

Guía de aprendizaje

La programación de los microservicios básicos

Exp 1 – Semana 3

Desarrollo Full Stack I (DSY2201)

***Facilitador disciplinar:*** Víctor Rosendo.

***Asesor par:*** Alonso Castillo.

# Índice

[Índice 2](#_Toc151559641)

[Introducción a la semana 3](#_Toc151559642)

[Resultado de aprendizaje 4](#_Toc151559643)

[Conceptos relevantes 5](#_Toc151559644)

[Preguntas activadoras 5](#_Toc151559645)

[Actividad 6](#_Toc151559646)

[Buenas prácticas para desarrollar mejor nuestro microservicio básico 7](#_Toc151559647)

[Aspectos importantes a tomar en cuenta al momento de desarrollar un microservicio 9](#_Toc151559648)

[Fundamentos de los Microservicios y Spring Boot 9](#_Toc151559649)

[Configuración del Entorno de Desarrollo 9](#_Toc151559650)

[Creación de un Proyecto Spring Boot 9](#_Toc151559651)

[Links de Interés 10](#_Toc151559652)

[Videos 10](#_Toc151559653)

[Cierre de la semana 11](#_Toc151559654)

[Referencias 12](#_Toc151559655)

[Apuntes 13](#_Toc151559656)

# Introducción a la semana

Iniciaremos esta semana conociendo de Spring Boot en Visual Studio Code.

Es una disciplina esencial para los desarrolladores modernos. Los microservicios permiten construir aplicaciones altamente escalables y modulares, mientras que Spring Boot simplifica la creación y el despliegue de servicios Java. Esta guía te ayudará a comprender los fundamentos y los conceptos avanzados necesarios para dominar esta tecnología, todo esto mediante la generación de consejos para codificar mediante buenas prácticas y recordar los aspectos más importantes que debes tomar en consideración al momento de comenzar a desarrollar tu actividad sumativa.

# Resultado de aprendizaje

### El estudiante será capaz de:

**RA1.** Configura proyectos con componentes de microservicios, usando tecnologías, herramientas y frameworks, para definir una línea base de trabajo construyendo piezas de backend.

**RA2.** Desarrolla componentes de microservicio creativos, conectándolos entre sí y accediendo a base de datos usando tecnologías del framework de backend, para realizar operaciones CRUD con datos enviados desde frontend.

### Indicador de logro:

**IL1.** Conoce los conceptos de la arquitectura para iniciar el desarrollo de los microservicios.

**IL2.** Conoce las herramientas y los frameworks para la arquitectura.

**IL3.**  Usa IDE (herramienta de trabajo para el frameworks y/ o lenguaje) para desarrollar los microservicios.

**IL8.**  Aplica las operaciones en un repositorio de trabajo colaborativo generado durante el desarrollo de la aplicación.

# Conceptos relevantes

| Microservicio | Controller | Class |
| --- | --- | --- |
| VsCode |  |  |

# Preguntas activadoras

* ¿Qué elementos de JAVA podemos usar en el desarrollo de microservicios?
* ¿Cómo ventajas nos ofrecen el uso de librerías en un proyecto Spring Boot?
* ¿Podemos crear más de un método de manipulación de los datos en el microservicio creado?

# Actividad

### Descripción de la actividad

1. En esta tercera semana se realizará la actividad sumativa de la experiencia 1 la cual tendrá dos instancias. Para la primera parte de manera individual deberán codificar dos microservicios simples asignados por el Docente para practicar lo aprendido en esta primera experiencia de aprendizaje, siguiendo la arquitectura del mismo, bajo el lenguaje y Framework asociados. Finalmente, para la segunda parte deberán realizar un video donde muestren sus microservicios en ejecución mostrando el correcto funcionamiento de todo lo solicitado anteriormente.

# Buenas prácticas para desarrollar mejor nuestro microservicio básico

Codificar un microservicio con Spring Boot en Java es una tarea común en el desarrollo de aplicaciones empresariales. Aquí tienes algunos consejos para mejorar y optimizar tu proceso de desarrollo, así como las bibliotecas comunes que puedes integrar en tu proyecto utilizando VSCode como tu IDE:

### Planificación y diseño

* Antes de comenzar, define claramente los objetivos del microservicio y su alcance.
* Diseña la arquitectura y la estructura de la base de datos si es necesario.
* Utiliza diagramas de clases o modelos de datos para representar las entidades y relaciones.

### Elije las dependencias de Spring Boot

* Usa Spring Initializer para configurar tu proyecto Spring Boot con las dependencias adecuadas. Las dependencias comunes incluyen:
* Spring Web: Para la creación de controladores REST.
* Spring Data JPA: Para la persistencia de datos.
* Spring Boot DevTools: Facilita el desarrollo en tiempo real.
* Spring Boot Actuator: Monitorea y administra la aplicación

### Organiza tu código

* Sigue las mejores prácticas de organización de paquetes para mantener tu código limpio y modular.
* Divide tu aplicación en capas, como controladores, servicios y repositorios, para facilitar el mantenimiento.

### Usa anotaciones de Spring

Aprende a utilizar anotaciones como @RestController, @Service, @Repository, y @Autowired para configurar componentes Spring de manera efectiva.

### Gestión de dependencias

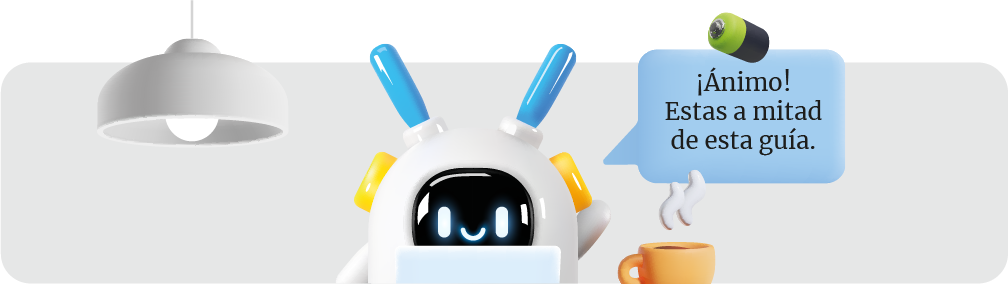
Utiliza herramientas como Maven o Gradle para gestionar las dependencias de tu proyecto de manera eficiente.

### Control de versiones

Utiliza un sistema de control de versiones como Git para llevar un registro de los cambios y colaborar con otros desarrolladores de manera efectiva.

### Mantén tu conocimiento actualizado

Dado que la tecnología está en constante evolución, mantente actualizado leyendo blogs, tutoriales y documentación oficial de Spring Boot.



# Aspectos importantes para tomar en cuenta al momento de desarrollar un microservicio

## Fundamentos de los microservicios y Spring Boot

Comienza con una introducción a los microservicios y Spring Boot. "Los microservicios son una forma de arquitectura de software que descompone aplicaciones en servicios independientes" (Martin Fowler).

Explora las ventajas de los microservicios y cómo Spring Boot facilita su desarrollo. "Spring Boot simplifica la construcción de aplicaciones Java autónomas" (Pro Spring Boot).

## Configuración del entorno de desarrollo

Configura tu entorno de desarrollo con Visual Studio Code y las extensiones necesarias para desarrollar en Spring Boot.

"Visual Studio Code es una herramienta poderosa para desarrollar aplicaciones Java con Spring Boot" (Desarrollador Anónimo).

## Creación de un proyecto Spring Boot

Crea un proyecto Spring Boot utilizando Spring Initializr o Visual Studio Code.

Agrega las dependencias adecuadas, como Spring Web.

"Spring Initializr es una herramienta esencial para comenzar proyectos de Spring Boot" (Desarrollador Anónimo).

# Links de Interés

Mantente al día con las actualizaciones del Framework y profundiza sobre los elementos aprendidos durante la experiencia de aprendizaje en el siguiente enlace:

<https://spring.io/projects/spring-boot>

# Videos

Puedes aprender más con estos vídeos de Youtube:

<https://www.youtube.com/watch?v=5WgmWV7zj9o>

# Cierre de la semana

Los microservicios, como parte integral de la arquitectura de servicios, representan una forma poderosa de diseñar aplicaciones empresariales. Utilizando Java y Spring Boot, hemos aprendido a construir estos servicios de manera efectiva. Hemos comprendido la importancia de una planificación y diseño sólidos, la selección cuidadosa de dependencias de Spring, y la organización del código en capas para una mayor modularidad y mantenimiento. Además, hemos visto cómo utilizar anotaciones de Spring para configurar componentes, gestionar excepciones y validar datos de entrada.

La creación de microservicios con Java Spring Boot implica una combinación de diseño inteligente, buenas prácticas de programación, uso eficaz de herramientas y tecnologías, y una mentalidad de mejora continua. Esta arquitectura proporciona la flexibilidad necesaria para construir aplicaciones escalables, lo que es fundamental en un mundo empresarial en constante evolución. Con un enfoque cuidadoso en estos conceptos y una sólida comprensión de la tecnología, puedes aprovechar al máximo el potencial de los microservicios en tus proyectos de desarrollo de software.

La gestión de dependencias y el control de versiones son aspectos fundamentales de la administración del proyecto, mientras que la automatización de despliegue y el seguimiento de errores contribuyen a la eficiencia del ciclo de desarrollo. Además, mantenerse actualizado con las últimas tendencias y mejores prácticas en el desarrollo de microservicios es crucial para el éxito continuo.



# Referencias

* Turnquist, G. Syer, D. Heckler, M.  Long, J. (2022). Learning Spring Boot 3.0: Simplify the development of production-grade applications using Java and Spring. Packt Publishing

# Apuntes

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_



Reservados todos los derechos Fundación Instituto Profesional Duoc UC. No se permite copiar, reproducir, reeditar, descargar, publicar, emitir, difundir, de forma total o parcial la presente obra, ni su incorporación a un sistema informático, ni su transmisión en cualquier forma o por cualquier medio (electrónico, mecánico, fotocopia, grabación u otros) sin autorización previa y por escrito de Fundación Instituto Profesional Duoc UC La infracción de dichos derechos puede constituir un delito contra la propiedad intelectual.