

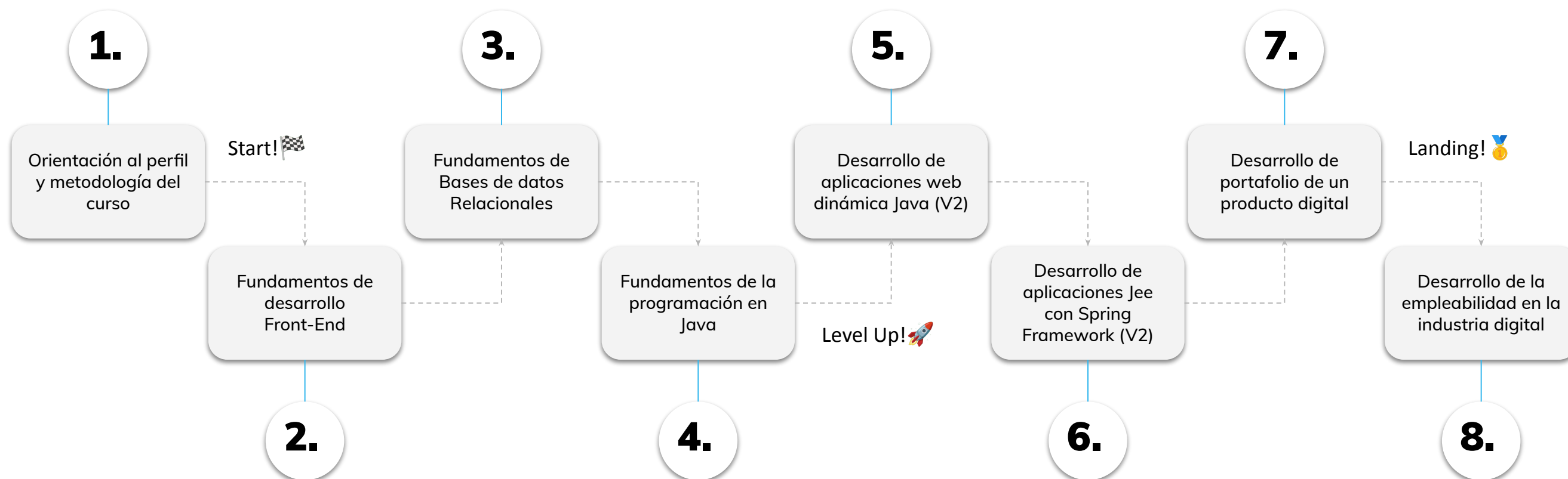
# ➤ La interoperabilidad entre los sistemas

---

**Plan formativo:** Desarrollo de Aplicaciones Full Stack Java Trainee V2.0

# HOJA DE RUTA

¿Cuáles **skill** conforman el programa?



# REPASO CLASE ANTERIOR


En la clase anterior trabajamos :

- ✓ Autenticación contra una base de datos utilizando JPA

# LEARNING PATHWAY

¿Sobre qué temas trabajaremos?

6.5

Start! 

**La interoperabilidad  
entre los sistemas**

Protocolos de intercambios  
de datos

Preguntas Frecuentes

La interoperabilidad entre sistemas  
Protocolos de intercambio de datos

# OBJETIVOS DE APRENDIZAJE

¿Qué aprenderemos?



***Conocer el concepto interoperatividad entre sistemas***



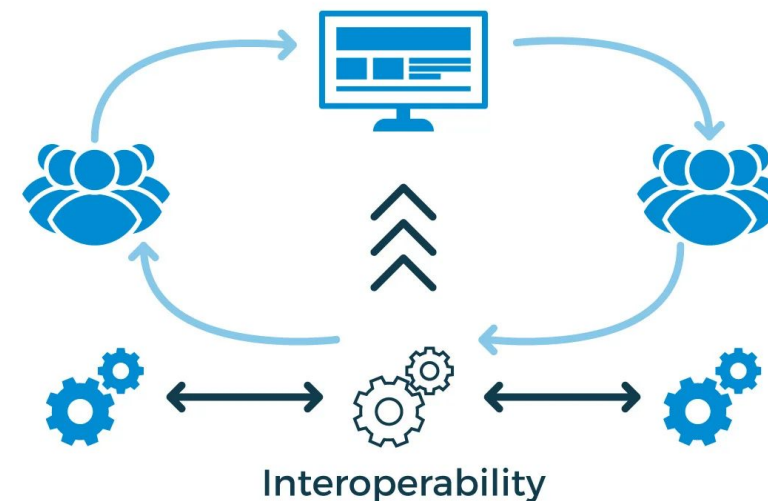
***Comprender los protocolos de intercambio de datos***



# ➤ La interoperabilidad entre los sistemas

# La interoperabilidad entre los sistemas

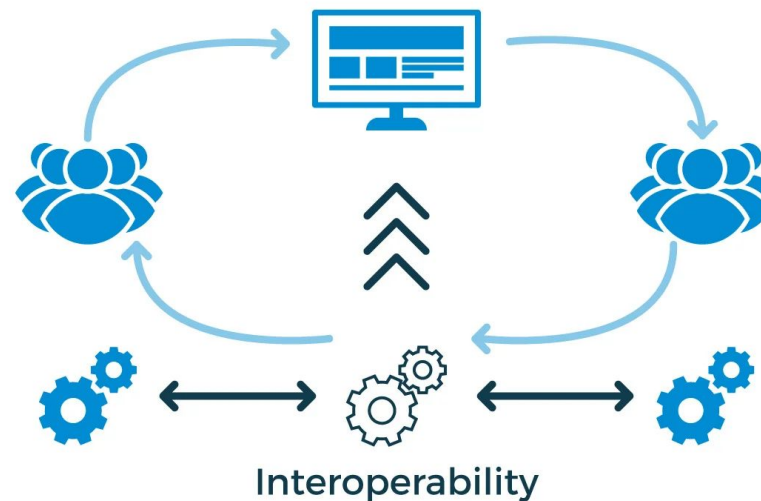
La interoperabilidad entre sistemas es un concepto clave en la arquitectura de software moderna. Se refiere a la **capacidad de diferentes sistemas, plataformas o aplicaciones para intercambiar datos y operar de manera conjunta de manera efectiva**. Este aspecto es crucial en un entorno tecnológico diverso, donde coexisten diferentes tecnologías, lenguajes de programación, arquitecturas y protocolos.



# La interoperabilidad entre los sistemas

Para lograr la interoperabilidad entre sistemas es necesario que:

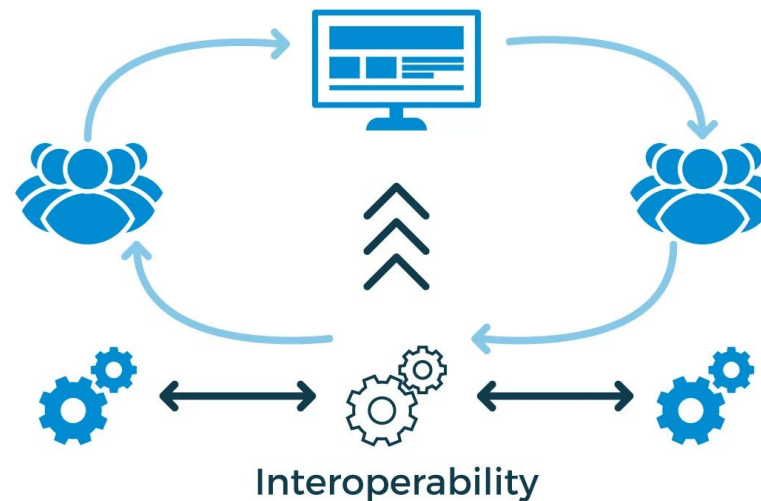
- **Hablen un lenguaje común:** esto se logra definiendo protocolos y formatos estándar para la comunicación.
- **Tengan interfaces compatibles:** las APIs de cada sistema deben poder conectarse e interactuar sin problemas.





# La interoperabilidad entre los sistemas

La interoperabilidad se define como la capacidad de diferentes sistemas para trabajar juntos y compartir información de manera efectiva. Es un principio esencial en el diseño de sistemas, ya que permite la integración fluida y la comunicación entre diversas tecnologías sin perder la funcionalidad.





# La interoperabilidad entre los sistemas



## Importancia de la Interoperabilidad:

- **Integración sin Fricción:** Facilita la integración sin problemas de sistemas heterogéneos, permitiendo que trabajen como un todo coherente.
- **Flexibilidad y Adaptabilidad:** Facilita la incorporación de nuevas tecnologías y la adaptación a cambios en el entorno tecnológico sin afectar la funcionalidad existente.
- **Reutilización de Recursos:** Permite la reutilización eficiente de recursos, ya que los sistemas pueden utilizar componentes y servicios de otros sistemas.





# La interoperabilidad entre los sistemas



## Conceptos claves de la Interoperabilidad

La interoperabilidad, en el contexto de sistemas distribuidos, engloba una serie de conceptos esenciales que son fundamentales para lograr una comunicación efectiva entre diferentes componentes o sistemas. Estos conceptos clave son la piedra angular que permite que sistemas heterogéneos trabajen de manera conjunta y compartan información de manera significativa.





# La interoperabilidad entre los sistemas

## Conceptos claves de la Interoperabilidad



**1. Sintaxis y Semántica:** Estos dos conceptos trabajan en conjunto para garantizar que los sistemas comprendan la forma y el significado de los datos intercambiados, esencial para evitar malentendidos y asegurar una comunicación precisa.



- **Sintaxis:** Este concepto se refiere a la estructura y el formato de los datos intercambiados entre sistemas. En otras palabras, se trata de cómo se organizan y presentan los datos. Un ejemplo común es el formato **JSON**, que define la estructura de los datos utilizando pares **clave-valor**.
- **Semántica:** Mientras que la sintaxis se ocupa de la estructura, la semántica se centra en el significado de los datos. Incluye la **interpretación y comprensión de la información intercambiada**. Por ejemplo, en un contexto de intercambio de datos sobre productos, la semántica podría especificar que el campo "precio" representa la cantidad en la moneda local.



# La interoperabilidad entre los sistemas

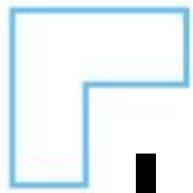
## Conceptos claves de la Interoperabilidad



### 2. Mensajes y Paquetes:

- **Mensaje:** En el contexto de interoperabilidad, un mensaje es la unidad básica de datos enviada de un sistema a otro. Puede contener información específica y estar estructurado de acuerdo con la sintaxis y la semántica acordadas entre los sistemas. Por ejemplo, en un sistema de pedidos en línea, un mensaje podría contener detalles sobre un artículo comprado.
- **Paquete:** Los mensajes no siempre viajan solos; a menudo, se agrupan en paquetes. Un paquete es simplemente un conjunto de datos agrupados para facilitar la transmisión eficiente entre sistemas. La segmentación de información en paquetes ayuda a optimizar la velocidad y la confiabilidad de la transmisión.





# La interoperabilidad entre los sistemas



## Conceptos claves de la Interoperabilidad

3. **Protocolos de Comunicación:** Los protocolos de comunicación son conjuntos de reglas y convenciones que rigen la forma en que los sistemas intercambian datos. Algunos ejemplos comunes incluyen: +

- **TCP/IP**
- **HTTP/HTTPS**
- **REST**
- **SOAP**





# La interoperabilidad entre los sistemas




## Conceptos claves de la Interoperabilidad

4. **Arquitecturas Orientadas a Servicios** (SOA): es un enfoque arquitectónico que utiliza servicios independientes y bien definidos para lograr la interoperabilidad. Los servicios se comunican a través de estándares como SOAP y REST.



## 5. Orquestación y Coreografía:

- **Orquestación:** Control centralizado de la secuencia de ejecución de servicios en un flujo de trabajo.
-  **Coreografía:** Coordinación descentralizada donde los servicios interactúan de manera autónoma sin un control central.



# La interoperabilidad entre los sistemas

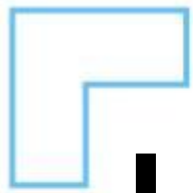


**Ventajas** de la interoperabilidad:

- **Flexibilidad y Adaptabilidad:** Facilita la integración de nuevas tecnologías, actualizaciones y cambios en el entorno tecnológico. Los sistemas interoperables son más adaptables a la evolución del panorama tecnológico.
- **Reutilización de Recursos:** Permite la reutilización eficiente de servicios, componentes y datos. Los sistemas pueden beneficiarse de los recursos existentes sin tener que reinventar la rueda.







# La interoperabilidad entre los sistemas



**Ventajas** de la interoperabilidad:

- **Eficiencia y Mejora del Rendimiento:** Facilita la comunicación eficiente y la optimización de procesos. Los sistemas interoperables pueden operar de manera más eficiente y brindar un mejor rendimiento.
- **Reducción de la Dependencia Tecnológica:** Permite la integración de sistemas independientemente de la tecnología subyacente. Se reduce la dependencia de una tecnología específica, lo que facilita la evolución y la actualización.






# La interoperabilidad entre los sistemas



**Desventajas** de la interoperabilidad:

- **Complejidad en la Implementación:** La implementación de interoperabilidad puede ser  compleja, especialmente en entornos con tecnologías diversas. Requiere una planificación cuidadosa y la gestión de la complejidad técnica.
- **Seguridad y Gobernanza:** La interoperabilidad mal gestionada puede plantear riesgos de seguridad. Es necesario establecer políticas de gobernanza y aplicar prácticas de seguridad robustas.





# La interoperabilidad entre los sistemas



**Desventajas** de la interoperabilidad:

- **Compatibilidad de Estándares:** La variabilidad en la adopción de estándares puede causar problemas de compatibilidad. Se deben seleccionar estándares comunes y respaldados por la industria para evitar conflictos.
- **Costos de Implementación:** La implementación y mantenimiento de la interoperabilidad pueden generar costos significativos. Se debe equilibrar la inversión con los beneficios esperados y la escalabilidad a largo plazo.





# La interoperabilidad entre los sistemas



## Consideraciones adicionales para mejores prácticas:

- **Documentación Exhaustiva:** Documenta claramente los protocolos, formatos de datos y puntos finales para facilitar la integración.
- **Pruebas Rigurosas:** Implementa pruebas unitarias y de integración para garantizar la interoperabilidad en diversas condiciones.
- **Gestión de Versiones:** Establece prácticas de gestión de versiones para evitar conflictos y asegurar la compatibilidad.
- **Monitoreo y Resolución de Problemas:** Implementa herramientas de monitoreo para identificar y resolver problemas de interoperabilidad de manera proactiva.





# La interoperabilidad entre los sistemas



La interoperabilidad entre sistemas es esencial en el panorama tecnológico actual, donde la diversidad tecnológica es la norma. Aunque presenta desafíos, los beneficios en términos de **flexibilidad, eficiencia y reutilización** de recursos hacen que la inversión en interoperabilidad sea valiosa. Al seguir mejores prácticas, gestionar la complejidad y adaptarse a estándares comunes, las organizaciones pueden construir mejores ecosistemas tecnológicos que impulsan la innovación y el crecimiento continuo.



# ➤ Protocolos de intercambio de datos



# Protocolos de intercambio de datos



## Navegando por el Mundo de la Comunicación Digital

Los protocolos de intercambio de datos son los lenguajes fundamentales que permiten la comunicación entre sistemas distribuidos. Estos protocolos definen las reglas y el formato para el intercambio de información, facilitando la interoperabilidad y la transmisión de datos entre aplicaciones y dispositivos.



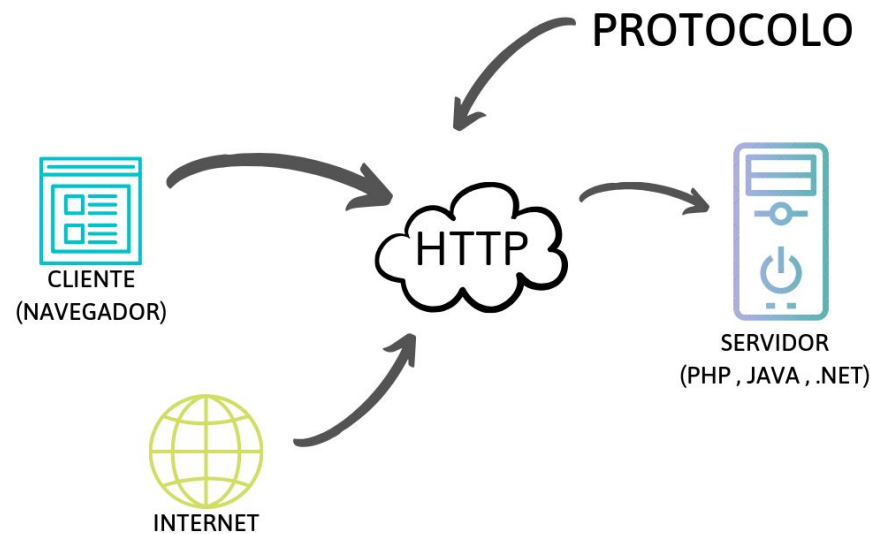
La elección del protocolo de intercambio de datos es fundamental para el éxito de la interoperabilidad entre sistemas. Veamos algunos de los protocolos más comunes, examinando sus características, aplicaciones y ventajas.



# Protocolos de intercambio de datos

**1- HTTP** (Protocolo de Transferencia de Hipertexto): es el protocolo fundamental para la transferencia de datos en la World Wide Web. Opera sobre el modelo cliente-servidor, donde el cliente realiza solicitudes y el servidor responde con los recursos solicitados.

Es ampliamente utilizado en aplicaciones web para la transferencia de documentos, imágenes y otros recursos.







# Protocolos de intercambio de datos



## HTTP: Características Principales

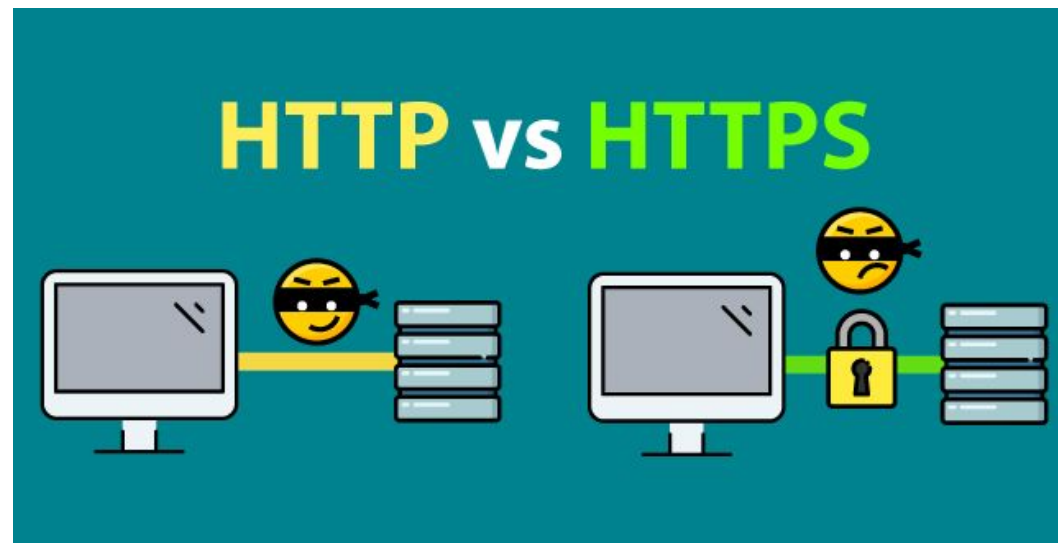
- Utiliza los métodos estándar como GET (obtener), POST (enviar datos al servidor), PUT (actualizar) y DELETE (eliminar).
- Es un protocolo sin estado, lo que significa que cada solicitud del cliente es independiente y no mantiene información sobre solicitudes anteriores.
- Simple y fácil de implementar.
- Compatible con una variedad de tipos de datos.



# Protocolos de intercambio de datos

**2- HTTPS** (HTTP Seguro): **HTTPS es la versión segura de HTTP que utiliza capas de cifrado**, generalmente SSL/TLS, para garantizar la seguridad de la comunicación entre el cliente y el servidor.

Es crucial para aplicaciones que manejan información sensible, como datos de tarjetas de crédito o contraseñas.





# Protocolos de intercambio de datos



## **HTTPS** Características Principales:

- Utiliza certificados digitales para autenticar la identidad del servidor.
- Proporciona cifrado de extremo a extremo para proteger la privacidad de los datos.
- Asegura la confidencialidad y la integridad de los datos transmitidos.

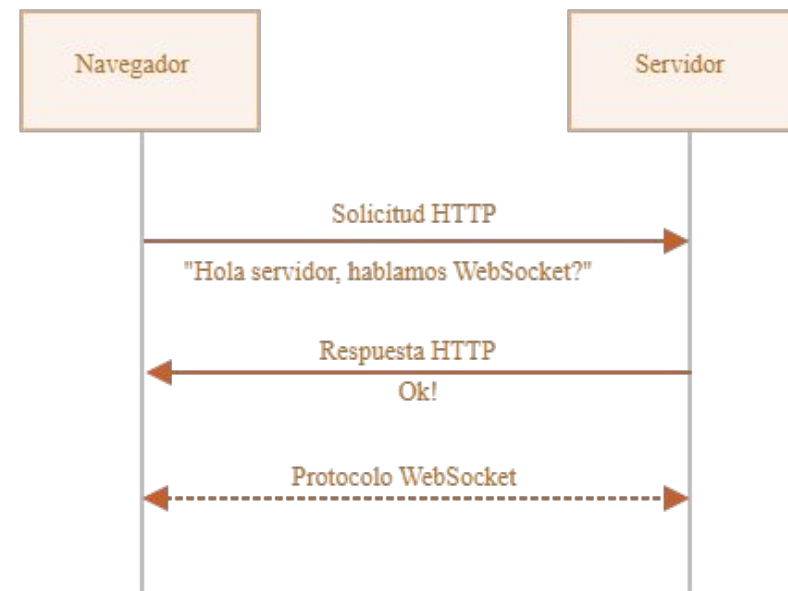




# Protocolos de intercambio de datos

**3- WebSocket:** es un protocolo de comunicación bidireccional que permite la comunicación en tiempo real entre un cliente y un servidor. A diferencia de HTTP, WebSocket establece una conexión persistente, lo que permite una comunicación más eficiente y de baja latencia.

Principalmente utilizado en aplicaciones que requieren actualizaciones en tiempo real, como chats en línea y transmisiones en vivo.





# Protocolos de intercambio de datos



## **WebSocket:** Características Principales

- Permite una comunicación bidireccional, lo que significa que tanto el cliente como el servidor pueden enviar mensajes en cualquier momento.
- Reduce la sobrecarga de la comunicación al eliminar la necesidad de establecer una conexión nueva para cada solicitud.
- Mayor eficiencia en comparación con los enfoques tradicionales basados en HTTP.

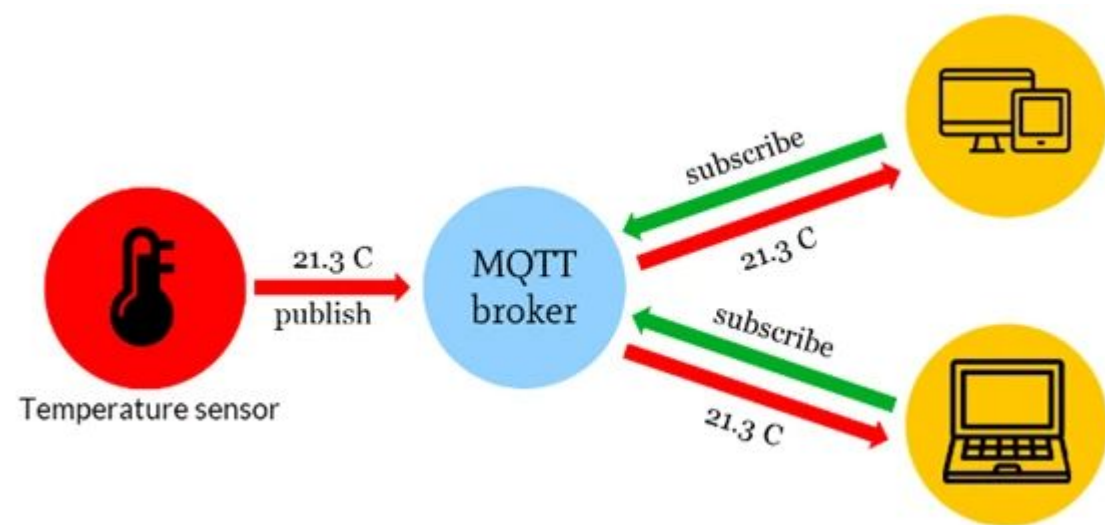


# Protocolos de intercambio de datos

## 4- MQTT (Protocolo de Telemetría de Mensajería y Calidad de Servicio Mínimo):

Es un protocolo ligero de mensajería diseñado para situaciones de baja capacidad y alta latencia, siendo ideal para el Internet de las cosas (IoT).

Ampliamente utilizado en entornos IoT para la transmisión eficiente de datos entre dispositivos.





# Protocolos de intercambio de datos



## **MQTT:** Características Principales

- Utiliza un modelo de publicación/suscripción donde los dispositivos pueden publicar mensajes en "temas" a los que otros dispositivos pueden suscribirse.
- Ofrece diferentes niveles de calidad de servicio (QoS) para garantizar la entrega de mensajes.
- Eficiente en términos de ancho de banda y recursos.



# Protocolos de intercambio de datos

**5- XML** (Lenguaje de Marcado Extensible): es un lenguaje de marcado que permite la representación estructurada de datos mediante el uso de etiquetas. Es ampliamente utilizado para el intercambio de datos entre sistemas heterogéneos.

Comúnmente utilizado en la representación de datos en documentos y configuraciones.







# Protocolos de intercambio de datos



## **XML:** Características Principales



- Define reglas para la creación de documentos estructurados mediante el uso de etiquetas anidadas.
- Es legible por humanos y fácil de entender.
- Versátil y extensible, lo que permite la representación de datos complejos.

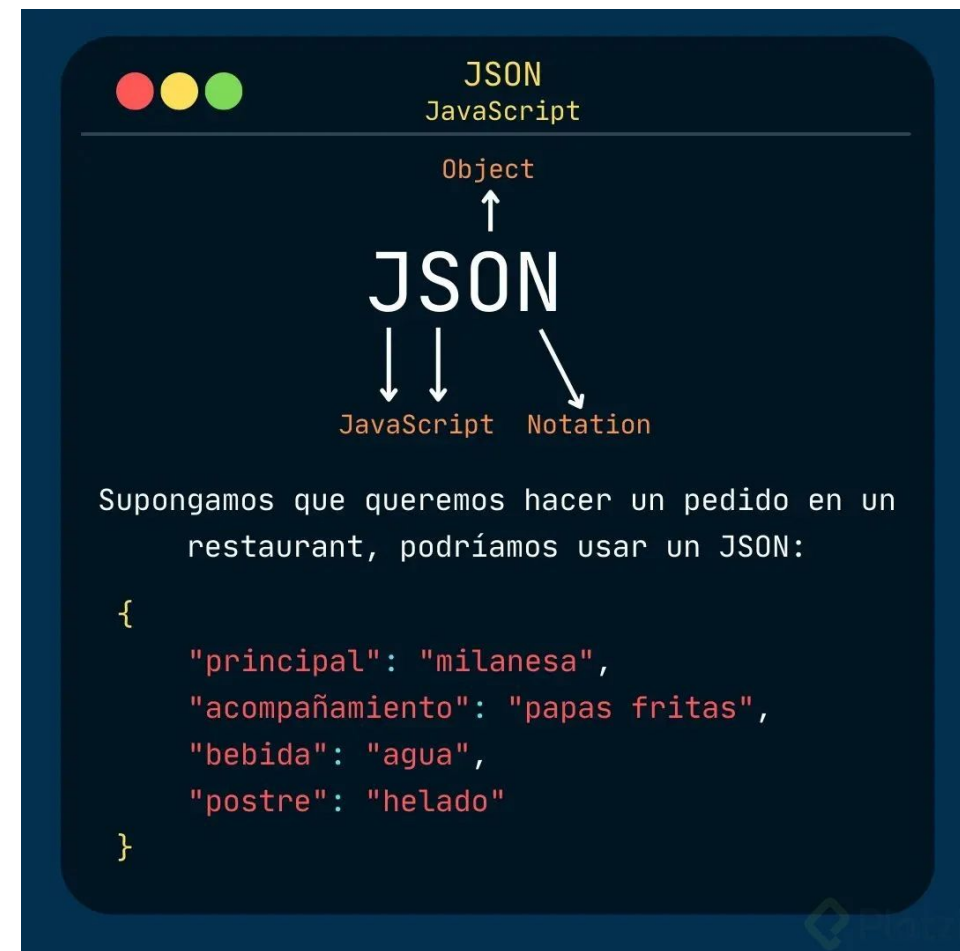




# Protocolos de intercambio de datos

**6- JSON** (Notación de Objetos de JavaScript): es un formato ligero de intercambio de datos que utiliza una sintaxis similar a la de los objetos en JavaScript. Es fácilmente legible por máquinas y humanos.

Ampliamente utilizado en servicios web y aplicaciones que requieren intercambio de datos eficiente.





# Protocolos de intercambio de datos



## 6- JSON Características Principales:

- Utiliza pares clave-valor para representar datos de manera estructurada.
- Es más compacto que XML y fácil de analizar.
- Ampliamente utilizado en servicios web y aplicaciones que requieren intercambio de datos eficiente.
- Ligero y fácil de procesar, especialmente en entornos web.





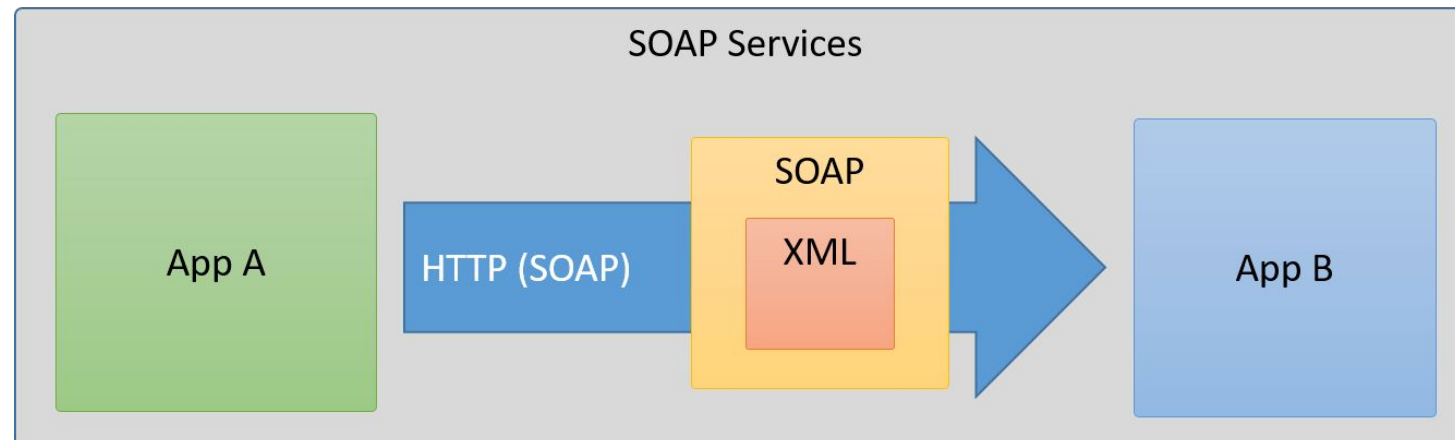
# Protocolos de intercambio de datos



**7- SOAP** (Protocolo Simple de Acceso a Objetos): es un protocolo basado en **XML** utilizado para el intercambio de mensajes estructurados entre sistemas. Se centra en la definición de reglas para la estructura de mensajes y se utiliza comúnmente en entornos donde se requiere una comunicación más formal y controlada.



Ampliamente utilizado en servicios web y aplicaciones empresariales donde se necesita una comunicación más estructurada.





# Protocolos de intercambio de datos



## **SOAP** Características Principales:

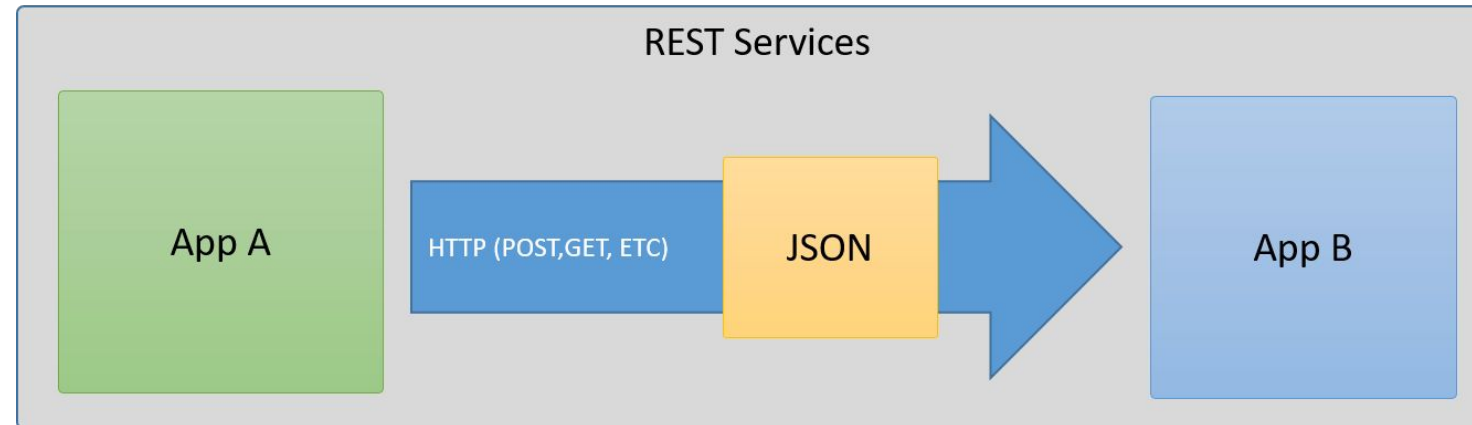
- Define una estructura de mensaje rica en XML con encabezados y cuerpo.
  - Utiliza protocolos de transporte como HTTP y SMTP para la transmisión.
  - Ampliamente utilizado en servicios web y aplicaciones empresariales donde se necesita una comunicación más estructurada.
  - Permite la comunicación entre aplicaciones independientes a través de estándares bien definidos.
- + Proporciona un nivel de seguridad y control adicional.



# Protocolos de intercambio de datos

**8- REST** (Transferencia de Estado Representacional): es un estilo arquitectónico que se basa en la representación de recursos y opera sobre el protocolo **HTTP**. Se centra en la simplicidad y la escalabilidad, utilizando operaciones estándar HTTP para realizar acciones sobre recursos.

Ampliamente utilizado en servicios web y aplicaciones que requieren una comunicación flexible y liviana.





# Protocolos de intercambio de datos



## **REST** Características Principales:

- Utiliza los métodos HTTP estándar (GET, POST, PUT, DELETE) para operaciones CRUD en recursos.
- Es un enfoque sin estado, lo que significa que cada solicitud del cliente contiene toda la información necesaria para entender y procesar la solicitud.
- Favorece la simplicidad y la escalabilidad.
- Permite la construcción de APIs web eficientes y fáciles de consumir.





# Protocolos de intercambio de datos

**Los protocolos de intercambio de datos cumplen varias funciones importantes:**



- Definen un formato estándar para representar los datos, como XML, JSON, CSV, etc. Esto permite que sistemas distintos entiendan la información.
- Establecen cómo debe realizarse la comunicación: solicitud, respuesta, secuenciación, etc. Por ejemplo, el protocolo HTTP define verbos como GET, POST, PUT.
- Permiten la detección y corrección de errores. Los protocolos pueden especificar códigos de error y solicitudes de retransmisión.
- Proporcionan seguridad, como encriptación y autenticación. Por ejemplo, HTTPS es HTTP sobre SSL.
- Facilitan la integración al actuar como un "lenguaje común". Los sistemas solo necesitan "hablar" ese protocolo para entenderse.







# **Ejercicio**

# Preguntas Frecuentes



# Preguntas Frecuentes

## Contexto: 🙌

Vamos a repasar los conceptos más importantes respondiendo algunas preguntas frecuentes sobre protocolos de intercambio de datos.

## Consigna: ✍️ Respondan levantando la mano!

- 1- **HTTP y HTTPS:** ¿Cuál es la diferencia clave entre HTTP y HTTPS?
- 2- **XML y JSON:** ¿Cuáles son las principales diferencias entre XML y JSON en términos de representación de datos?
- 3- **SOAP y REST:** ¿Cómo difieren SOAP y REST en sus enfoques de comunicación?

**Tiempo** 🕒: 10 minutos

○

# ¿Alguna consulta?

+



# RESUMEN

¿Qué logramos en esta clase?

- ✓ **Comprender la importancia de la interoperabilidad entre sistemas**
- ✓ **Conocer los protocolos de intercambio de datos**



# #WorkingTime

Continuemos ejercitando

**¡Antes de cerrar la clase!** Te invitamos a: 📌 📌 📌

1. Repasar nuevamente la grabación de esta clase
2. Revisar el material compartido en la plataforma de Moodle (lo que se vio en clase y algún ejercicio adicional)
  - a. *Lectura Modulo 6, Lección 4: páginas 1 - 3*
3. Traer al próximo encuentro, todas tus dudas y consultas para verlas antes de iniciar nuevo tema.

Momento: ✚

# Time-out!

🕒 5 min.

