



**Universidad
Israel**

Programación Avanzada

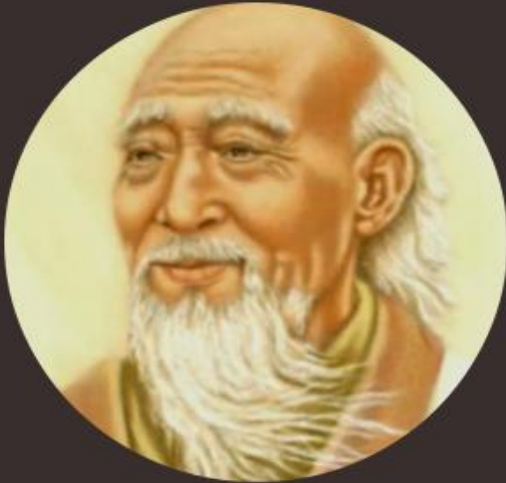
Tema: Spring Boot Framework Desarrollo



Docente: Mg. Mario Rubén Pérez C.



Universidad
Israel



“

Un viaje de mil millas comienza con el
primer paso.
— Lao-Tse —

”



Universidad
Israel

Objetivo

Conocer los concepto básicos
del framework Spring Boot

Contenido

- Antecedentes Previos
- Spring Boot y Spring Boot MVC
- Herramientas
- Crear Proyecto

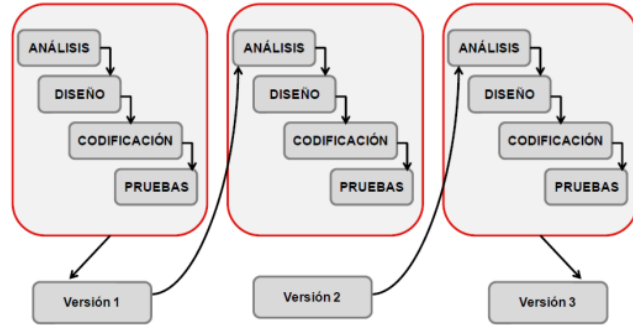


**Universidad
Israel**

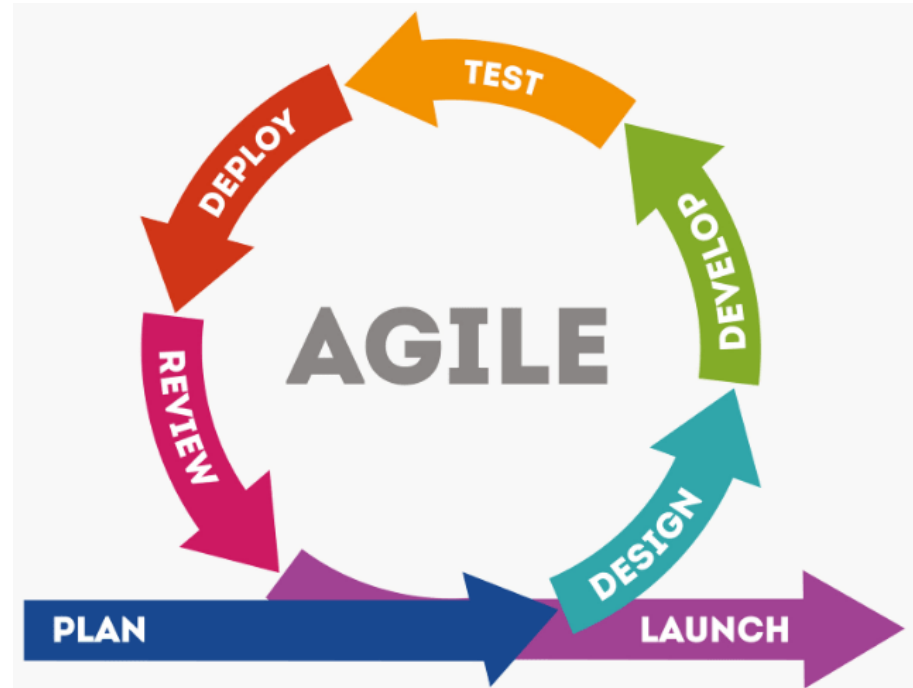
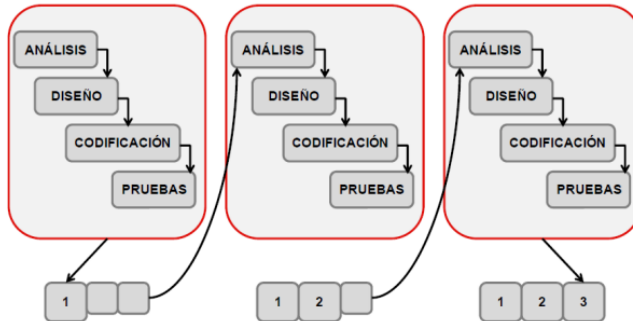
Antecedentes

Antecedentes

Modelo Iterativo



Modelo Incremental



<https://www.youtube.com/watch?v=S0tlXcTP-gw>



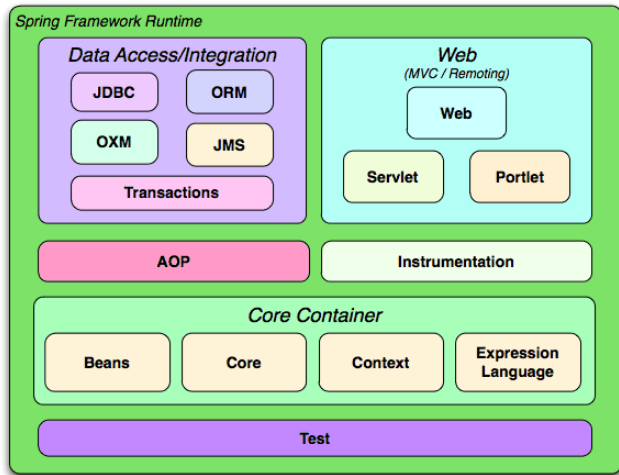
**Universidad
Israel**

Spring Boot y Spring Boot MVC

Antecedentes



Es un *framework* Open Source que facilita la creación de aplicaciones de todo tipo en **Java**, **Kotlin** y **Groovy**.



- **Core container:** proporciona inyección de dependencias e inversión de control.
- **Web:** nos permite crear controladores Web, tanto de vistas MVC como aplicaciones REST.
- **Acceso a datos:** abstracciones sobre JDBC, ORMs como Hibernate, sistemas OXM (*Object XML Mappers*), JSM y transacciones.
- **Programación orientada a Aspectos (AOP):** ofrece el soporte para aspectos.
- **Instrumentación:** proporciona soporte para la instrumentación de clases.
- **Pruebas de código:** contiene un *framework* de *testing*, con soporte para JUnit y TestNG y todo lo necesario para probar los mecanismos de Spring.



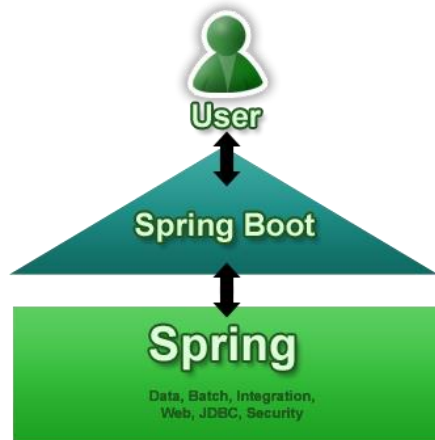
Spring Boot®

Spring Boot es un **framework de Java** que se utiliza para **simplificar y acelerar el proceso de desarrollo** de aplicaciones basadas en Spring. Proporciona características y herramientas para configurar rápidamente una aplicación y manejar la mayoría de los aspectos técnicos de la infraestructura, lo que **permite a los desarrolladores centrarse en la lógica de negocios de la aplicación**. Además, Spring Boot es compatible con múltiples tecnologías y permite una fácil integración con otros frameworks de Spring



Si bien es cierto que Spring Framework es muy potente, **la configuración inicial y la preparación de las aplicaciones para producción** son tareas bastante tediosas. **Spring Boot simplifica el proceso al máximo** gracias a sus dos principales mecanismos. **Contenedor de aplicaciones integrado** (publicar .jar) y **Starters** (proporciona dependencias)

Como funciona Spring Boot



Spring Boot **proporciona una capa de abstracción sobre la plataforma de Spring Framework**, lo que facilita la configuración de una aplicación y reduce la cantidad de código necesario para crear una aplicación basada en Spring. En lugar de configurar manualmente cada una de las funcionalidades de Spring, **Spring Boot ofrece una configuración predeterminada** para muchas de las características comunes, lo que permite a los desarrolladores comenzar rápidamente a desarrollar su aplicación

Además, Spring Boot ofrece herramientas para la gestión de dependencias, la creación de contenedores de aplicaciones, la automatización de pruebas y el empaquetamiento y despliegue de aplicaciones, lo que simplifica el ciclo de desarrollo de la aplicación.

Spring Boot MVC



Spring MVC (Model-View- Controller) es un framework de Java que se utiliza para desarrollar aplicaciones web. Proporciona un patrón de arquitectura de software en el que la aplicación se divide en tres componentes principales: modelo, vista y controlador.

- **El modelo** representa los datos y la lógica de negocios de la aplicación,
- **La vista** representa la interfaz de usuario y
- **El controlador** se encarga de manejar las solicitudes de los usuarios y gestionar la interacción entre el modelo y la vista.

Spring Boot MVC

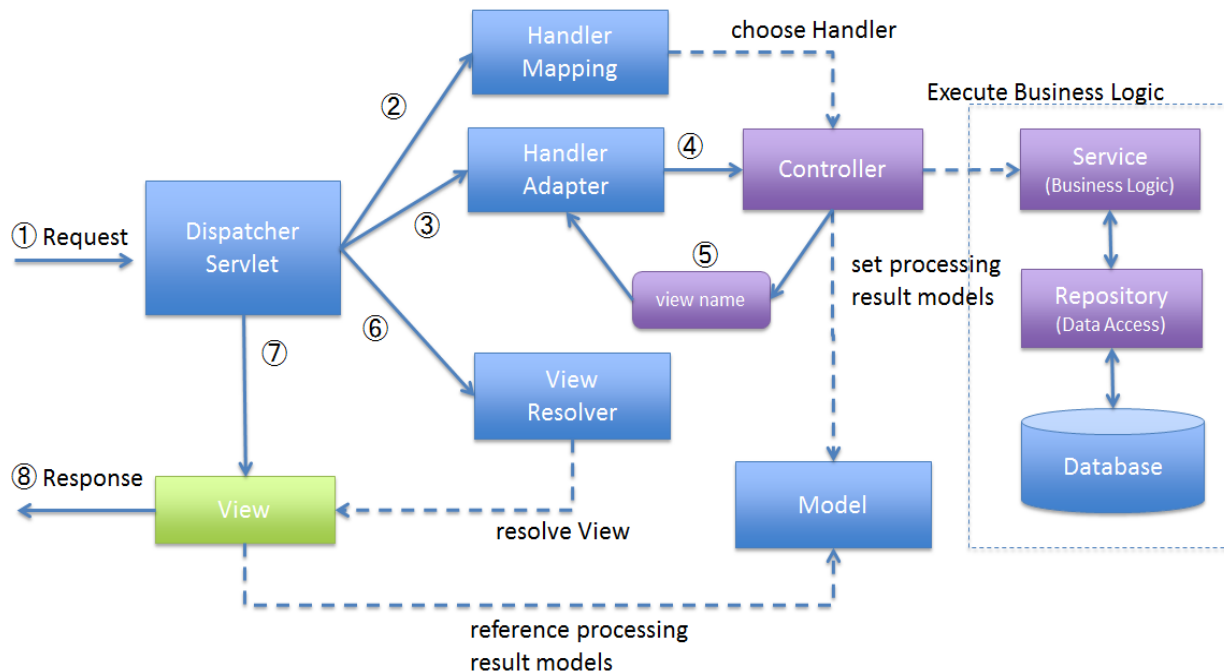
Spring MVC utiliza la plataforma de Spring Framework y proporciona herramientas y características para:

- Gestión de controladores
- Validación de datos
- Internacionalización,
- Gestión de errores,
- La integración con tecnologías web como JSP, Thymeleaf y otros

Además, Spring MVC permite una fácil integración con otros frameworks de Spring, como:

- Spring Security para la autenticación y autorización de usuarios.

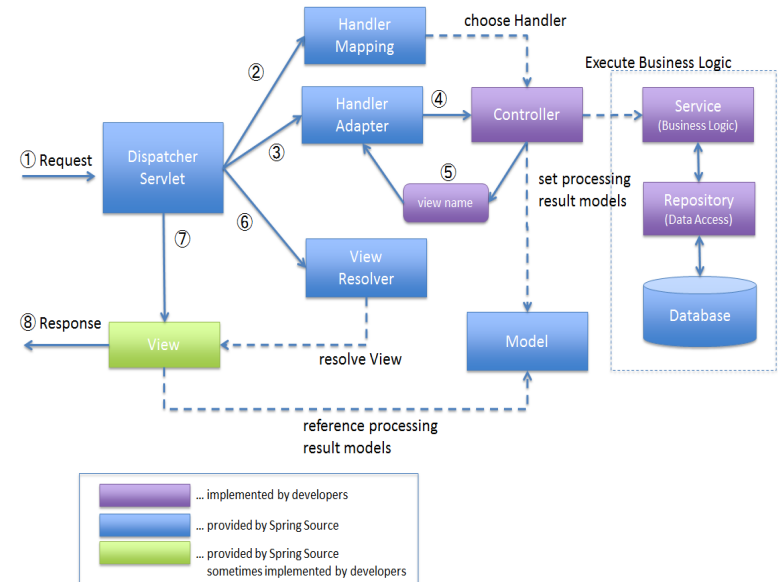
Spring Boot MVC



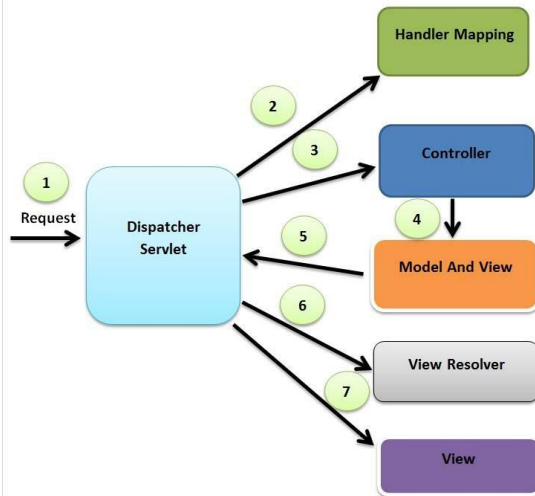
Spring Boot MVC



1. **DispatcherServlet** recibe la solicitud.
2. **DispatcherServlet** envía la tarea de seleccionar un controlador apropiado a **HandlerMapping**. **HandlerMapping** selecciona el controlador que está asignado a la URL de solicitud entrante y devuelve el (manejador seleccionado) y el controlador a **DispatcherServlet**.
3. **DispatcherServlet** despacha la tarea de ejecutar la lógica empresarial de **Controller** a **HandlerAdapter**.
4. **HandlerAdapter** llama al proceso de lógica empresarial de **Controller**.
5. El controlador ejecuta la lógica empresarial, establece el resultado del procesamiento en el modelo y devuelve el nombre lógico de la vista a **HandlerAdapter**.
6. **DispatcherServlet** envía la tarea de resolver la Vista correspondiente al nombre de Vista a **ViewResolver**. **ViewResolver** devuelve la vista asignada al nombre de la vista.
7. **DispatcherServlet** envía el proceso de representación a la Vista devuelta.
8. Ver representa los datos del modelo y devuelve la respuesta.



DispatcherServlet



DispatcherServlet es un componente fundamental del framework Spring MVC. Es el punto de entrada principal de todas las **solicitudes de los usuarios** en una aplicación basada en Spring MVC.

DispatcherServlet actúa como un **Front Controller** (controlador frontal), es decir, recibe todas las solicitudes de los usuarios y las dirige a los controladores correspondientes para su procesamiento.

DispatcherServlet realiza varias tareas importantes, como:

- **Resolución** de las rutas de URL
- **Selección del controlador** adecuado para procesar la solicitud
- **Gestión de las excepciones** y errores que puedan ocurrir durante el procesamiento de la solicitud
- **Generación de la respuesta** que se enviará al usuario.



**Universidad
Israel**

Herramientas

Herramientas a Utilizar



<https://spring.io/>



<https://projectlombok.org/download>



<https://mvnrepository.com/>



<https://www.postgresql.org/download/>



<https://getbootstrap.com/docs/5.3/getting-started/introduction/>



<https://www.thymeleaf.org/>

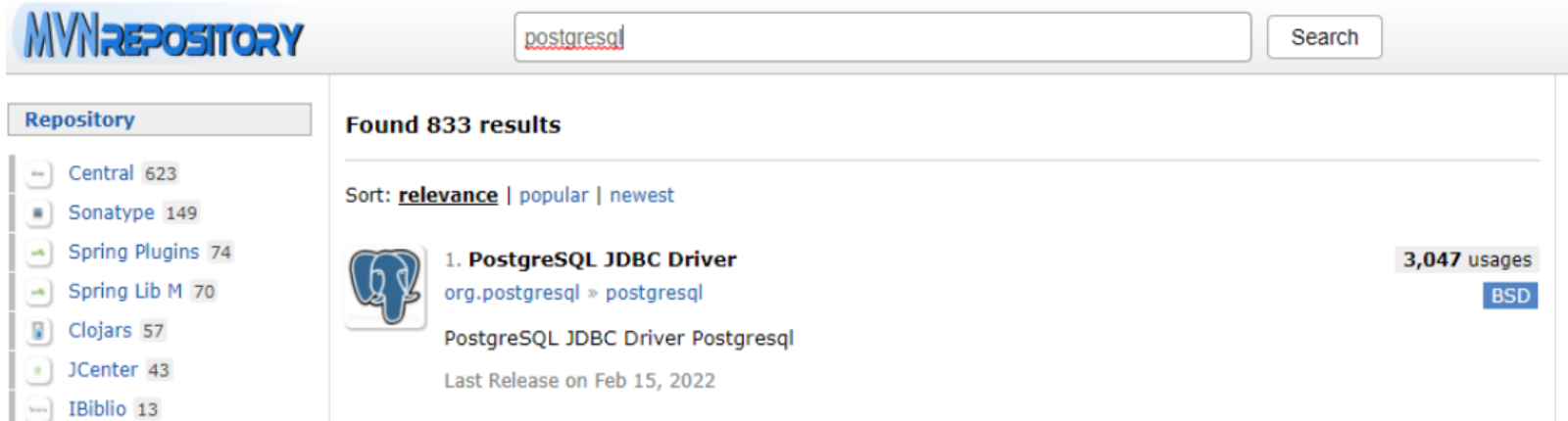
Maven

Maven es una herramienta open-source, que se creó en 2001 con el objetivo de simplificar los procesos de build (compilar y generar ejecutables a partir del código fuente).

Antes de que Maven proporcionara una interfaz común para hacer builds del software, cada proyecto solía tener a alguna persona dedicada exclusivamente a configurar el proceso de build.

<https://www.javiergarzas.com/2014/06/maven-en-10-min.html>






The screenshot shows the Maven Repository search results for the query 'postgresql'. The search bar at the top contains 'postgresql' and a 'Search' button. On the left, a 'Repository' sidebar lists various repositories with their artifact counts: Central (623), Sonatype (149), Spring Plugins (74), Spring Lib M (70), Clojars (57), JCenter (43), and IBiblio (13). The main content area displays 'Found 833 results' and a sort menu with options: **relevance**, popular, and newest. The top result is '1. PostgreSQL JDBC Driver' by 'org.postgresql » postgresql', featuring the PostgreSQL logo (a blue elephant). To the right of the result, it shows '3,047 usages' and a 'BSD' license badge. Below the title, the text 'PostgreSQL JDBC Driver Postgresql' and 'Last Release on Feb 15, 2022' are visible.

Repository

- Central 623
- Sonatype 149
- Spring Plugins 74
- Spring Lib M 70
- Clojars 57
- JCenter 43
- IBiblio 13

Found 833 results

Sort: **relevance** | popular | newest

 **1. PostgreSQL JDBC Driver** 3,047 usages
[org.postgresql](#) » [postgresql](#) BSD
PostgreSQL JDBC Driver Postgresql
Last Release on Feb 15, 2022

<https://mvnrepository.com/>

**PostgreSQL JDBC Driver » 42.3.2**

PostgreSQL JDBC Driver postgresql

License	BSD 2-clause
Categories	PostgreSQL Drivers
Organization	PostgreSQL Global Development Group
HomePage	https://jdbc.postgresql.org
Date	(Feb 01, 2022)
Files	pom (2 KB) jar (1015 KB) View All
Repositories	Central
Used By	3,047 artifacts

Note: There is a new version for this artifact

New Version

42.3.3



Maven

Maven Gradle Gradle (Short) Gradle (Kotlin) SBT Ivy Grape Leiningen Buildr

```
<!-- https://mvnrepository.com/artifact/org.postgresql/postgresql -->
<dependency>
  <groupId>org.postgresql</groupId>
  <artifactId>postgresql</artifactId>
  <version>42.3.2</version>
</dependency>
```

☒ Include comment with link to declaration

Maven Gradle Gradle (Short) Gradle (Kotlin) SBT Ivy Grape Leiningen Buildr

```
// https://mvnrepository.com/artifact/org.postgresql/postgresql
implementation group: 'org.postgresql', name: 'postgresql', version: '42.3.2'
```

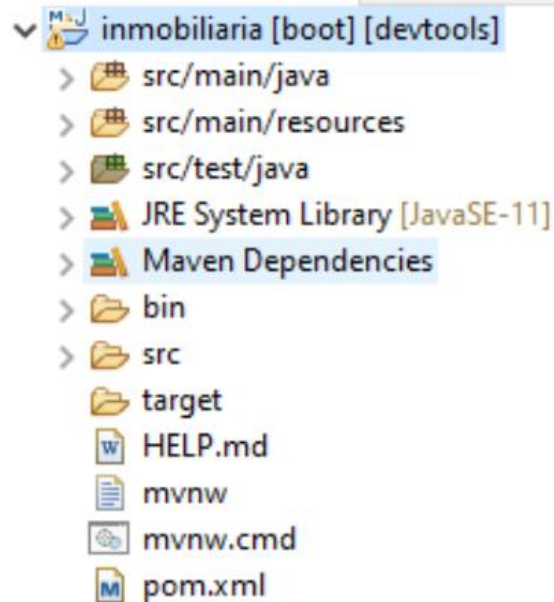
☒ Include comment with link to declaration

¿Qué es un POM ?

Un modelo de objeto de proyecto o POM es la unidad de trabajo fundamental en Maven. Es un archivo XML que contiene información sobre el proyecto y **los detalles de configuración** utilizados por Maven para construir el proyecto.

Contiene valores predeterminados para la mayoría de los proyectos.

- El directorio fuente, que es `src/main/java`
- El directorio fuente de recursos, que es `src/main/resources`
- El directorio fuente de prueba, que es `src/test/java`



¿Qué es un POM ?

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <project xmlns="http://maven.apache.org/POM/4.0.0"
3   xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
4   xsi:schemaLocation="http://maven.apache.org/POM/4.0.0 http://maven.apache.org/xsd/maven-4.0.0.xsd">
5   <modelVersion>4.0.0</modelVersion>
6
7   <parent>
8     <groupId>org.springframework.boot</groupId>
9     <artifactId>spring-boot-starter-parent</artifactId>
10    <version>2.6.3</version>
11    <relativePath /> <!-- lookup parent from repository -->
12  </parent>
13  <groupId>com.uisrael</groupId>
14  <artifactId>inmobiliaria</artifactId>
15  <version>0.0.1-SNAPSHOT</version>
16  <name>inmobiliaria</name>
17  <description>registro_propiedades</description>
18  <properties>
19    <java.version>11</java.version>
20  </properties>
21  <dependencies>
22    <dependency>
23      <groupId>org.springframework.boot</groupId>
24      <artifactId>spring-boot-starter-data-jpa</artifactId>
25    </dependency>
26    <dependency>
27      <groupId>org.springframework.boot</groupId>
28      <artifactId>spring-boot-starter-thymeleaf</artifactId>
29    </dependency>
30    <dependency>
31      <groupId>org.springframework.boot</groupId>
32      <artifactId>spring-boot-starter-web</artifactId>
33    </dependency>

```





**Universidad
Israel**

Estructura de Proyectos

Estructura 1: Por característica

En este enfoque, todas las clases pertenecientes a una característica determinada se colocan en el mismo paquete.

Las ventajas de esta estructura son las siguientes:

- Encontrar una clase para modificar es fácil.
- Al eliminar un subpaquete en particular, se pueden eliminar todas las clases relacionadas con una característica determinada.
- Probar y refactorizar es fácil.
- Las características se pueden enviar por separado


```
com
+-
+- demostración
    +- MiAplicacion.java
    |
+- cliente
    | +- Cliente.java
    | +- CustomerController.java
    | +- Servicio al Cliente.java
    | +- CustomerRepository.java
    |
+- orden
    +- Pedido.java
    +- OrderController.java
    +- OrderService.java
    +- OrderRepository.java
```

Estructura 1: Por característica

Estructura 2: Por capas

La estructura por capa, todos los controladores se pueden colocar en el paquete de controladores y los servicios en el paquete de servicios y todas las entidades en el dominio o modelo, etc.

Las desventajas de esta estructura son las siguientes:

- Las funciones o los módulos no se pueden enviar por separado.
- Es difícil localizar una clase que pertenezca a una determinada característica.
- La refactorización de código en una determinada entidad es difícil debido a que las clases de entidad se encuentran en cada capa.
- Provoca conflictos de fusión entre desarrolladores que utilizan GitHub, BitBucket , etc. para la colaboración..

Spring – Estructura de Proyectos

```
com
+- gfg
  +- demo
    +- MyApplication.java
    |
    +- domain
      +- Customer.java
      +- Order.java
      |
      +- controllers
        +- OrderController.java
        +- CustomerController.java
        |
        +- services
          +- CustomerService.java
          +- OrderService.java
          |
          +- repositories
            +- CustomerRepository.java
            +- OrderRepository.java
```

Estructura 2: Por capas

Crear Proyecto - Base



Project

☒ Gradle - Groovy ☐ Gradle - Kotlin ☐ Maven

Language

☒ Java ☐ Kotlin ☐ Groovy

Spring Boot

☐ 3.1.0 (SNAPSHOT) ☐ 3.1.0 (M1) ☐ 3.0.4 (SNAPSHOT) ☒ 3.0.3 ☐ 2.7.10 (SNAPSHOT) ☐ 2.7.9

Project Metadata

Group

Artifact

Name

Description

Package name

Packaging ☒ Jar ☐ War

Java ☐ 19 ☒ 17 ☐ 11 ☐ 8

**Dependencies**

No dependency selected

Procedimiento

1. **Tipo de proyecto:** Maven
2. **Lenguaje:** Java
3. **Spring Boot:** última versión estable
4. **Grupo:** com.uisrael
5. **Artefacto:** nombre_proyecto
6. **Nombre:** nombre_modulo
7. **Descripción:** no necesaria
8. **Packaging:** Jar
9. **Versión Java:** ultima versión estable

GENERATE CTRL + G

EXPLORE CTRL + SPACE

SHARE...

Crear Proyecto - Dependencias

Dependencies

ADD DEPENDENCIES... CTRL + B

PostgreSQL Driver SQL

A JDBC and R2DBC driver that allows Java programs to connect to a PostgreSQL database using standard, database independent Java code.

Spring Data JPA SQL

Persist data in SQL stores with Java Persistence API using Spring Data and Hibernate.

Lombok DEVELOPER TOOLS

Java annotation library which helps to reduce boilerplate code.

Spring Web WEB

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

Spring Boot DevTools DEVELOPER TOOLS

Provides fast application restarts, LiveReload, and configurations for enhanced development experience.

Thymeleaf TEMPLATE ENGINES

A modern server-side Java template engine for both web and standalone environments. Allows HTML to be correctly displayed in browsers and as static prototypes.


















Procedimiento

1. Conexión a BD
2. ORM Persistencia
3. Código limpio
4. Componentes Web
5. Entorno de Publicación
6. Lenguaje de Expresiones Web



Spring – MVC - Arquitectura






















Arquitectura Maven

- ▼  clinica [boot] [devtools]
 - >  src/main/java
 - >  src/main/resources
 - >  src/test/java
 - >  JRE System Library [JavaSE-17]
 -  target/generated-sources/annotations
 -  target/generated-test-sources/test-annotations
 - >  Maven Dependencies
 - >  bin
 - >  src
 - >  target
 -  HELP.md
 -  mvnw
 -  mvnw.cmd
 -  pom.xml



La arquitectura del proyecto es la necesaria para que maven reconozca un proyecto y puede ser configurable dentro del servidor, sea un .Jar o .War

Arquitectura MVC

- ▼  clinica [boot] [devtools]
 - ▼  src/main/java
 - >  com.uisrael.clinica
 - >  com.uisrael.clinica.controlador
 - >  com.uisrael.clinica.modelo.entidades
 - >  com.uisrael.clinica.modelo.repositorio
 - >  com.uisrael.clinica.servicios
 - >  com.uisrael.clinica.servicios.impl
 - >  src/main/resources
 - >  src/test/java
 - >  JRE System Library [JavaSE-17]
 -  target/generated-sources/annotations
 -  target/generated-test-sources/test-annotations
 - >  Maven Dependencies
 - >  bin
 - >  src
 - >  target
 -  HELP.md
 -  mvnw
 -  mvnw.cmd
 -  pom.xml



**Universidad
Israel**

Gracias

Responsabilidad con pensamiento positivo



Tarea Semana

Crear la arquitectura MVC con Spring Boot para el proyecto de tareas detallado en la plataforma

NOTA: El deber será enviado archivo .rar o .zip de la solución con su nombre