

### Rúbrica experiencia 1

Evaluación	Porcentaje dentro de la asignatura	Tipo de situación evaluativa	Distribución de porcentajes
Evaluación Parcial N° 2	35%	Entrega de encargo	40%
		Presentación	60% (individual)

### Instrucciones específicas de la Evaluación:

En esta segunda etapa cada equipo continúa con el caso de la experiencia anterior. De acuerdo al diseño de arquitectura de microservicios, el equipo seleccionará e implementará tres servicios a través de API Full Rest comenzando por el servicio de usuarios. Las indicaciones específicas se definen a continuación:

### Situación evaluativa 1: Entrega por encargo

En el informe, los estudiantes deben:

1. Diseñar un proyecto utilizando el framework Spring, demostrando comprensión de su estructura y funcionalidades básicas.
  - a. Gestionar las dependencias y el ciclo de vida del proyecto de microservicios usando Maven, mostrando habilidades en la configuración de archivos POM y en la ejecución de comandos de Maven.
  - b. Construir componentes backend del proyecto, definiendo una línea base de trabajo clara, demostrando habilidades en diseño y desarrollo de servicios REST.
  - c. Demostrar una comprensión de la arquitectura interna de los microservicios y cómo estos se integran en un entorno de desarrollo fullstack, explicando la estructura y la interacción entre los diferentes componentes.
2. El estudiante debe demostrar la aplicación de buenas prácticas de diseño y arquitectura en el desarrollo de componentes de backend, asegurando que el código sea modular, mantenible y escalable.
  - a. El estudiante debe mostrar competencia en el uso de herramientas y frameworks como Spring y Maven para construir componentes de backend, configurando y gestionando correctamente los proyectos.
  - b. El estudiante debe demostrar habilidades de trabajo colaborativo utilizando Git, manejando correctamente el control de versiones, integraciones y revisiones de código para asegurar la funcionalidad y calidad del código generado.
3. Implementar operaciones CRUD para tres API Full Rest, utilizando tecnologías del framework de backend (por ejemplo, JPA, ORM) y garantizar su funcionamiento con datos provenientes de Postman.
4. Validar la comunicación entre microservicios usando Postman, probando y consumiendo los servicios del backend para asegurar que funcionen correctamente.
5. Implementar buenas prácticas en la implementación del proyecto.

*El informe debe incluir los siguientes apartados:*

1. *Portada*
2. *Índice*
3. *Introducción*
4. *Conclusión.*

*El apartado de índice debe considerar el desarrollo de los siguientes temas:*

- a. *Diagrama de arquitectura de microservicios.*
- b. *Estructura del proyecto: dependencias, componentes implementados, detallando cada parte importante del proyecto con su respectiva explicación.*
- c. *Base de datos: explicar motor utilizado, estructura de la o las bases de datos con su respectiva imagen de tablas.*
- d. *Implementación de los servicios: explicar los servicios que se definieron de manera específica, con imágenes de su ejecución en **postman** para las diferentes peticiones a través de API Full Rest interactuando con base de datos.*
- e. *Implementación de **vistas** que permitan desplegar petición get de cada servicio.*
- f. *Git – Git Hub: explicar comandos utilizados (incluir print de pantalla) para la subida de archivos en el repositorio creado en GitHub, proyecto y base de datos. Adicionalmente en archivo Readme de Github añadir path de cada servicio REST con sus peticiones para su ejecución.*

## **Situación evaluativa 2: Presentación**

Durante la presentación, cada estudiante debe:

1. Cada integrante debe presentar, explicar la ejecución de un servicio.
2. Presentar el diseño del proyecto utilizando Spring. Debe demostrar comprensión de la estructura del framework y sus funcionalidades básicas, explicando cómo se ha aplicado en el proyecto.
3. Explicar cómo utilizó Maven para gestionar las dependencias y el ciclo de vida del proyecto. Debe mostrar conocimiento en la configuración de archivos POM y la ejecución de comandos de Maven, proporcionando ejemplos específicos.
4. Detallar la línea base de trabajo establecida y cómo se construyeron los componentes backend. Debe demostrar habilidades en diseño y desarrollo de microservicios, explicando los pasos y decisiones clave en el proceso.
5. Describir cómo configuró el proyecto utilizando herramientas y frameworks específicos (como Spring y Maven). Debe explicar el uso del control de versiones con Git, incluyendo la configuración inicial del repositorio y el seguimiento de cambios en el código.
6. Demostrar comprensión de la arquitectura interna de los microservicios y cómo estos se integran en un entorno de desarrollo fullstack. Debe explicar la estructura de los microservicios y la interacción entre los diferentes componentes del sistema.
7. Exponer sobre la aplicación de buenas prácticas de diseño y arquitectura en el desarrollo de componentes backend. Debe explicar cómo aseguraron que el código fuera modular, mantenible y escalable, proporcionando ejemplos específicos.
8. Presentar el diseño del proyecto utilizando Spring. Debe demostrar comprensión de la estructura del framework y sus funcionalidades básicas, explicando cómo se ha aplicado en el proyecto.

9. Mostrar competencia en el uso de herramientas y frameworks como Spring y Maven para construir componentes de backend. Debe explicar la configuración y gestión de proyectos, detallando cómo utilizaron estas herramientas en su proyecto.
10. Demostrar habilidades de trabajo colaborativo utilizando Git.
11. Describir el desarrollo de componentes de microservicio utilizando tecnologías del framework de backend (como JPA y ORM) para realizar operaciones CRUD. Debe explicar cómo validaron estas operaciones con datos provenientes de Postman.
12. Mostrar cómo validó la comunicación entre microservicios utilizando Postman. Debe explicar cómo probaron y consumieron los servicios del backend para asegurar que funcionen correctamente.

### **Indicaciones Generales**

1. Cada equipo debe ser coordinado por Team Leader.
2. **Fecha de evaluación:** semana comprendida a partir del 26 de mayo.
3. El tiempo asignado para cada equipo es de 10 minutos.
4. Todos los entregables se deben subir en Github, con su respectivo commit "Implementación de servicios REST".
5. El no envío de la información solicitada en la fecha indicada, será calificado con nota mínima.
6. El Team Leader debe enviar url de repositorio por mensaje interno de AVA en la siguiente fecha:
  - a. Ambas Secciones: 26 de mayo hasta 23:59 horas.
7. Si un integrante del equipo se ausenta el día de la presentación, debe justificar su inasistencia como lo indica esta instancia formal.