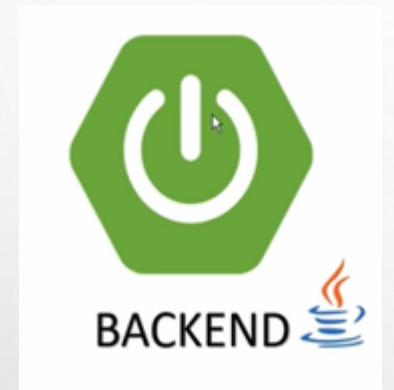


SPRING BOOT MAVEN

EJEMPLO PRACTICO



ECLIPSE
maven – spring - java

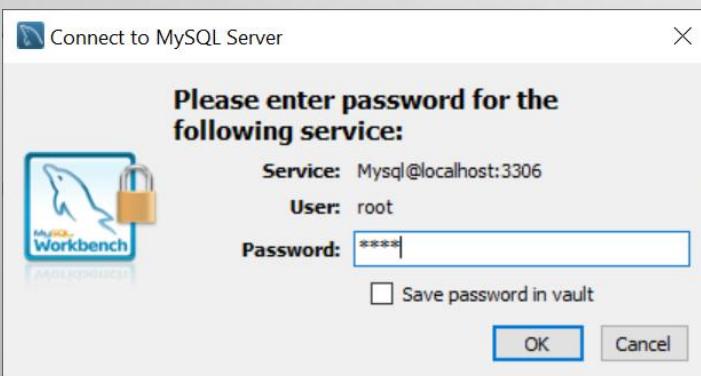
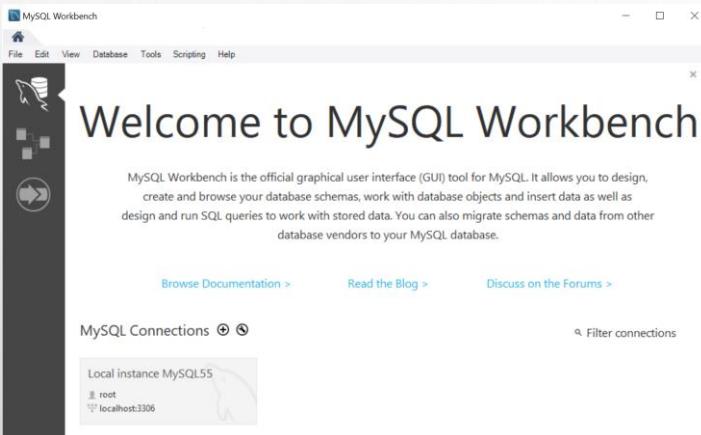
**Tener todos
estos productos
instalados**

ARQUITECTURA

- CREAR LA **BD**
 - MYSQL
- CREAR PROYECTO **BACK END**
 - ECLIPSE CON SPRING Y JAVA

BASE DE DATOS

Login



Crear nuevo Schema → ejemplo
Crear nueva Table → personas
Campos **id** (autoincrement, pk, int)
name (varchar)
apellidos (varchar)

Agregar 3 registros a mano

MySQL Workbench

Local instance MySQL55

File Edit View Query Database Server Tools Scripting Help

Navigator: Schemas

Filter objects

bookstore
contactmanager
demo
ejemplo

Tables

personas

Columns: id, name, apellidos

Indexes
Foreign Keys
Triggers
Views
Stored Procedures
Functions

ejemplotriggers
hibernateejemplos
springdemo
test

Management Schemas

Information Object Info Session

Query Completed

Result Grid

SQLAdditions

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

1 ① SELECT * FROM ejemplo.personas;

	id	name	apellidos
1	Juan	Perez	
2	Usuario2	Morales	
4	Ale	Buduete	
5	Daniel	Parisi	
6	Pedro	Roman	
NULL	NULL	NULL	NULL

personas 1

Action Output

#	Time	Action	Message	Duration / Fetch
1	08:28:20	SELECT * FROM ejemplo.personas LIMI...	5 row(s) returned	0.000 sec / 0.000 sec

BACKEND

- Crear estructura vacia usando Sprint Initializr
 - <http://start.spring.io>
 - Elegir
 - Agregar las dependencias: WEB, JPA, MYSQL
 - Cambiar el nombre en Artifact, a ejemplo01 que será el nombre de nuestro proyecto
 - Presionar GENERATE y obtendremos ejemplo01.zip

The screenshot shows the Spring Initializr web application at <https://start.spring.io>. A red box highlights the configuration section on the left, which includes:

- Project:** Maven Project (selected)
- Language:** Java (selected)
- Spring Boot:** 2.4.5 (selected)
- Dependencies:** Spring Web (WEB), MySQL Driver (SQL), Spring Data JPA (SQL)
- Project Metadata:** Group: com.example, Artifact: ejemplo01, Name: ejemplo01

At the bottom right are buttons for **GENERATE**, **EXPLORE**, and **SHARE...**.

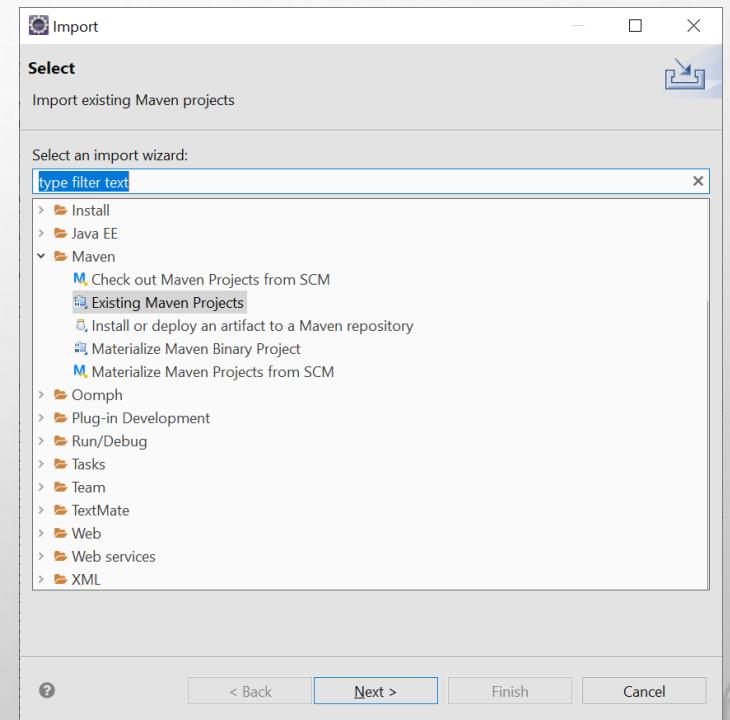
This is a screenshot of the Spring Initializr configuration page, showing the same settings as the previous screenshot:

- Project:** Maven Project (selected)
- Language:** Java (selected)
- Spring Boot:** 2.4.5 (selected)
- Dependencies:** Spring Web (WEB), MySQL Driver (SQL), Spring Data JPA (SQL)
- Project Metadata:** Group: com.example, Artifact: ejemplo01, Name: ejemplo01

BACKEND

En eclipse

- Extraer el archivo compactado para poder importarlo dentro del WORKSPACE
- **IMPORTAR EL PROYECTO → MAVEN**
- **Seleccionar que SI, a la hora de bajar e instalar todas las dependencias**



BACKEND

En eclipse

- Modificar el archivo **application.properties** con los datos de la base de datos

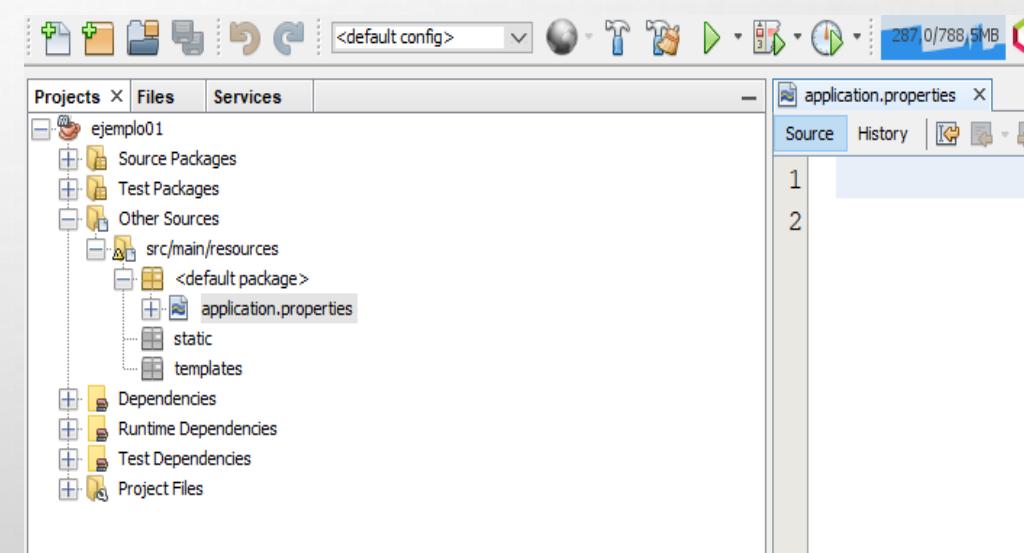
SERVER.CONTEXTPATH=/EJEMPLO01

SPRING.DATASOURCE.URL=JDBC:MySQL://localhost:3306/EJEMPLO

SPRING.DATASOURCE.USERNAME=ROOT

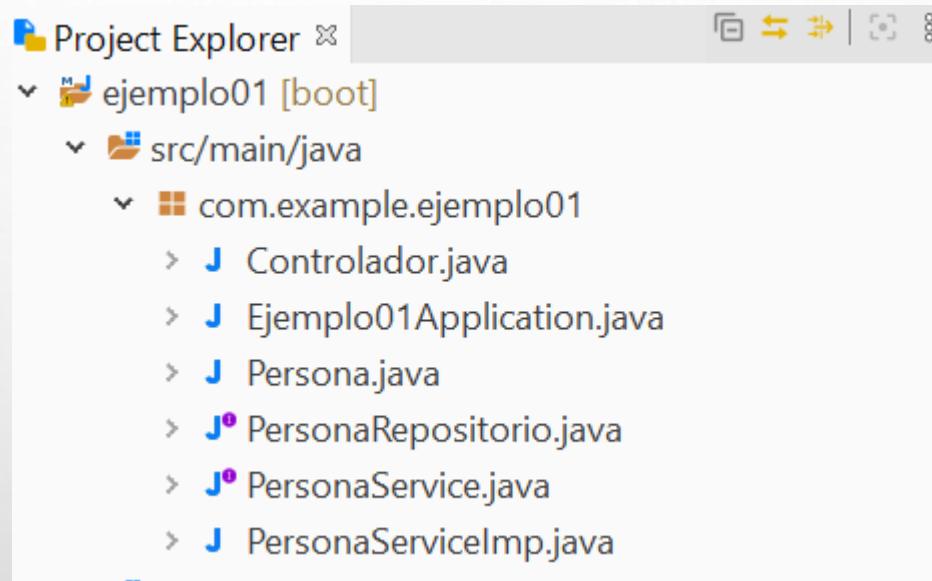
SPRING.DATASOURCE.PASSWORD=1234

SPRING.DATASOURCE.DRIVER-CLASS-NAME=COM.MYSQL.CJ.JDBC.DRIVER



BACKEND

Copiar estos archivos



Presupone conocimiento previo de JAVA + SPRING
No lo explico en este curso

BACKEND

- **persona.java**
 - es un ENTITY BEAN
 - @ ANNOTATION para establecer la estructura de la table
 - Getters/setters
- Interfaces
 - Dialogar con la BD → **PersonaRepository.java**
 - Dialogar con el FrontEnd → **PersonaService.java**
- **personaServicelmp.java**
 - La componenete que hace de intermediaria entre el servicio al front end con los datos del backend
- **Controlador.java**
 - @CROSSORIGIN(ORIGINS = "HTTP://LOCALHOST:4200")
 - @GETMAPPING
 - @POSTMAPPING
 - @REQUESTMAPPING({"/PERSONAS"}) → "HTTP://LOCALHOST:8080/PERSONAS"

BACKEND

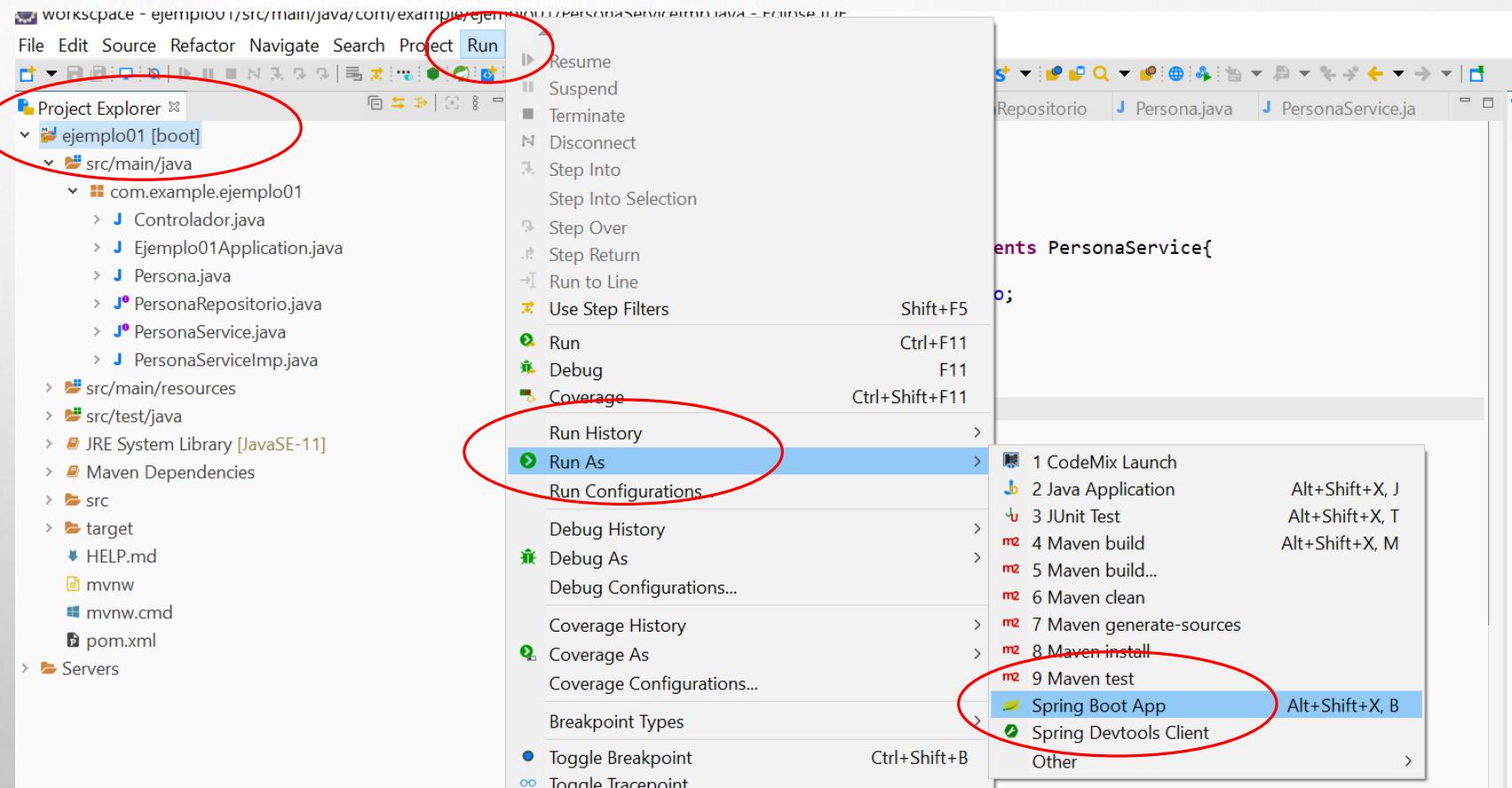
FINALIZAMOS DEJANDO EN EJECUCION EL PROYECTO ...

PARA ESO HACER

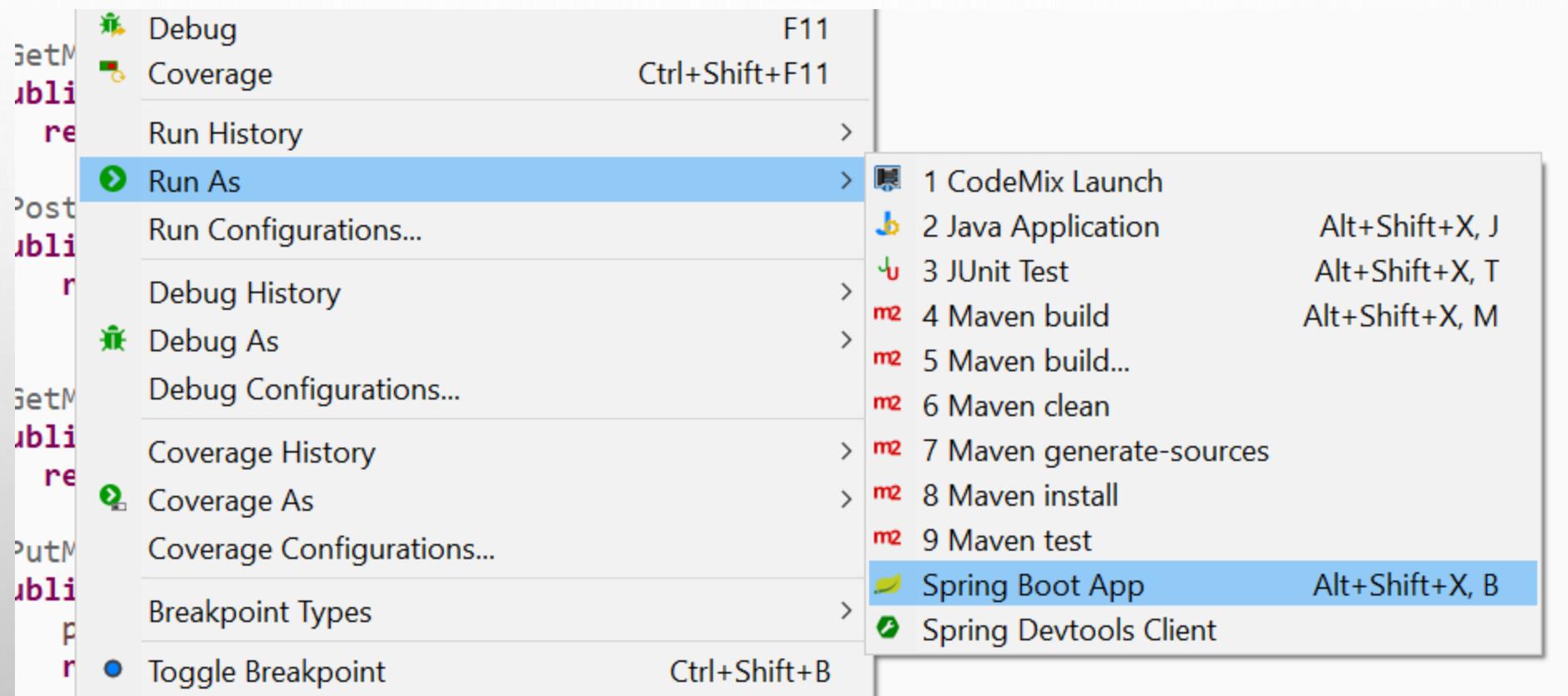
RUN SPRING BOOT APP

VERIFICAR EN EL BROWSER →

http://localhost:8080/personas



EJECUTAMOS COLOCAR EL PUNTERO DEL MOUSE EN EL RAIZ DEL PROYECTO EJEMPLO01



Project Explorer PersonaServiceImp.java Controlador.java PersonaRepository.java Persona.java PersonaService.java Outline Task List

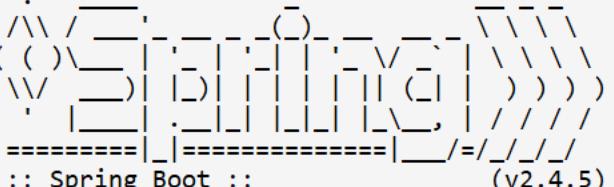
```

J PersonaServiceImp.java   J Controlador.java   J PersonaRepository.java   J Persona.java   J PersonaService.java   □ □
1 package com.example.ejemplo01;
2
3o import java.util.List;□

```

Markers Properties Servers Data Source Explorer Snippets Console Progress

ejemplo01 - Ejemplo01Application [Spring Boot App] C:\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_15.0.2.v20210201-0955\jre\bin\javaw.exe (3 may. 2021 08:49:02)



=====|_=====: Spring Boot :: (v2.4.5)

```

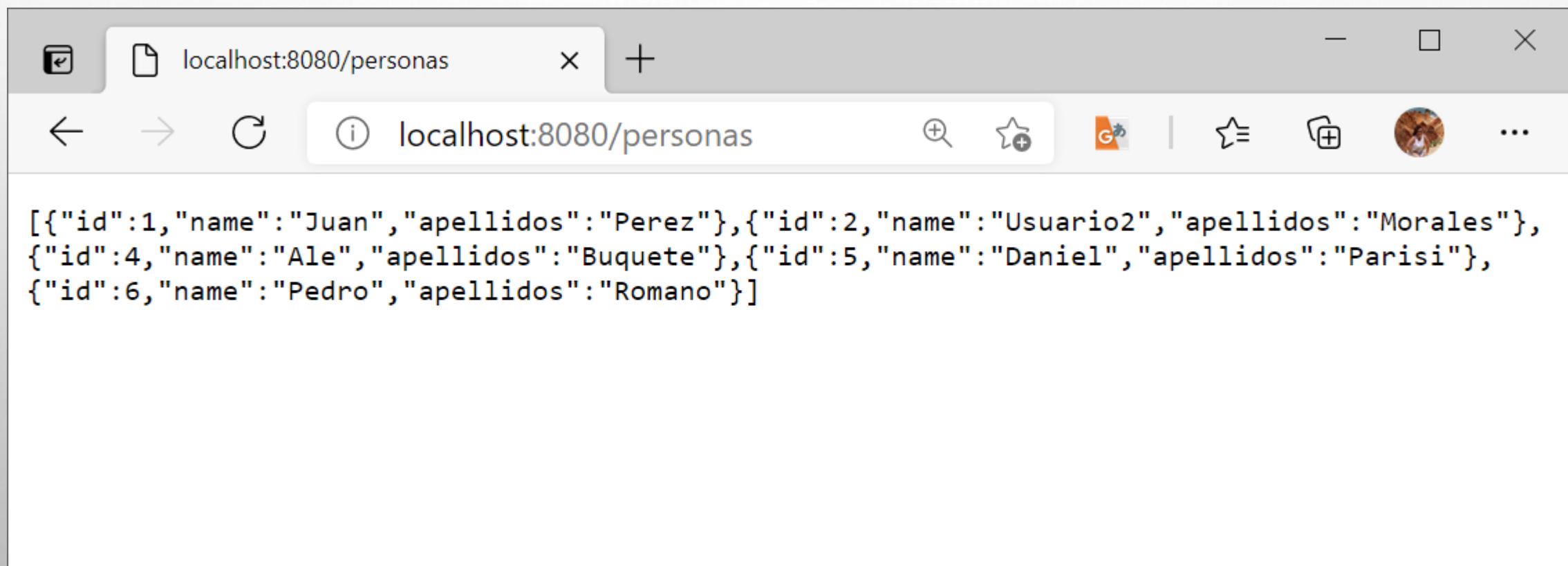
2021-05-03 08:49:07.635 INFO 15304 --- [
2021-05-03 08:49:07.637 INFO 15304 --- [
2021-05-03 08:49:08.250 INFO 15304 --- [
2021-05-03 08:49:08.290 INFO 15304 --- [
2021-05-03 08:49:08.740 INFO 15304 --- [
2021-05-03 08:49:08.750 INFO 15304 --- [
2021-05-03 08:49:08.750 INFO 15304 --- [
2021-05-03 08:49:08.852 INFO 15304 --- [
2021-05-03 08:49:08.852 INFO 15304 --- [
2021-05-03 08:49:09.023 INFO 15304 --- [
2021-05-03 08:49:09.189 INFO 15304 --- [
2021-05-03 08:49:09.234 INFO 15304 --- [
2021-05-03 08:49:09.290 INFO 15304 --- [
2021-05-03 08:49:09.466 INFO 15304 --- [
2021-05-03 08:49:09.603 INFO 15304 --- [
2021-05-03 08:49:10.129 INFO 15304 --- [
2021-05-03 08:49:10.138 INFO 15304 --- [
2021-05-03 08:49:10.433 WARN 15304 --- [
2021-05-03 08:49:10.589 INFO 15304 --- [
2021-05-03 08:49:10.933 INFO 15304 --- [
2021-05-03 08:49:10.950 INFO 15304 --- [
main] c.e.ejemplo01.Ejemplo01Application : Starting Ejemplo01Application using Java
main] c.e.ejemplo01.Ejemplo01Application : No active profile set, falling back to de
main] .s.d.r.c.RepositoryConfigurationDelegate : Bootstrapping Spring Data JPA repository
main] .s.d.r.c.RepositoryConfigurationDelegate : Finished Spring Data repository scanning
main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat initialized with port(s): 8080 (ht
main] o.apache.catalina.core.StandardService : Starting service [Tomcat]
main] org.apache.catalina.core.StandardEngine : Starting Servlet engine: [Apache Tomcat/9
main] o.a.c.c.C.[Tomcat].[localhost].[/] : Initializing Spring embedded WebApplication
main] w.s.c.ServletWebServerApplicationContext : Root WebApplicationContext: initializatio
main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Starting...
main] com.zaxxer.hikari.HikariDataSource : HikariPool-1 - Start completed.
main] o.hibernate.jpa.internal.util.LogHelper : HHH000204: Processing PersistenceUnitInfo
main] org.hibernate.Version : HHH000412: Hibernate ORM core version 5.4
main] o.hibernate.annotations.common.Version : HCANN000001: Hibernate Commons Annotation
main] org.hibernate.dialect.Dialect : HHH000400: Using dialect: org.hibernate.dialect
main] o.h.e.t.j.p.i.JtaPlatformInitiator : HHH000490: Using JtaPlatform implementatio
main] j.LocalContainerEntityManagerFactoryBean : Initialized JPA EntityManagerFactory for
main] JpaBaseConfiguration$JpaWebConfiguration : spring.jpa.open-in-view is enabled by de
main] o.s.s.concurrent.ThreadPoolTaskExecutor : Initializing ExecutorService 'application
main] o.s.b.w.embedded.tomcat.TomcatWebServer : Tomcat started on port(s): 8080 (http) wi
main] c.e.ejemplo01.Ejemplo01Application : Started Ejemplo01Application in 3.684 sec

```

CodeMix ejemplo01

BACKEND

COMPROBACION DE FUNCIONAMIENTO



A screenshot of a web browser window. The address bar shows the URL `localhost:8080/personas`. The page content displays a JSON array of six person objects:

```
[{"id":1,"name":"Juan","apellidos":"Perez"}, {"id":2,"name":"Usuario2","apellidos":"Morales"}, {"id":4,"name":"Ale","apellidos":"Buquete"}, {"id":5,"name":"Daniel","apellidos":"Parisi"}, {"id":6,"name":"Pedro","apellidos":"Romano"}]
```