



# Módulo 3: Back-end

## Sesión 1: API-REST

**Equipo de desarrolladores:**

Cristian David Ríos MSc  
Daniel Escobar Grisales MSc  
Nestor Rafael Calvo MSc

**Coordinador del proyecto:**

Prof. Dr.-Ing. Juan Rafael Orozco Arroyave





# Hola!

## Mi nombre es Daniel Escobar

You can find me at:

 @dangrisales



# Sesión 1: Introducción a API-REST

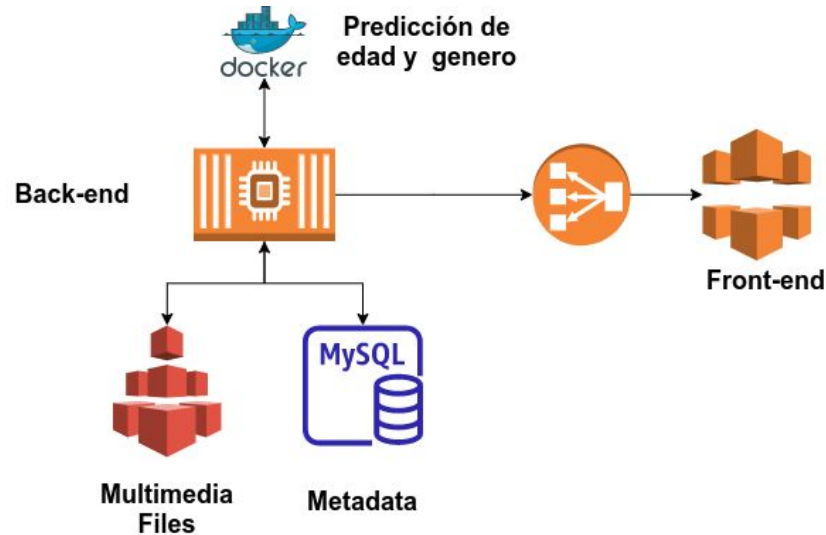
## Temas

- Introducción
- Que es un API-REST
  - Ventajas de un API-REST
  - Características de un API-REST
- Spring-Boot
  - Arquitectura
  - Crear un proyecto usando
  - Spring Initializr
- Recursos y conocimientos básicos

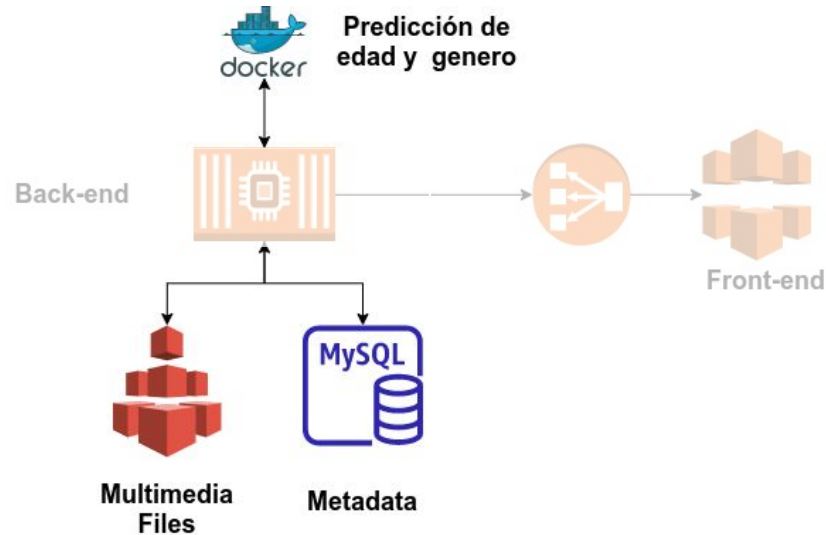
A decorative network diagram in the top-left corner, featuring a complex web of interconnected nodes and lines. The nodes are represented by small circles, some of which are larger and have concentric circles, suggesting different levels of connectivity or importance. The lines are thin and gray, creating a mesh-like structure.

# 2. **Introducción**

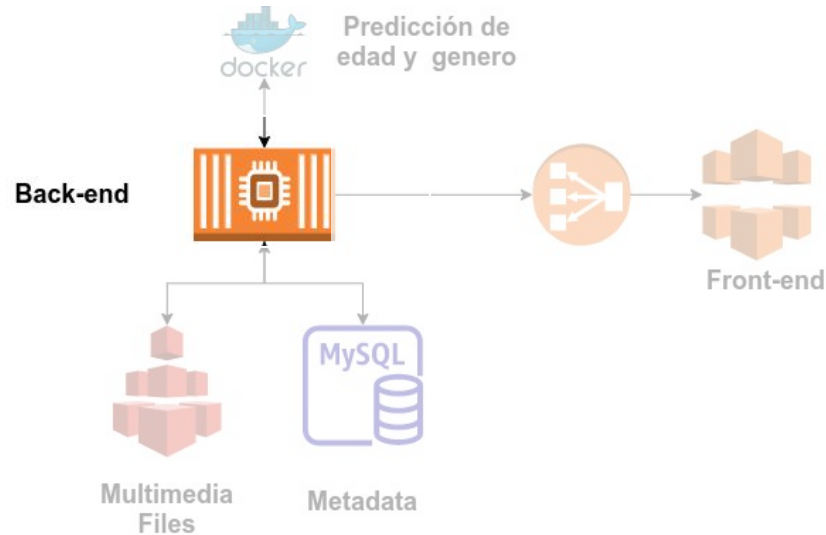
# Arquitectura y características



# Arquitectura y características



# Arquitectura y características



A decorative network diagram in the top-left corner, featuring a complex web of interconnected nodes and lines, with some nodes highlighted in blue.

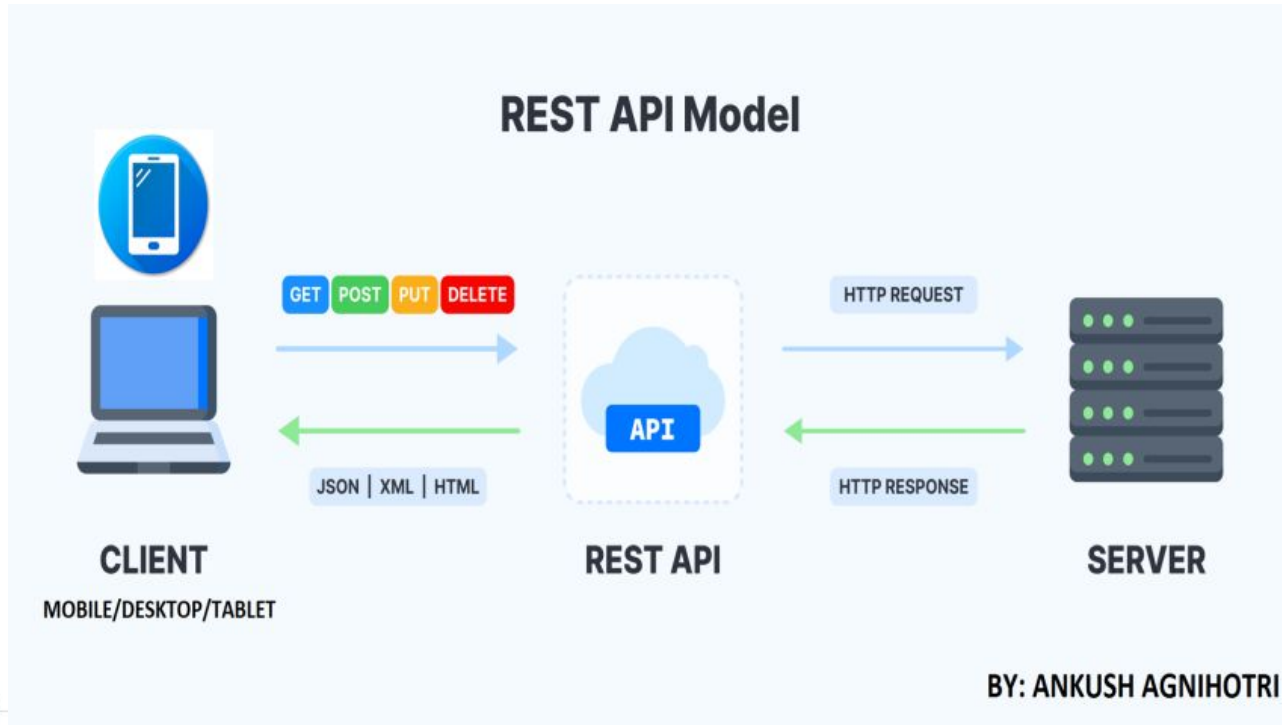
# 2.

# API-REST

REST: REpresentational State Transfer



# API-REST



## Ventajas de API-REST

- ⦿ Fácil integración con la interfaz gráfica de usuario
- ⦿ Escalabilidad
- ⦿ Portabilidad y compatibilidad
- ⦿ Independencia de lenguajes entre el back-end y el front-end

## Características de API-REST

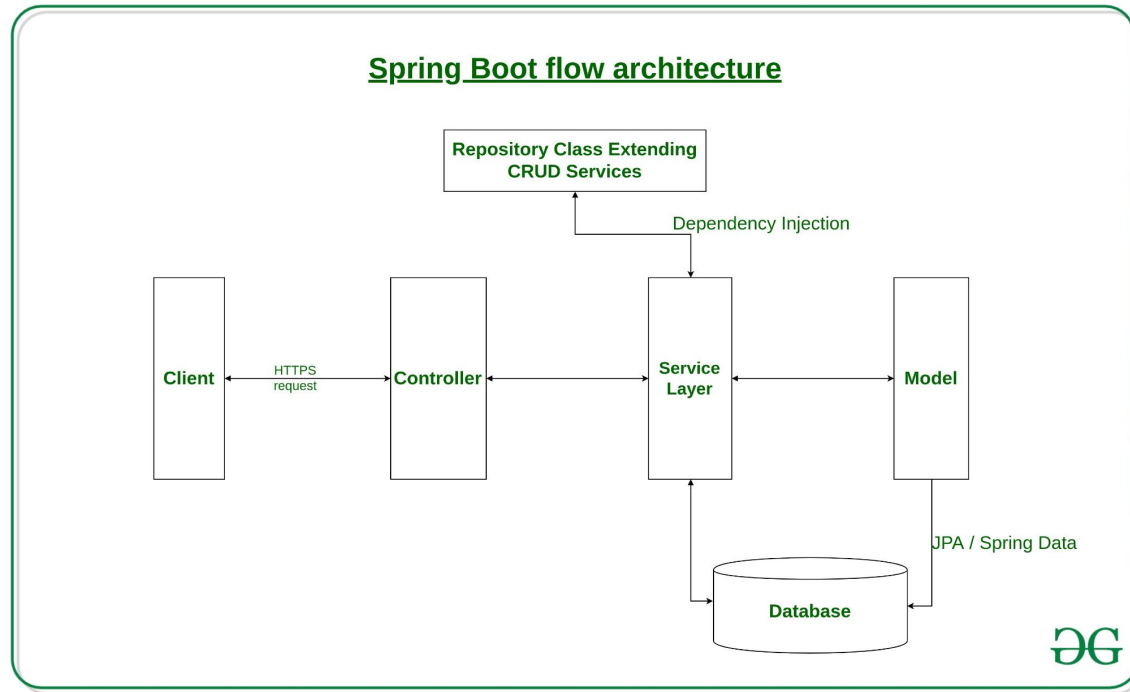
- ⦿ Recursos accesibles por identificadores (URLs)
- ⦿ Se comunica mediante protocolo HTTP
- ⦿ Una misma URL modifica la operación mediante los “verbos”
  - GET: Para operaciones de consulta de datos
  - POST: Operaciones de inserción
  - PUT: Para realizar modificaciones
  - DELETE: Para eliminar un recurso dado

3.

# Spring Boot

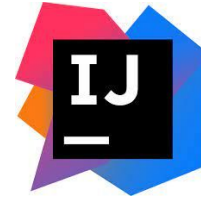
Java Framework open-source para crear  
APIs REST

# Spring Boot



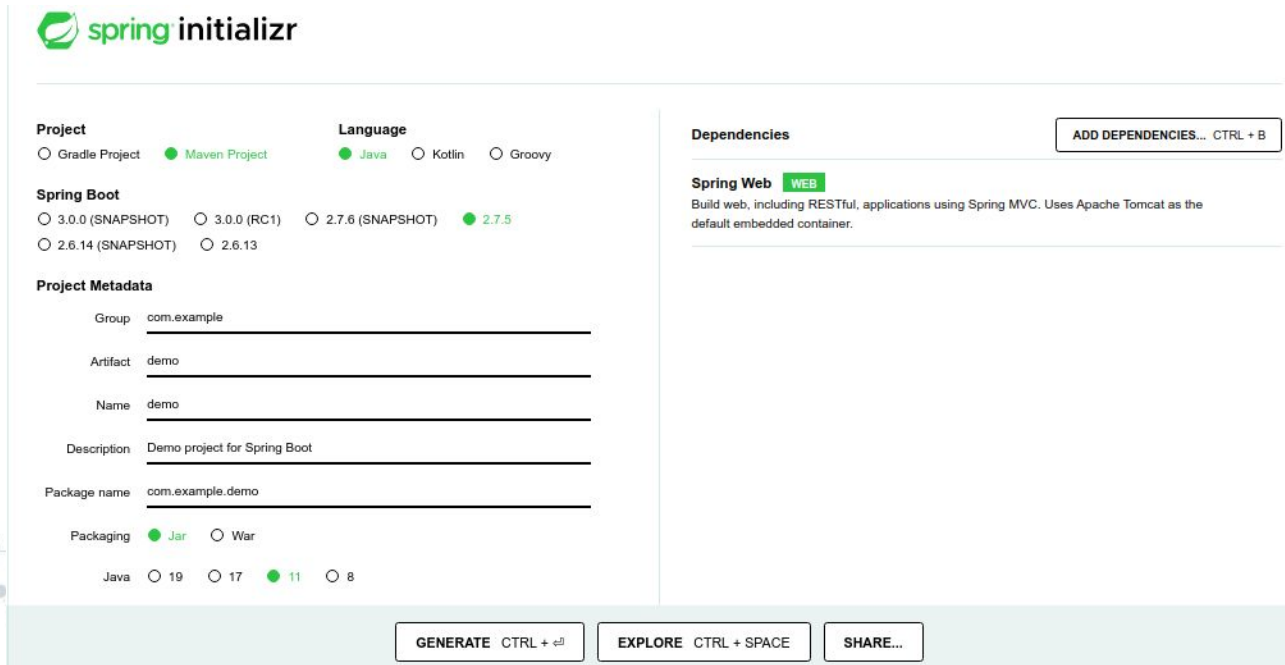
# Requisitos básicos

- ⦿ Entorno de Desarrollo Integrado (IDE)
  - IntelliJ IDEA (recomendado)
  - Visual Studio Code
- ⦿ Lenguajes de programación
  - Java (recomendado)
  - Kotlin
- ⦿ Programación orientada a objetos.
- ⦿ Java Development Kit (JDK)
  - Version 11.0.16 (recomendado)



# Crear un proyecto Spring Boot

Link: [start.spring.io](https://start.spring.io)



The image shows the Spring Initializr web form for creating a new project. The form is divided into several sections: Project, Language, Spring Boot, Project Metadata, Dependencies, and Packaging. The 'Project' section has radio buttons for Gradle Project, Maven Project (selected), and others. The 'Language' section has radio buttons for Java (selected), Kotlin, and Groovy. The 'Spring Boot' section has radio buttons for various versions, with 2.7.5 (selected) highlighted in green. The 'Project Metadata' section contains text input fields for Group, Artifact, Name, Description, and Package name. The 'Dependencies' section has a button to add dependencies. The 'Packaging' section has radio buttons for Jar (selected) and War. At the bottom, there are three buttons: GENERATE, EXPLORE, and SHARE...

**spring initializr**

**Project**  
☐ Gradle Project ☒ Maven Project

**Language**  
☒ Java ☐ Kotlin ☐ Groovy

**Spring Boot**  
☐ 3.0.0 (SNAPSHOT) ☐ 3.0.0 (RC1) ☐ 2.7.6 (SNAPSHOT) ☒ 2.7.5  
☐ 2.6.14 (SNAPSHOT) ☐ 2.6.13

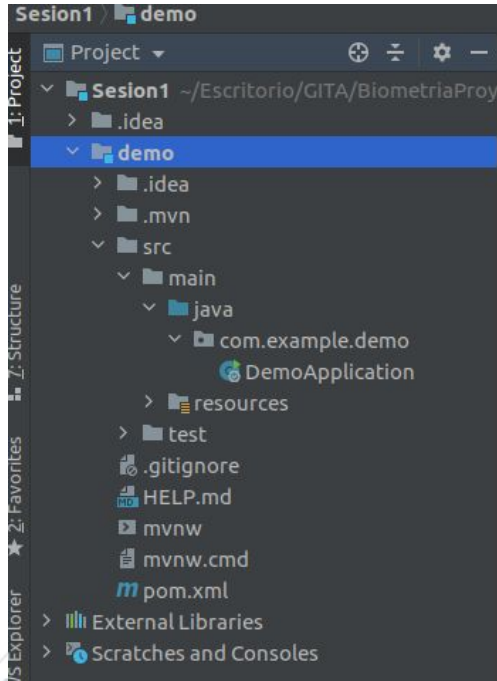
**Project Metadata**  
Group   
Artifact   
Name   
Description   
Package name   
Packaging ☒ Jar ☐ War  
Java ☐ 19 ☐ 17 ☒ 11 ☐ 8

**Dependencies** ADD DEPENDENCIES... CTRL + B

**Spring Web** WEB  
Build web, including RESTful, applications using Spring MVC. Uses Apache Tomcat as the default embedded container.

**GENERATE** CTRL + G **EXPLORE** CTRL + SPACE **SHARE...**

# Crear un proyecto Spring Boot



```
1 package com.example.demo;  
2  
3 import ...  
4  
5  
6 @SpringBootApplication  
7 public class DemoApplication {  
8  
9     public static void main(String[] args) {  
10  
11         SpringApplication.run(DemoApplication.class, args);  
12     }  
13  
14 }  
15
```

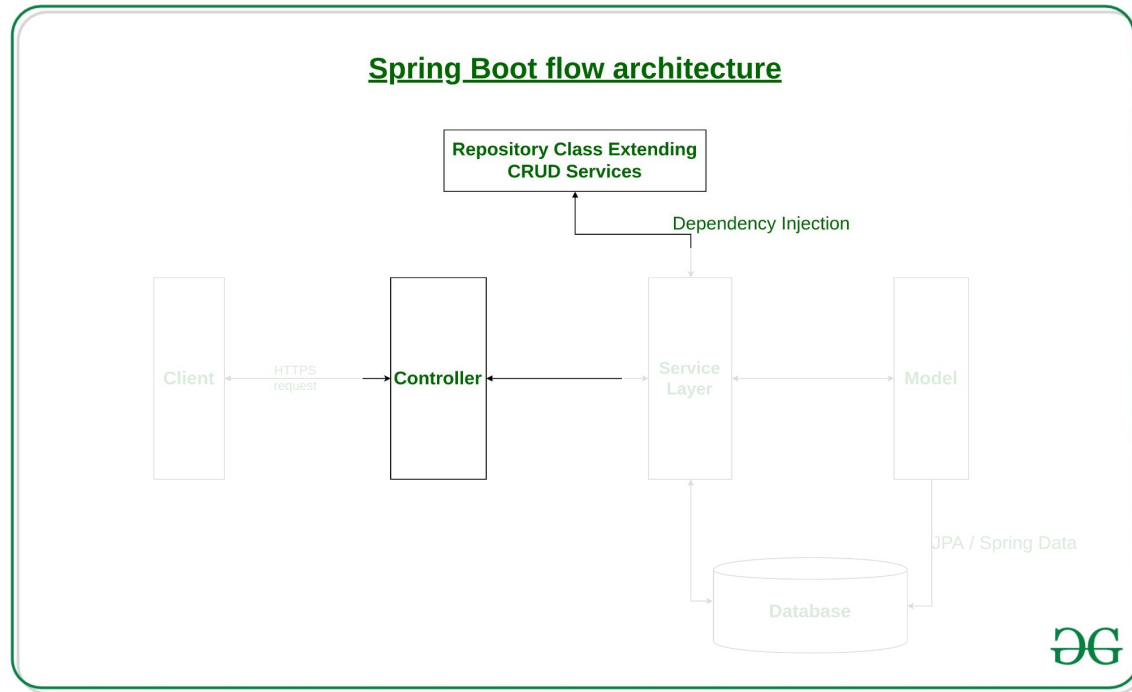


A decorative network diagram in the top-left corner, featuring a complex web of interconnected nodes and lines, with some nodes highlighted in blue.

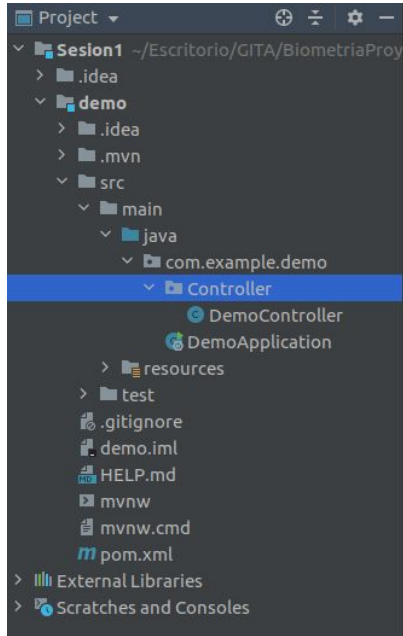
4.

# Creación de recursos

# Spring Boot



# Spring Boot



```
1 package com.example.demo.Controller;
2
3 import org.springframework.web.bind.annotation.PostMapping;
4 import org.springframework.web.bind.annotation.RequestMapping;
5 import org.springframework.web.bind.annotation.RestController;
6
7 @RestController
8 @RequestMapping(value = "/")
9 public class DemoController {
10
11     @PostMapping(value = "/hola")
12     public String helloGFG()
13     {
14         return "Hola Mundo";
15     }
16 }
17
```

A decorative graphic in the top-left corner consisting of a network of interconnected nodes and lines, resembling a molecular structure or a web. The nodes are represented by small circles, some solid and some hollow, connected by thin lines.

3.

# **Manos a la obra**

Creación del primer recurso



# Gracias!

## ¿Alguna pregunta o comentario?

Puedes escribirme a:

[daniel.esobar@udea.edu.co](mailto:daniel.esobar@udea.edu.co)