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

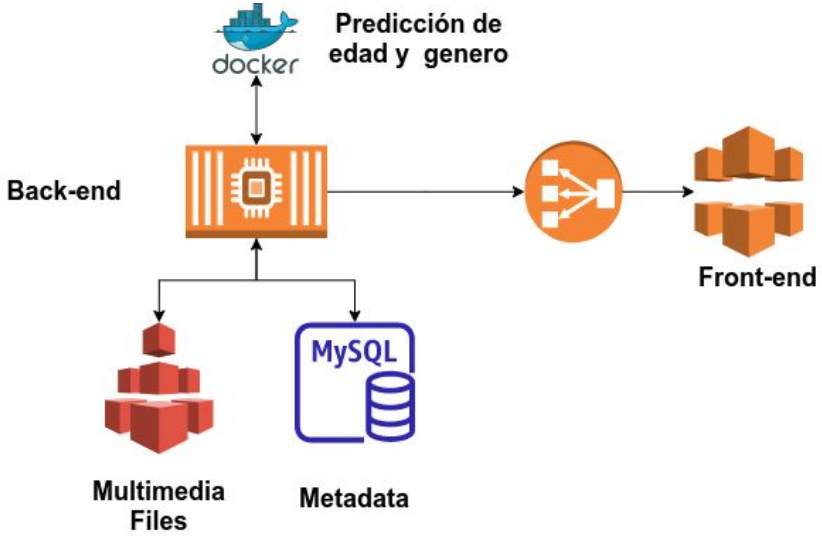
Sesión 1: Introducción a API-REST

Temas

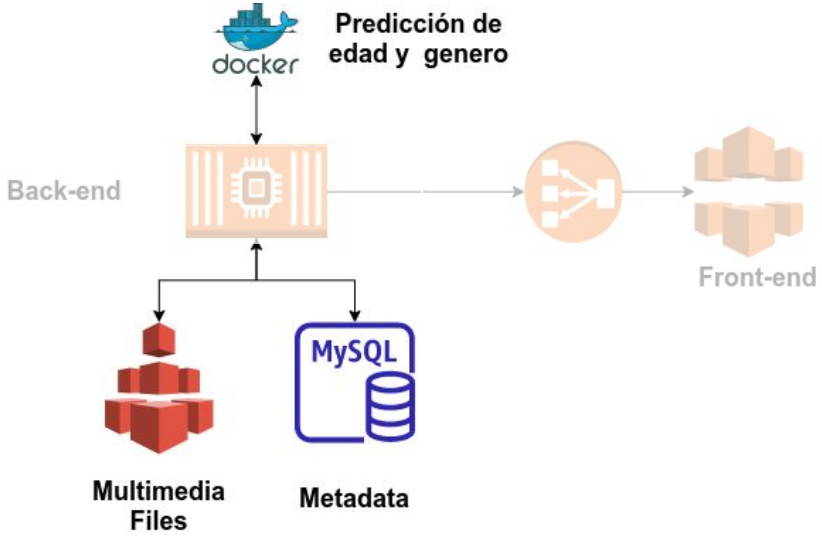
- Introducción
- Que es un API-REST
  - Ventajas de un API-REST
  - Características de un API-REST
- Spring-Boot
  - Arquitectura
  - Crear un proyecto usando
  - Spring Initializr
- Recursos y conocimientos básicos

2.
Introducción

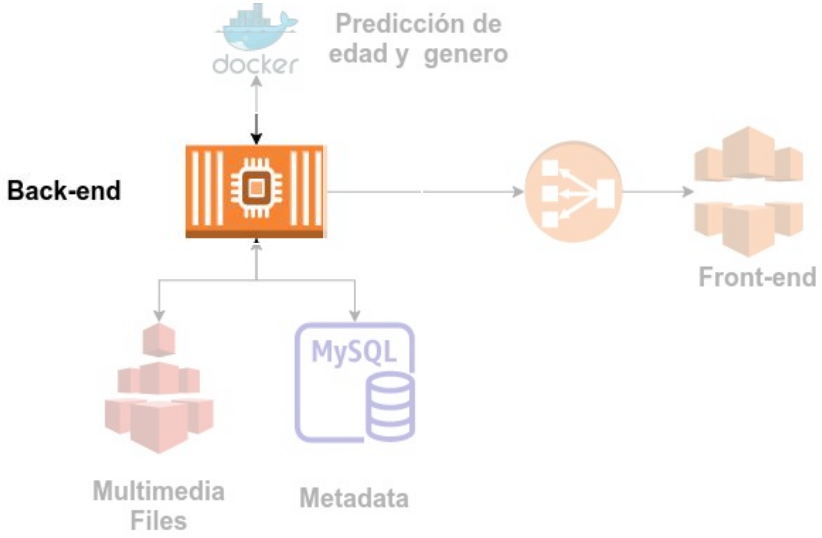
Arquitectura y características



Arquitectura y características



Arquitectura y características

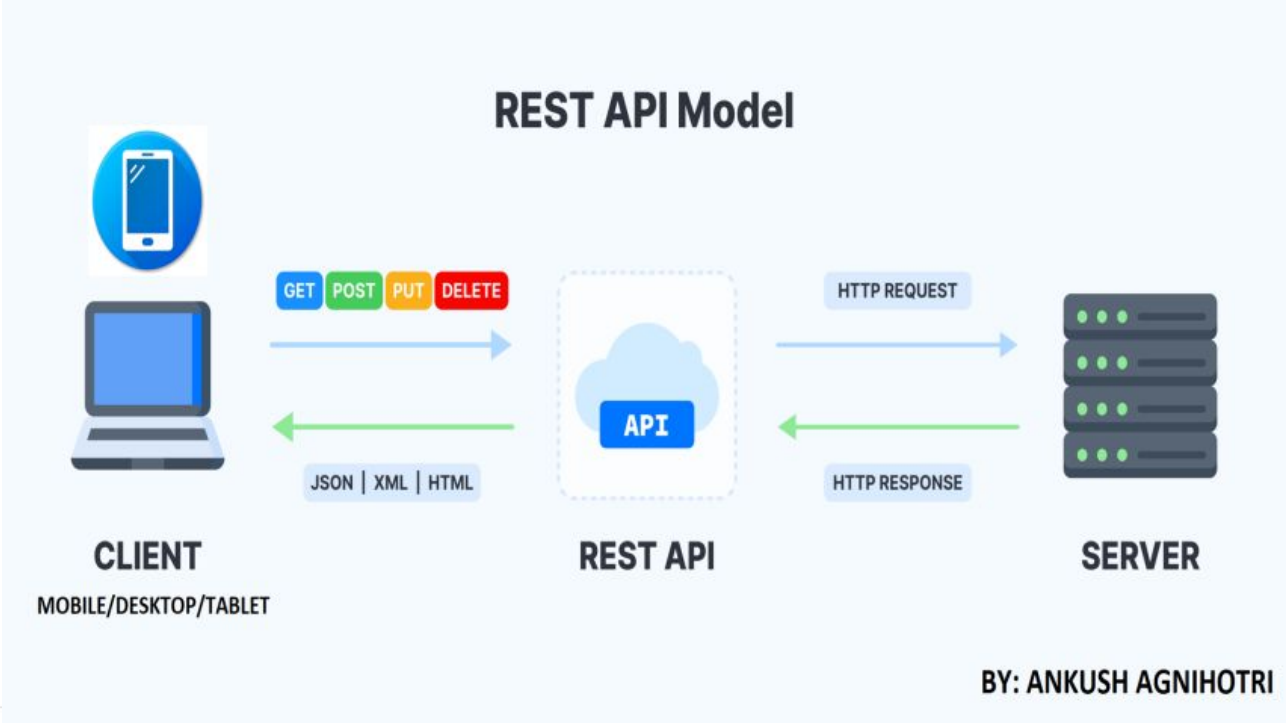


2.

API-REST

REST: REpresentational State Transfer

API-REST


REST API Model
GET
POST
PUT
DELETE
JSON | XML | HTML
HTTP REQUEST
HTTP RESPONSE
API
CLIENT
MOBILE/DESKTOP/TABLET
REST API
SERVER
BY: ANKUSH AGNIHOTRI

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

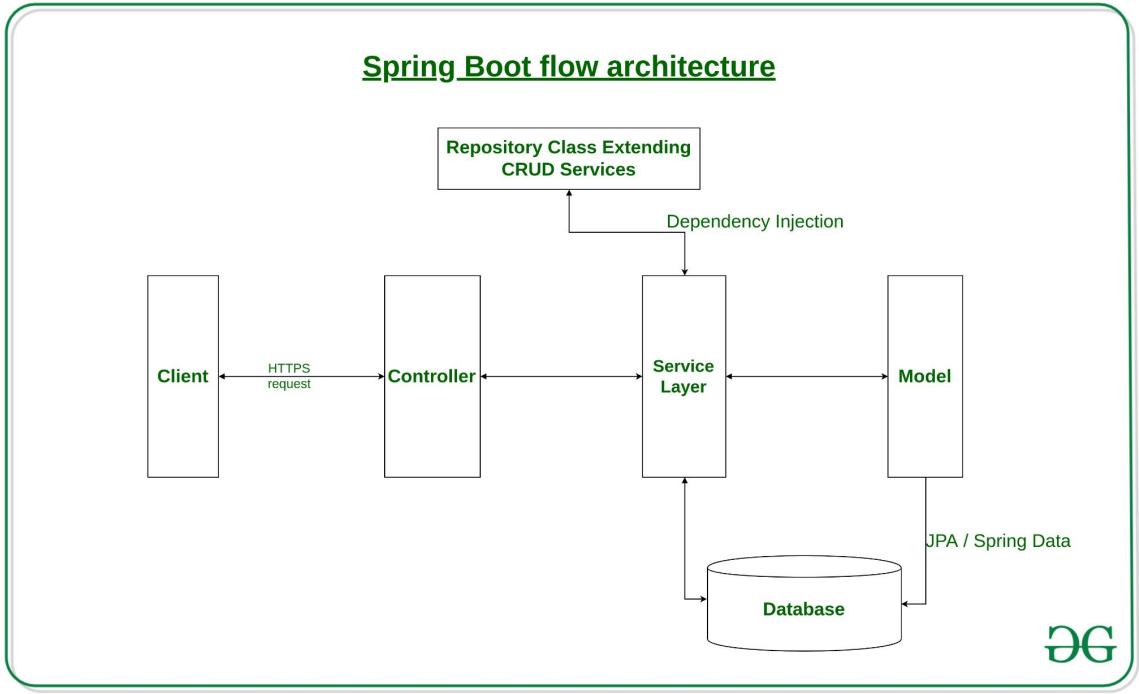
- Recursos accesibles por identificadores (URLs)
- Se comunica mediante protocolo HTTP
- Una misma URL modifica la operación mediante los “verbos”
  - GET: Para operaciones de consulta de datos
  - POST: Operaciones de inserción
  - PUT: Para realizar modificaciones
  - DELETE: Para eliminar un recurso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

3.

Spring Boot 

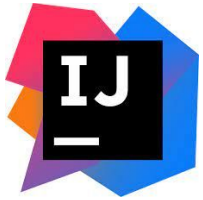
Java Framework open-source para crear APIs REST

Spring Boot


Spring Boot flow architecture
Repository Class Extending CRUD Services
Dependency Injection
Client
HTTPS request
Controller
Service Layer
Model
JPA / Spring Data
Database

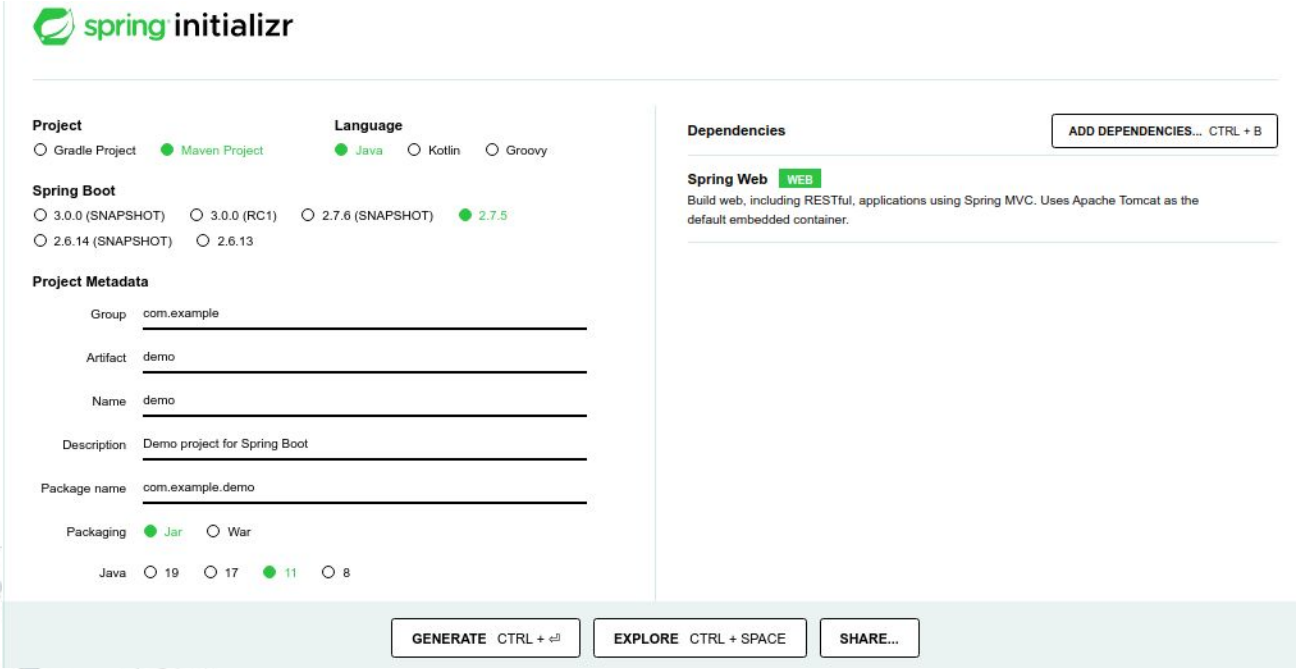
Requisitos básicos

- 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)

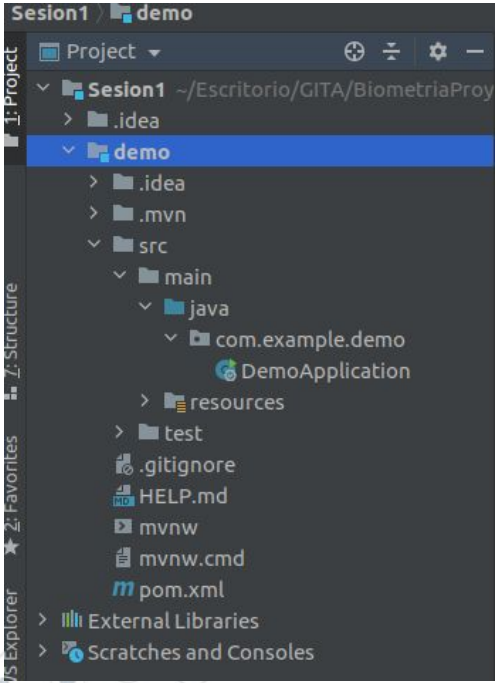


Crear un proyecto Spring Boot

Link: start.spring.io



Crear un proyecto Spring Boot

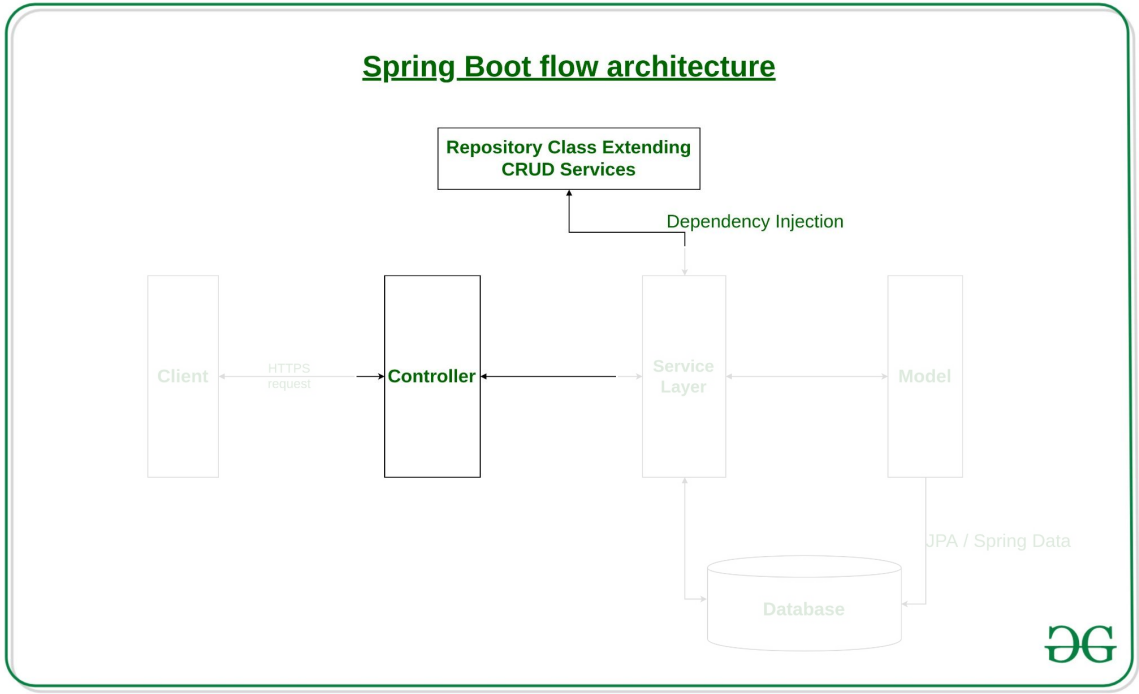



```java
package com.example.demo;

import ...

@SpringBootApplication
public class DemoApplication {

    public static void main(String[] args) {

        SpringApplication.run(DemoApplication.class, args);
    }

}
```

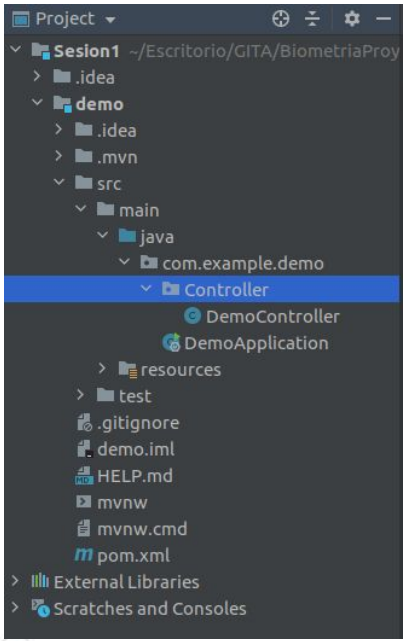
4.

Creación de recursos

Spring Boot


Spring Boot flow architecture
Repository Class Extending CRUD Services
Dependency Injection
Client
HTTPS request
Controller
Service Layer
Model
JPA / Spring Data
Database

Spring Boot




```java
package com.example.demo.Controller;

import org.springframework.web.bind.annotation.PostMapping;
import org.springframework.web.bind.annotation.RequestMapping;
import org.springframework.web.bind.annotation.RestController;

@RestController
@RequestMapping(value = "/")
public class DemoController {

    @PostMapping(value = "/hola")
    public String helloGFG()
    {
        return "Hola Mundo";
    }
}
```

3.

Manos a la obra

Creación del primer recurso

Gracias!

¿Alguna pregunta o comentario?

Puedes escribirme a:

daniel.esobar@udea.edu.co