

# DATUM



DATUMREDSOFT.COM

GUATEMALA • EL SALVADOR • HONDURAS • NICARAGUA • COSTA RICA • PANAMÁ

# Agenda del día

- Retroalimentación
- Configuración de IDE
- Introducción a Apache Camel.
- Creación de Proyecto

# Apache Camel

Apache Camel es una biblioteca de integración para Java. Se trata de un lenguaje, un conjunto de APIs en Java y una gran cantidad de componentes que te permiten integrar y procesar datos entre diferentes sistemas informáticos.



# Apache Camel

Beneficios del uso de Apache Camel:

- Camel viene con más de 320 componentes para trabajar con diversas aplicaciones.
- Camel puede comunicarse con aplicaciones como Twitter, Facebook y Salesforce.
- Comunicación entre diversas APIs con pocas sentencias.



# Apache Camel

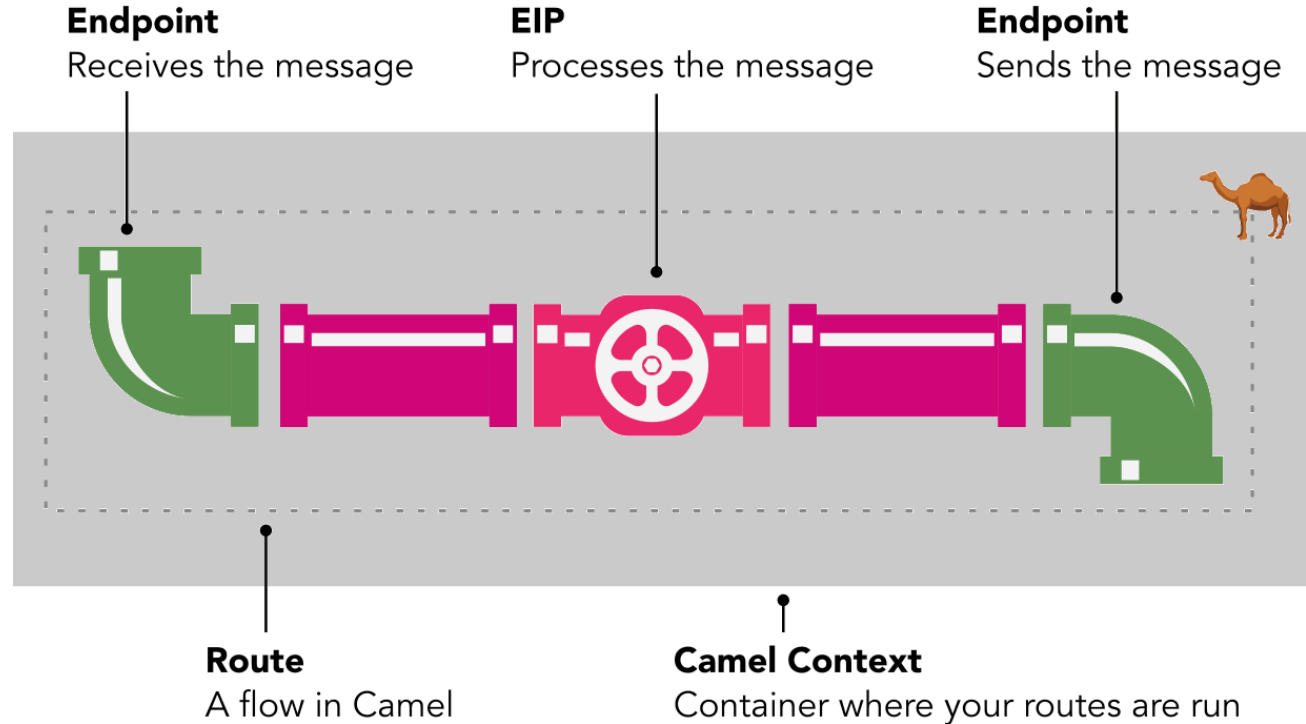
## Uso de Apache Camel

Está más adecuado para situaciones en las que deseas obtener datos de archivos o aplicaciones, procesarlos y combinarlos con otros datos, y luego transferirlos a otra aplicación.



# Apache Camel

Un ejemplo básico de Camel es la fontanería, una ruta como un conducto que mueve datos de un lugar a otro. Transporta datos entre cosas llamadas puntos finales (endpoints).



# Apache Camel

**Enterprise Integration Patterns:** Cuando deseas realizar algunas actividades comunes en un mensaje, como transformación, división y registro, utilizarás un Patrón de Integración Empresarial (EIP, por sus siglas en inglés). Aquí hay algunos EIP comunes en Camel:

EIP (Name)	Java Syntax
Log	.log()
From	from()
To	.to()



# Apache Camel

**Componentes:** Cada vez que se necesite introducir o extraer datos de una aplicación, ya existe un componente de Camel para hacer el proceso más fácil. En otras palabras puede simplificar el código para leer un archivo o invocar un servicio web. Entre algunos se pueden destacar:

Componente	Java Syntax
File	file:
Direct	direct:
HTTP	http:





# Configuración IDE

Apache Camel puede ser utilizado en diversos IDE, entre los cuales podemos mencionar:

- Red Hat CodeReady Studio
- Spring Tool Suit 4
- Eclipse
- Visual Studio Code
- IntelliJ Idea
- NetBeans



# Configuración IDE

Para los IDE como Eclipse, NetBeans, IntelliJ Idea y Visual Studio Code, es necesario la instalación de la extensión necesario o configuración adicionales para el correcto funcionamiento del Framework Apache Camel. Entre las configuraciones adicionales se destaca:

- Instalación de una Java JDK a nivel del sistema operativo.
- Instalación del repositorio Maven.



# Configuración IDE

Con relación a los IDE como Spring Tool Suit 4 y Red Hat CodeReady Studio, nos brindan diversos beneficios entre los cuales destacamos:

- No tener que instalar varias versiones de Java JDK.
- No tener que instalar Maven a nivel de la máquina.
- Contar con un servidor de aplicaciones embebido.



# Anotación

Una anotación es una forma de añadir metadatos al código fuente que están disponibles para la aplicación en su tiempo de ejecución o de compilación

Anotaciones destacadas:

- SpringBootApplication
- Component
- Configuration
- ComponentScan



# Lectura de Properties

Las properties son un modo conveniente de proporcionar pares clave-valor que podremos utilizar desde nuestra propia aplicación. De hecho, lo que podemos definir mediante una property es una clave, a la que le asignaremos un valor, y en nuestra aplicación podemos usar esa clave.



# Lectura de Properties

```
camel.servlet.mapping.context-path=/api/*  
  
spring.datasource.url=${DATASOURCE_URL}  
spring.datasource.username=${DATASOURCE_USER}  
spring.datasource.password=${DATASOURCE_PWD}  
spring.datasource.driver-class-name=com.mysql.cj.jdbc.Driver  
spring.jpa.properties.hibernate.dialect=org.hibernate.dialect.MySQLDialect  
spring.jpa.show-sql=false
```



# Lectura de Properties

## Anotación: Value

```
import org.apache.camel.builder.RouteBuilder;

@Component
public class ApacheCamelEndPoint extends RouteBuilder{

    @Value("${camel.servlet.mapping.context-path}")
    private String contextPath;

    @Override
    public void configure() throws Exception {
```

# Patrón

## From

Se asocia comúnmente con el concepto de "Endpoint" en Camel, que actúa como el origen de los mensajes para esa ruta específica. Es importante entender qué tipo de Endpoint estás utilizando como fuente, ya que esto determinará cómo Camel interactúa con el sistema subyacente.





# Patrón

## To

En el contexto de Apache Camel, el patrón TO se utiliza para dirigir el flujo del mensaje hacia un destino específico. En el patrón TO, se especifica un Endpoint que representa el destino al cual se enviará el mensaje.



# Componentes

## Direct

El componente Direct proporciona la invocación directa y sincrónica de cualquier consumidor cuando un productor envía un intercambio de mensajes. Este componente se puede utilizar para conectar rutas existentes en el mismo contexto de Camel.



# Componentes

## Bean

Camel solo admite el modo de productor. Esto significa que se utiliza principalmente para invocar métodos en beans y enviar los resultados a través de los intercambios de mensajes de Camel.



# Ejercicio

Desarrolle un nuevo EndPoint con las siguientes características:

1. En el mismo componente **rest** agregue 2 método http tipo GET.
2. El primer Get debe recibir un parámetro usando PathParam en conjunto con el patron **.to** y el componente **bean**.
3. El segundo Get debe recibir un parámetro usando QueryParam en conjunto con el patrón **from, to** y el componente direct.



# Patrón

## WireTap

Camel copiará el Intercambio original y establecerá su Patrón de Intercambio en "InOnly", ya que queremos que el Intercambio tipeado se envíe de forma "fire and forget" (disparar y olvidar). El Intercambio tipeado se envía entonces en un hilo separado, permitiendo que se ejecute en paralelo con el original.



# Componente

## SEDA

SEDA (Staged Event-Driven Architecture) en Apache Camel proporciona un comportamiento asíncrono utilizando la arquitectura SEDA. Esta arquitectura implica el intercambio de mensajes a través de una cola de bloqueo (BlockingQueue), donde los consumidores son invocados en un hilo separado del productor.





Gracias por la Atención

