AMS Bootcamp Fullstack Developer.

**REST y SpringREST**



Índice

[1. Personal Notebook 3](#_Toc157634064)

[1.1 Datos alumno 3](#_Toc157634065)

[1.2 Personal Notebook 3](#_Toc157634066)

[2. REST, Spring REST, Spring Boot 4](#_Toc157634067)

[2.1 Objetivos de aprendizaje 4](#_Toc157634068)

[2.2 ¿Qué es MVC? ¿Cómo funciona (teórico)? 5](#_Toc157634069)

[2.2.1 Objetivos 5](#_Toc157634070)

[2.2.2 Recursos formativos 5](#_Toc157634071)

[2.2.3 Posibles preguntas 5](#_Toc157634072)

[2.2.4 Tareas 5](#_Toc157634073)

[2.2.5 Dudas 5](#_Toc157634074)

[2.3 REST 6](#_Toc157634075)

[2.3.1 Objetivos 6](#_Toc157634076)

[2.3.2 Recursos formativos 6](#_Toc157634077)

[2.3.3 Posibles preguntas 6](#_Toc157634078)

[2.3.4 Tareas 7](#_Toc157634079)

[2.3.5 Dudas 7](#_Toc157634080)

[2.4 Spring Boot 8](#_Toc157634081)

[2.4.1 Objetivos 8](#_Toc157634082)

[2.4.2 Recursos formativos 8](#_Toc157634083)

[2.4.3 Posibles preguntas 8](#_Toc157634084)

[2.4.4 Tareas 8](#_Toc157634085)

[2.4.5 Dudas 8](#_Toc157634086)

[2.5 Spring REST 9](#_Toc157634087)

[2.5.1 Objetivos 9](#_Toc157634088)

[2.5.2 Recursos formativos 9](#_Toc157634089)

[2.5.3 Posibles preguntas 9](#_Toc157634090)

[2.5.4 Tareas 9](#_Toc157634091)

[2.5.5 Dudas 9](#_Toc157634092)

# Personal Notebook

## Datos alumno

* DAS
* Nombre Apellidos
* email
* URL de GitHub corporativo

## Personal Notebook

Este es tu cuaderno de trabajo. En él tendrás que registrar tu aprendizaje, tus avances y tus dudas. ¡Vas a generar tu propio material!

Puede ser una gran herramienta si la utilizas correctamente.

El cuaderno estará compartido con los tutores y el resto de colaboradores para ver tu progreso y ayudarte.

Al final del Bootcamp tendrás que entregarlo como evidencia de tu aprendizaje.



Instrucciones de uso:

El fichero está ubicado en tu carpeta del espacio que tenemos en Teams.

Siempre deberá ser editado en modo online y/o con guardado automático en la nube.

Responde a las preguntas con la información que necesites para dar una respuesta cualificada al formador o a tus compañeros. No es necesario copiar texto solo por “rellenar”. Cita las Fuentes en caso de utilizar otros recursos.

# REST, Spring REST, Spring Boot

## Objetivos de aprendizaje

* Objetivo:
  + **Comprender el funcionamiento de las aplicaciones basadas en REST, las diferencias con arquitecturas web clásicas, los principales métodos y herramientas para trabajar con servicios REST en Spring.**
  + **Conocer las ventajas de usar Spring Boot y su configuración inicial**
* Conceptos principales a adquirir:
  + Arquitectura MCV y REST
  + Spring Boot
  + Spring REST

Lecciones:

1. MVC (teoría)
2. REST (teoría)
3. Sprint Boot
4. Spring REST

## ¿Qué es MVC? ¿Cómo funciona (teórico)?

### Objetivos

1. **Comprender el flujo de las peticiones, desde que sale de un navegador hasta la bbdd.**
2. **Explicar el modelo de capas**

### Recursos formativos

<https://www.youtube.com/watch?v=T_DOdKInZNk&list=PLU8oAlHdN5Blq85GIxtKjIXdfHPksV_Hm&index=26> (26-30)

<https://www.youtube.com/watch?v=z6WppAQ3LUg>

VIP: recuerda que tienes que responder a estas preguntas de forma que te ayude a explicárselas al formador o a los compañeros!!

<https://www.youtube.com/watch?v=-0b6VtyU5S0>

### Posibles preguntas

¿Cómo funciona una petición http?

¿Qué es MVC?

¿Cuáles son los componentes por los que pasa la petición? ¿Para que sirven las 3 capas? ¿Ventajas?

¿Hay otros modelos y/o arquitecturas? (MVVM, MVW, MVP...) ¿Cuándo usar uno u otro? ¿Cuales?

¿Qué es el modelo por capas?

### Tareas

NA

### Dudas

¿?

¿?

VIP: registra las dudas que tengas para la sesión con el formador!!

¿?

¿?

¿?

¿?

## REST

### Objetivos

1. **Comprender el funcionamiento de las aplicaciones basadas en REST,**
2. **Comprender las diferencias con arquitecturas web clásicas,**
3. **Conocer los principales métodos y herramientas para trabajar con servicios REST**

### Recursos formativos

* Píldoras video
  + Intro a REST: <https://www.youtube.com/watch?v=aoOKxUOhg18> (8 min)
  + 6 consejos para que diseñes bien tu API REST <https://www.youtube.com/watch?v=bUmy7Nvsh4s&t=413s> (16 min)
  + Aprende A Documentar Tu API Con Swagger
* Long videos
  + Diseño de APIs REST <https://www.youtube.com/watch?v=3EceTpFV4Ys> (70 min)
  + API Rest. Introducción. Buenas prácticas <https://www.youtube.com/watch?v=t_L88F3lTdQ> (42 min)
* Lecturas
  + <https://www.arquitecturajava.com/servicios-rest/>
  + <https://bravedeveloper.com/2022/06/06/estilos-de-arquitectura-para-construir-apis/>
  + <https://desarrolloweb.com/articulos/que-es-rest-caracteristicas-sistemas.html>
  + <https://www.baeldung.com/spring-rest-openapi-documentation>

### Posibles preguntas

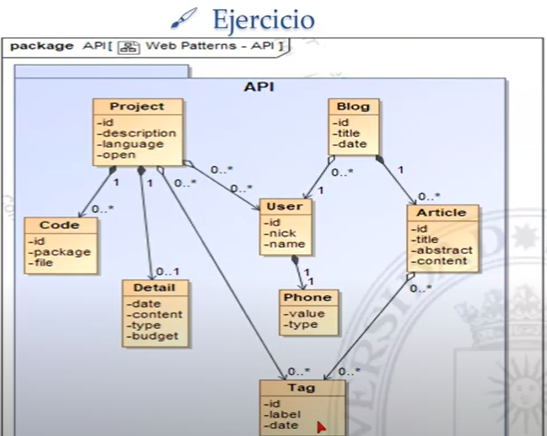
VIP: recuerda que tienes que responder a estas preguntas de forma que te ayude a explicárselas al formador o a los compañeros!!

* ¿Qué significa REST? ¿Para qué se usa? ¿En qué tipo de aplicaciones?
* Explicar la arquitectura de una aplicación Front - Back
* ¿Cómo funcionan las peticiones? ¿Qué formatos de datos puede manejar?
* ¿Qué es API Rest? ¿Cuál es la diferencia con SOAP?
* ¿Qué es el concepto Stateless?
* ¿Cómo puedo trabajar con WebService con Java?
* ¿Qué es el principio HATEOAS?
* En el contexto de Rest: ¿Qué es un Resource?
* ¿Cuáles son los niveles de madurez de REST?

### Tareas

* Plantear distintos Endpoints para el siguiente modelo de datos
* Documentarlo con OpenAPI (Swagger)

Nota: No hay que diseñar la BBDD, son los endpoinds para esos modelos de datos.



### Dudas

¿?

¿?

VIP: registra las dudas que tengas para la sesión con el formador!!

¿?

¿?

¿?

¿?

## Spring Boot

### Objetivos

* **Conocer que es Spring boot, que ventajas aporta y sus configuraciones básicas**

### Recursos formativos

* Píldoras videos
  + <https://www.youtube.com/watch?v=rnyObUvNEdU> (7 min)
* Lecturas:
  + https://spring.io/guides/gs/spring-boot/
  + <https://www.arquitecturajava.com/que-es-spring-boot/>
  + <https://www.arquitecturajava.com/spring-boot-starter-un-concepto-fundamental/>
  + <https://www.arquitecturajava.com/spring-boot-devtools-y-recarga-de-aplicaciones/>
  + <https://www.baeldung.com/spring-boot>
  + https://www.baeldung.com/spring-vs-spring-boot
* Ejemplos
  + <https://github.com/spring-projects/spring-petclinic> (<https://www.baeldung.com/spring-boot-package-structure>)

### Posibles preguntas

* ¿Qué es Spring Boot? ¿Cómo se usa/configura? ¿Qué ventajas aporta?
* ¿Es cierto que solo sirve para crear aplicaciones web?
* ¿Cuál es la diferencia con Spring? ¿Si uso Spring estoy usando Spring Boot?
* ¿Para qué sirve Spring initializer?
* ¿Por qué se asocia tanto Spring Boot a los Microservicios?

### Tareas

### Dudas

## Spring REST

### Objetivos

* **Conocer las opciones que proporciona Spring para crear servicios REST.**
* **Adquirir conocimiento sobre buenas prácticas en diseño de servicios REST**

### Recursos formativos

Lecturas

* + <https://www.arquitecturajava.com/responseentity-spring-y-rest/>
  + <https://www.baeldung.com/spring-rest-openapi-documentation>
  + <https://spring.io/guides/gs/rest-service/>

Curso

* + Cómo crear una API REST (<https://www.youtube.com/watch?v=wT-hIeYyxBg>)
  + Seguridad en Spring Boot <https://www.youtube.com/watch?v=nwqQYCM4YT8>

Antiguos, pero sirven para afianzar conceptos:

* + Seguridad en Spring boot [Protege tu API usando JWT y Spring Security (2022) - YouTube](https://www.youtube.com/watch?v=_p-Odh3MZJc)
  + Curso práctico de APIs REST en Spring Boot con JWT , Spring Security , MySQL y Spring Data JPA <https://www.youtube.com/watch?v=dJaY43Butm8> (complete 5hrs)

### Posibles preguntas

* ¿Cuáles son los componentes de Spring Rest?
* ¿Qué ocurre cuando uso la anotación @RestController? ¿Es distinto a @Controler?
* ¿Cómo podemos gestionar la seguridad en el backend?
* ¿Para qué sirve RestTemplate?
* ¿para qué nos sirve un Header?
* ¿Qué podemos hacer con el ResponseEntity?
* ¿Qué es JWT?
* ¿Cuál crees que son las ventajas y desventajas de usar Spring Rest?

### Tareas

### Dudas

¿?