

Ejercicios 6 de febrero de 2025

1. Conceptos Básicos

1.1. Definición: explica con tus propias palabras qué es un servicio web y mención al menos dos ejemplos de su uso en la vida cotidiana.

Es una tecnología que utiliza un conjunto de protocolos y estándares que sirven para intercambiar datos entre aplicaciones. **Ejemplos:**

- Mandar un correo.
- Hacer una compra online.

1.2. Diferencias: ¿Cuál es la diferencia entre una API REST y una API SOAP? Proporciona un ejemplo en cada caso.

SOAP es un protocolo de comunicación basado en XML que se utiliza para intercambiar mensajes estructurados en el desarrollo de servicios web. Proporciona un conjunto de reglas para definir el formato de los mensajes y la forma en que deben procesarse. SOAP es altamente extensible y puede utilizarse con una variedad de protocolos de transporte.

REST, que significa Representational State Transfer, es un enfoque arquitectónico para diseñar servicios web basados en la representación de recursos y operaciones sobre esos recursos. En REST, los recursos se identifican mediante URLs, y las operaciones se realizan utilizando métodos HTTP estándar, como GET, POST, PUT y DELETE.

Ejemplos:

- SOAP se usa para servicios bancarios en línea o para integración empresarial.
- REST se usa en aplicaciones de red social y en servicios de pronóstico del tiempo.

1.3. Ventajas: Menciona tres ventajas de usar servicios web en el desarrollo de aplicaciones.

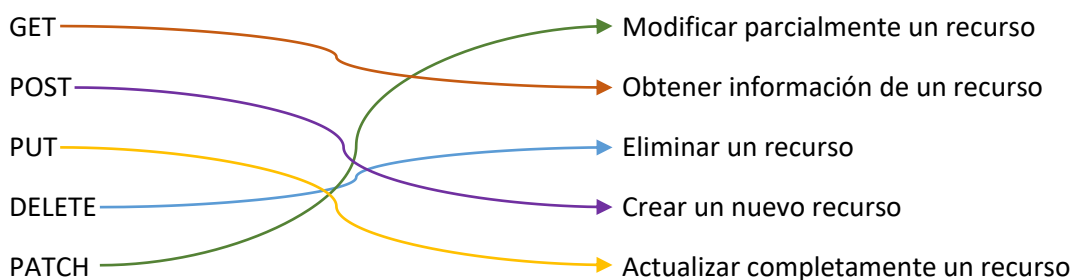
- Puedes crear tus propias herramientas.
- No requiere de equipos especializados.
- Funciona en una gran diversidad de dispositivos.

1.4. Estándares: Enumera y explica brevemente los principales estándares utilizados en los servicios web.

Los estándares web más conocidos y difundidos son: HTML, XML, CSS, ECMAScript, DOM y estándares gráficos como PNG y SVG.

2. Protocolos HTTP y Métodos

2.1. Identificación de Métodos HTTP



2.2. Práctica con HTTP

Observa la siguiente petición HTTP y responde las preguntas:

POST /usuarios HTTP/1.1

Host: api.ejemplo.com

Content-Type: application/json

```
{  
    "nombre": "Juan",  
    "email": "juan@example.com"  
}
```

- **¿Qué acción está realizando esta solicitud?**
Está creando un nuevo usuario en la API.
- **¿Por qué se usa POST en lugar de GET?**
Porque POST envía datos al servidor mientras que GET los recupera, a parte, es mucho más seguro.
- **¿Qué significa Content-Type: application/json?**
Que el contenido de la solicitud tiene datos en formato JSON.
- **¿Cómo respondería el servidor si la solicitud se ha procesado correctamente?**
No podría dar un mensaje indicando que se ha creado correctamente.