**Backend RRHH (Spring Boot + Java 24 + MySQL)**

1. Crear Proyecto Spring Boot.

* Web: https://start.spring.io
* Configuración:
* Project: Maven
* Language: Java
* Spring Boot: 3.5.0 o superior
* Group: gm
* Artifact: rh
* Name: rh
* Description: Sistema de Recursos Humanos
* Package: gnrh
* Packaging: Jar
* Java: 24
* Dependencias:
* Spring Web
* Spring Data JPA
* Lombok
* MySQL Driver

1. Abrir en IntelliJ IDEA

- File > Open > [carpeta descomprimida]

1. Añadir el pom.xml

|  |
| --- |
| **xml**  <properties>  <java.version>24</java.version>  </properties>  <dependencies>  <dependency>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-starter-web</artifactId>  </dependency>  <dependency>  <groupId>org.springframework.boot</groupId>  <artifactId>spring-boot-starter-data-jpa</artifactId>  </dependency>  <dependency>  <groupId>mysql</groupId>  <artifactId>mysql-connector-j</artifactId>  <scope>runtime</scope>  </dependency>  <dependency>  <groupId>org.projectlombok</groupId>  <artifactId>lombok</artifactId>  <optional>true</optional>  </dependency>  </dependencies> |

|  |
| --- |
| Configurar conexión MySQL en `application.properties`java  spring.datasource.url=jdbc:mysql://localhost:3306/recursos\_humanos\_db?createDatabaseIfNotExist=true  spring.datasource.username=root  spring.datasource.password=admin  spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver  spring.jpa.hibernate.ddl-auto=update  spring.jpa.show-sql=true |

1. Crear estructura de paquetes

* gnrh.modelo
* gnrh.repositorio
* gnrh.servicio
* gnrh.controlador

1. Entidad: Empleado.java (paquete modelo)

|  |
| --- |
| java  package gnrh.modelo;  import jakarta.persistence.\*;  import lombok.\*;  @Entity  @Data  @NoArgsConstructor  @AllArgsConstructor  @ToString  public class Empleado {  @Id  @GeneratedValue(strategy = GenerationType.IDENTITY)  private Integer idEmpleado;  private String nombre;  private String departamento;  private Double sueldo;  } |

1. Repositorio: EmpleadoRepositorio.java

|  |
| --- |
| java  package gnrh.repositorio;  import gnrh.modelo.Empleado;  import org.springframework.data.jpa.repository.JpaRepository;  public interface EmpleadoRepositorio extends JpaRepository<Empleado, Integer> {} |

1. Servicio: IEmpleadoServicio.java

|  |
| --- |
| java  package gnrh.servicio;  import gnrh.modelo.Empleado;  import java.util.List;  public interface IEmpleadoServicio {  List<Empleado> listarEmpleados();  Empleado buscarEmpleadoPorId(Integer id);  Empleado guardarEmpleado(Empleado empleado);  void eliminarEmpleado(Empleado empleado);  } |

1. Servicio: EmpleadoServicio.java

|  |
| --- |
| java  package gnrh.servicio;  import gnrh.modelo.Empleado;  import gnrh.repositorio.EmpleadoRepositorio;  import org.springframework.beans.factory.annotation.Autowired;  import org.springframework.stereotype.Service;  import java.util.List;  @Service  public class EmpleadoServicio implements IEmpleadoServicio {  @Autowired  private EmpleadoRepositorio repositorio;  public List<Empleado> listarEmpleados() {  return repositorio.findAll();  }  public Empleado buscarEmpleadoPorId(Integer id) {  return repositorio.findById(id).orElse(null);  }  public Empleado guardarEmpleado(Empleado empleado) {  return repositorio.save(empleado);  }  public void eliminarEmpleado(Empleado empleado) {  repositorio.delete(empleado);  }  } |

1. Controlador: EmpleadoControlador.java

|  |
| --- |
| java  package gnrh.controlador;  import gnrh.modelo.Empleado;  import gnrh.servicio.IEmpleadoServicio;  import org.springframework.beans.factory.annotation.Autowired;  import org.springframework.web.bind.annotation.\*;  import java.util.List;  @RestController  @RequestMapping("/rh-app")  @CrossOrigin(value = "http://localhost:3000")  public class EmpleadoControlador {  @Autowired  private IEmpleadoServicio servicio;  @GetMapping("/empleados")  public List<Empleado> obtenerEmpleados() {  return servicio.listarEmpleados();  }  @PostMapping("/empleados")  public Empleado crearEmpleado(@RequestBody Empleado e) {  return servicio.guardarEmpleado(e);  }  @PutMapping("/empleados")  public Empleado actualizarEmpleado(@RequestBody Empleado e) {  return servicio.guardarEmpleado(e);  }  @DeleteMapping("/empleados")  public void eliminarEmpleado(@RequestBody Empleado e) {  servicio.eliminarEmpleado(e);  }  } |

1. Verificación:

* Ejecutar RhApplication.java
* Usar Postman con:
* GET: http://localhost:8080/rh-app/empleados
* POST/PUT/DELETE con cuerpo JSON.

1. Insertar Datos Prueba o Definitivos.

|  |
| --- |
| Sql  INSERT INTO empleado (nombre, departamento, sueldo) VALUES  ('Juan Pérez', 'Sistemas', 20000),  ('Carla Jiménez', 'Finanzas', 23000); |