

# **DOCUMENTACION ACTIVIDAD 1**

**Autor:** Rodrigo Gallardo Romero

**Asignatura:** Arquitecturas Orientadas a Servicios

## Índice

DOCUMENTACION ACTIVIDAD 1 .....	1
INTRODUCCION DEL SISTEMA A DESARROLLAR .....	1
DESCRIPCION DE LOS CASOS DE USO Y SU DIVISION EN MICROSERVICIOS. .....	1
DESCRIPCION DE LA BASE DE DATOS QUE POSEE CADA UNO DE LOS MICROSERVICIOS .....	2
ESQUEMA GENERAL DEL SISTEMA PROPUESTO .....	3
DEFINICION DE LA API DE CADA UNO DE LOS MICROSERVICIOS .....	4
SCRIPTS DE COMPILACION EMPAQUETADO Y EJECUCION .....	5
PROBLEMAS ENCONTRADOS Y CONCLUSIONES .....	5

# INTRODUCCION DEL SISTEMA A DESARROLLAR

En el marco de la arquitectura de microservicios, se han creado tres componentes fundamentales para la gestión eficiente de diferentes aspectos dentro de un sistema integral. Estos microservicios, denominados "gestion\_pedidos," "gestion\_horarios," y "gestion\_clientes," se han diseñado con el objetivo de proporcionar funcionalidades especializadas que aborden distintas áreas de una aplicación distribuida.

## DESCRIPCION DE LOS CASOS DE USO Y SU DIVISION EN MICROSERVICIOS.

**Gestión de Pedidos (gestion\_pedidos):** Este microservicio se enfoca en la administración y seguimiento de pedidos dentro de un sistema. Ofrece capacidades para recuperar detalles específicos de los pedidos, así como para realizar operaciones cruciales, como la eliminación de pedidos previamente registrados. La modularidad de este servicio permite una integración fluida con otros componentes del sistema, asegurando un flujo eficiente de información relacionada con los pedidos.

**Gestión de Horarios (gestion\_horarios):** Concentrándose en la organización temporal, el microservicio "gestion\_horarios" proporciona funcionalidades esenciales para la gestión de horarios y eventos dentro de una aplicación. Desde la recuperación de horarios específicos hasta la eliminación de eventos programados, este servicio permite a otros módulos interactuar de manera eficaz con la información temporal, garantizando una coordinación efectiva en el sistema.

**Gestión de Clientes (gestion\_clientes):** Orientado a la administración de clientes, el microservicio "gestion\_clientes" facilita la recuperación de datos de clientes almacenados en el sistema y ofrece operaciones para gestionar la información asociada a ellos. Desde la obtención de detalles de clientes hasta la eliminación de registros, este componente proporciona una interfaz coherente para la gestión eficiente de información relacionada con los clientes en una arquitectura de microservicios.

Estos tres microservicios, al operar de manera independiente pero interconectada, permiten construir un sistema modular y escalable, donde cada

componente se especializa en una tarea específica, brindando flexibilidad y facilitando el mantenimiento y la expansión del sistema en el futuro.

## DESCRIPCION DE LA BASE DE DATOS QUE POSEE CADA UNO DE LOS MICROSERVICIOS

### Gestión de Pedidos (*gestion\_pedidos*):

- **Entidades Principales:** La base de datos de "*gestion\_pedidos*" albergará principalmente la entidad de "Pedido," que contendrá información vital como el estado del pedido, el cliente asociado, el valor del pedido, las unidades solicitadas, y detalles relevantes para la entrega.
- **Relaciones:** Se establecerán relaciones para vincular pedidos con clientes, asegurando una referencia eficiente a la información del cliente asociado.

### Gestión de Horarios (*gestion\_horarios*):

- **Entidades Principales:** La base de datos de "*gestion\_horarios*" se centrará en la entidad "Horario" o "Evento," que contendrá detalles temporales, descripciones y cualquier otra información relevante para la programación de eventos en el sistema.
- **Relaciones:** No posee ninguna relacion.

### Gestión de Clientes (*gestion\_clientes*):

- **Entidades Principales:** La base de datos de "*gestion\_clientes*" contendrá la entidad "Cliente," que almacenará información clave como el nombre, dirección, y detalles de contacto de los clientes registrados en el sistema.
- **Relaciones:** Posee una relación con el microservicio de "*gestion\_pedidos*" para asociar clientes con los pedidos realizados.

# ESQUEMA GENERAL DEL SISTEMA PROPUESTO

El sistema propuesto se basa en una arquitectura de microservicios, donde cada componente representa una entidad autónoma y especializada que cumple una función específica. Aquí se esboza el esquema general del sistema:

## Configuración y Registro de Servicios:

- **Config Server:** Encargado de centralizar la configuración del sistema y proporcionarla a los demás microservicios.
- **Eureka Server:** Actúa como un servidor de registro y descubrimiento de servicios, permitiendo que los microservicios encuentren y se comuniquen entre sí de manera dinámica.

## Microservicios Principales:

- **Gestión de Pedidos (gestion\_pedidos):** Administra la creación, modificación y eliminación de pedidos, manteniendo la información relacionada con el estado, cliente asociado, y detalles del pedido.
- **Gestión de Horarios (gestion\_horarios):** Maneja la programación y gestión de eventos o actividades, proporcionando información detallada sobre los horarios disponibles.
- **Gestión de Clientes (gestion\_clientes):** Gestiona la información de clientes, incluyendo datos personales, histórico de pedidos, y cualquier detalle relevante

## Microservicio de Usuario (user):

- El microservicio de usuario (user) actúa como un componente central que orquesta la interacción entre los tres microservicios principales: gestión\_pedidos, gestión\_horarios y gestión\_clientes. Su funcionalidad incluye la gestión de las solicitudes de los usuarios finales, coordinando las operaciones que involucran la consulta de pedidos, horarios y clientes.

## Interacción entre Microservicios:

- **Comunicación HTTP/REST:** Los microservicios interactúan entre sí mediante solicitudes HTTP/RESTful, lo que facilita la comunicación y la integración.
- **Eureka Server para Descubrimiento de Servicios:** Los microservicios utilizan Eureka Server para descubrir dinámicamente la ubicación y la disponibilidad de otros servicios en la red.

# DEFINICION DE LA API DE CADA UNO DE LOS MICROSERVICIOS

## **Microservicio de Gestión de Pedidos (gestion\_pedidos):**

El microservicio de gestión de pedidos expone una API que permite realizar operaciones relacionadas con la gestión de pedidos. Algunos de los puntos finales de su API incluyen:

- GET /pedidos: Recupera la lista de todos los pedidos.
- GET /pedidos/{id}: Recupera detalles específicos de un pedido según su ID.
- POST /pedidos: Crea un nuevo pedido.
- PUT /pedidos/{id}: Actualiza los detalles de un pedido existente.
- DELETE /pedidos/{id}: Elimina un pedido según su ID.

## **Microservicio de Gestión de Horarios (gestion\_horarios):**

El microservicio de gestión de horarios proporciona una API centrada en las operaciones relacionadas con la gestión de horarios. Algunos de los puntos finales de su API incluyen:

- GET /horarios: Recupera la lista de todos los horarios disponibles.
- GET /horario/{id}: Obtiene detalles específicos de un horario según su ID.
- POST /horarios: Crea un nuevo horario.
- PUT /horario/{id}: Actualiza los detalles de un horario existente.
- DELETE /horario/{id}: Elimina un horario según su ID.

## **Microservicio de Gestión de Clientes (gestion\_clientes):**

El microservicio de gestión de clientes expone una API para la gestión de información de clientes. Algunos de los puntos finales de su API incluyen:

- GET /clientes: Recupera la lista de todos los clientes registrados.
- GET /cliente/{id}: Obtiene detalles específicos de un cliente según su ID.
- POST /clientes: Crea un nuevo cliente.
- PUT /cliente/{id}: Actualiza la información de un cliente existente.
- DELETE /cliente/{id}: Elimina un cliente según su ID.

# SCRIPTS DE COMPILACION EMPAQUETADO Y EJECUCION

Para poder compilar, empaquetar y ejecutar todos los proyectos se han desarrollado 3 scripts que se encuentran disponibles dentro de la carpeta actividad1. **Se ha de tener en cuenta que dichos scripts están hechos para poder ejecutarlos únicamente a través de Windows.**

Para empaquetar y compilar utilizar el script:

- `script_compilacion_empaquetado.sh`

Para ejecutar utilizar el script:

- `script_ejecucion_sistema.sh`

Para realizar las pruebas utilizar el script:

- `script_ejecucion_pruebas.sh`

## PROBLEMAS ENCONTRADOS Y CONCLUSIONES

La mayoría de problemas que he encontrado han sido siempre a la hora de hacer el empaquetado y compilación de los archivos ya que en muchas ocasiones he obtenido errores y la terminal de visual o la propia cmd no te especificaba nunca el error que está sucediendo específicamente. Esto me ha hecho perder un montón de tiempo para casi todos los micro servicios. En ocasiones se trataba de falta de dependencias en el archivo pom, otras veces dicho archivo no se sincronizaba por lo que no pasaban a ser efectivos los cambios. Por lo que en ocasiones ha resultado ser bastante tedioso.

Sin embargo, creo que me he llevado un gran conocimiento sobre cómo se construyen micro servicios y cómo pueden funcionar entre ellos.