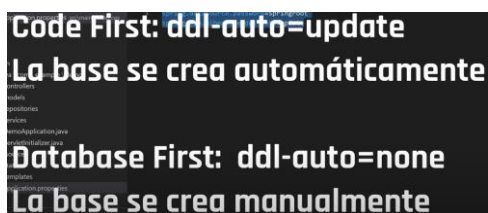
/Controllers	<--	Recibe la petición y ejecuta la lógica principal
UsuarioController.java		
/Services	<--	Acciones permitidas en el sistema
UsuarioService.java		
/Repositories	<--	Consultas Sql o métodos de mapeo de objetos
UsuarioRepository.java		
/Models	<--	Representación de una tabla en una clase Java
UsuarioModel.java		



En el archivo de propiedades escribimos los atributos para conectarnos a la BBDD.

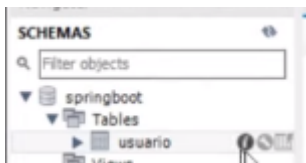
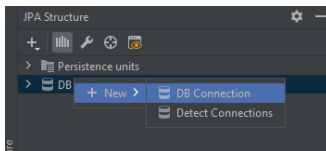
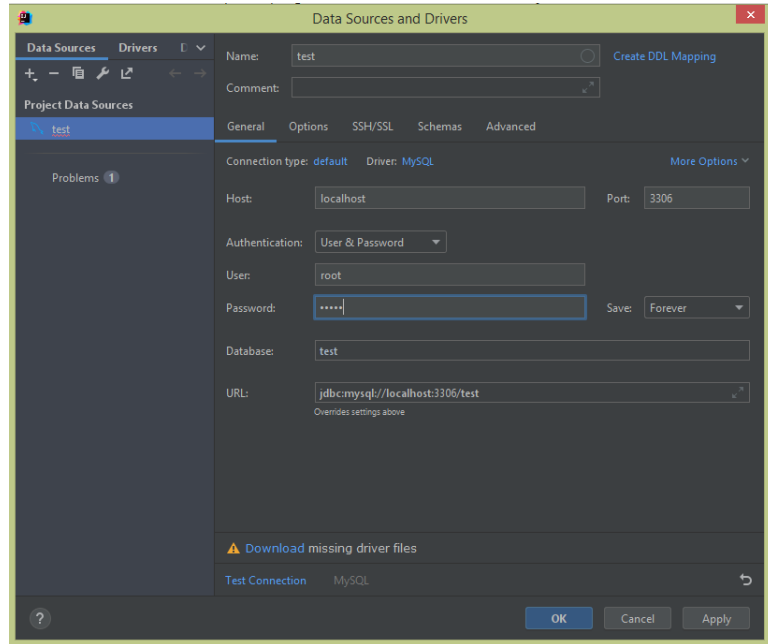
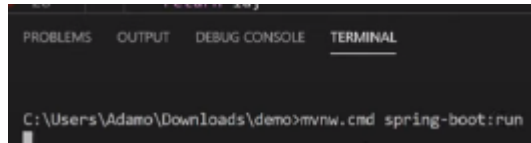


Si la bbdd ya existe, en el atributo ddl-auto colocamos none, si la vamos a crear le ponemos el valor update.

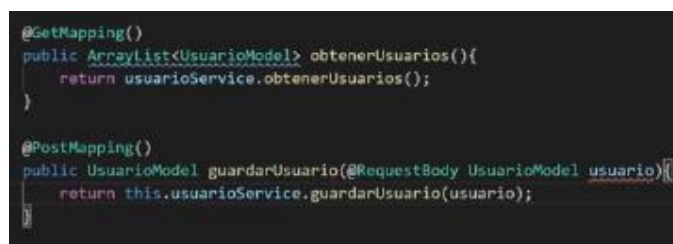
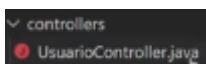


Probar la conexión a la bbdd desde la consola:

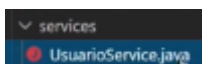
Necesita de la Java EE6.



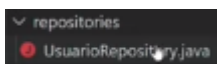
Flujo de trabajo:



se recibe la petición e inmediatamente llama al servicio



: va a ejecutar la lógica de la aplicación llamando a (Get y Post)



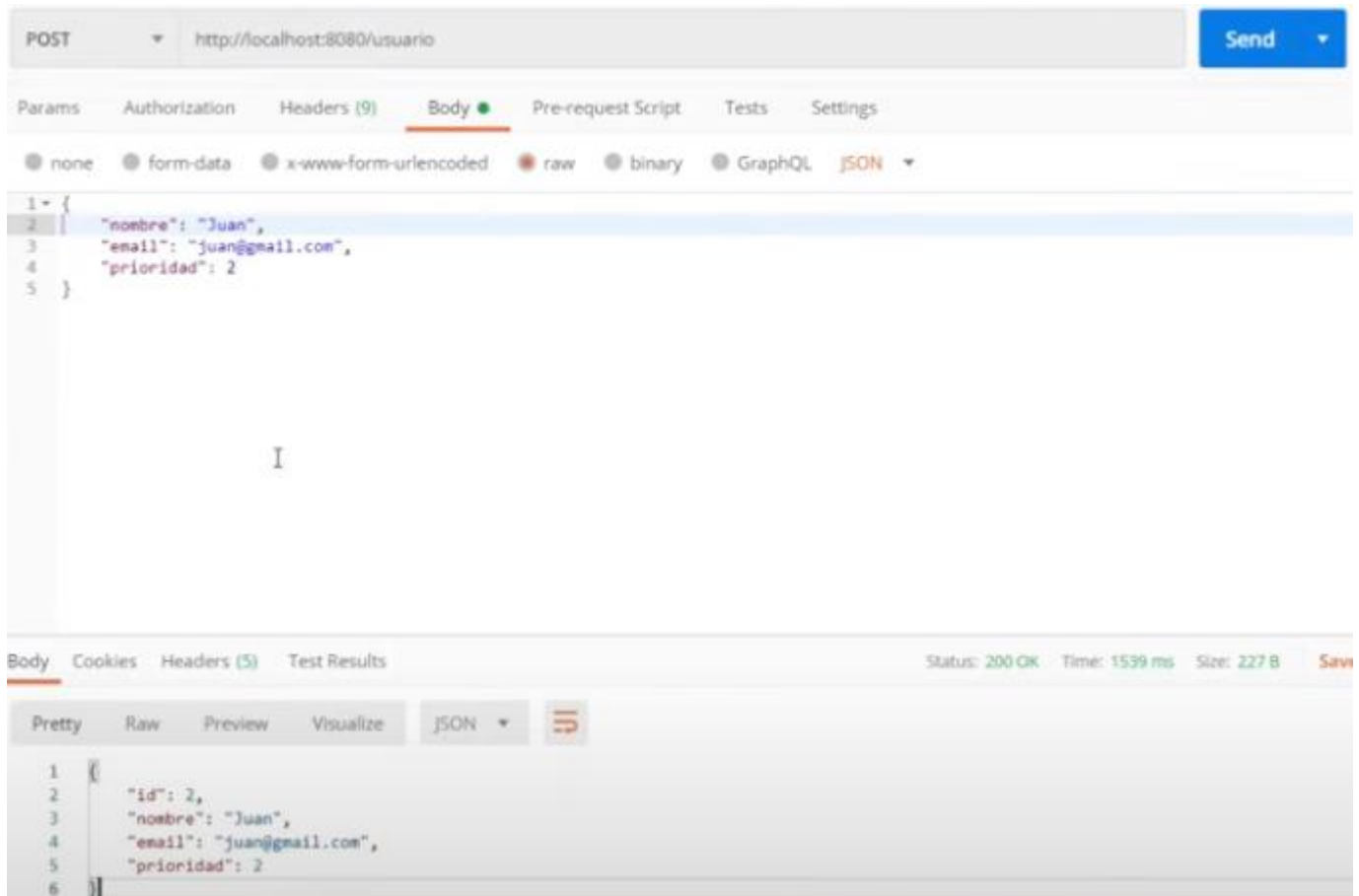
: hace las conexiones reales con la BBDD y utiliza el modelo para saber

models
UsuarioModel

: que tipo de informacion va a traer

CRUD con postman

1-Dar de alta un usuario



2-Modificar un usuario:



3-Consultar por id con servicios:

Ejemplo

```
@Autowired
private UsuarioService usuarioService;

@GetMapping("/{id}")
public ResponseEntity<Usuario> getUsuarioPorId(@PathVariable Long id) {
    return ResponseEntity.ok(usuarioService.findById(id));
}
```

http://localhost:8080/usuario/1

UsuarioController.java:

```
@GetMapping( path =("/{id}")
public Optional<UsuarioModel> obtenerUsuarioPorId(@PathVariable("id") Long id) {
    return this.usuarioService.obtenerPorId(id);
}
```

Consulta pasando por parametro la prioridad y para activarlos ponemos el ?, la variable que queremos pasar y luego el valor:

Ejemplo

```
@PostMapping()
public UsuarioModel guardarUsuario(@RequestBody UsuarioModel usuario){
    return this.usuarioService.guardarUsuario(usuario);
}
```

http://localhost:8080/usuario/query?prioridad=5

```
@GetMapping("/query")
public ArrayList<UsuarioModel> obtenerUsuarioPorPrioridad(@RequestParam("prioridad") Integer prioridad){
    return this.usuarioService.obtenerPorPrioridad(prioridad);
}
```

4- Eliminar un registro:

```
@DeleteMapping( path =("/{id}")
public String eliminarPorId(@PathVariable("id") Long id){
    boolean ok = this.usuarioService.eliminarUsuario(id);
    if (ok){
        return "Se eliminó el usuario con id " + id;
    }else{
        return "No pudo eliminar el usuario con id" + id;
    }
}
```

