

UNIVERSIDAD ALEXANDER VON HUMBOLDT

Actividad:

Consultas API

Alumno:

Juan Esteban Hoyos Rojas

Docente:

Hugo Hernán Henao

Asignatura:

Programación II

III Semestre II-2024

1. Beans (JavaBeans)

Un **JavaBean** es una clase Java que cumple con ciertas convenciones de diseño. Los beans son objetos que encapsulan varios atributos en una única clase y proporcionan métodos de acceso para estos atributos (getters y setters).

Características:

- **Propiedades:** Los atributos son privados y se acceden a través de métodos públicos.
 - **Getters y Setters:** Se usan para obtener y modificar las propiedades.
 - **Constructor sin argumentos:** Debe tener un constructor por defecto (sin parámetros).
 - **Serializable:** Debe implementar la interfaz `Serializable` para permitir que su estado se guarde y se recupere.
-

2. Estructura EAR (Enterprise Archive File)

Un **EAR (Enterprise Archive File)** es un archivo utilizado en aplicaciones empresariales que empaqueta módulos web, módulos EJB (Enterprise JavaBeans), y otros recursos para desplegar en un servidor de aplicaciones Java EE.

Características:

- **Módulos WAR:** Contiene aplicaciones web empaquetadas en archivos `.war`.
 - **Módulos JAR:** Incluye componentes empresariales como EJBs y otros archivos `.jar`.
 - **Estructura jerárquica:** Facilita el empaquetamiento de aplicaciones empresariales complejas en un solo archivo para su despliegue.
 - **Descriptor de despliegue:** Contiene el archivo `application.xml` que describe los módulos incluidos y su configuración.
-

3. JSF (JavaServer Faces)

JSF (JavaServer Faces) es un framework de Java para construir interfaces de usuario basadas en componentes para aplicaciones web. Facilita la creación de aplicaciones web MVC (Modelo-Vista-Controlador).

Características:

- **Componentes reutilizables:** Proporciona una serie de componentes de interfaz de usuario listos para usar.

- **Manejo del ciclo de vida:** Administra el ciclo de vida de una página web desde la solicitud hasta la respuesta.
 - **Integración con Beans:** Se integra fácilmente con Managed Beans, facilitando la interacción entre la vista y la lógica empresarial.
 - **Soporte para plantillas:** Permite crear vistas utilizando tecnologías como XHTML y JSP.
 - **Navegación gestionada:** Soporta la navegación entre páginas web mediante reglas predefinidas.
-

4. SOAP Web Services

SOAP (Simple Object Access Protocol) es un protocolo estándar para el intercambio de mensajes entre aplicaciones sobre HTTP o SMTP. Los **SOAP Web Services** permiten que diferentes aplicaciones se comuniquen entre sí independientemente del sistema operativo o el lenguaje de programación.

Características:

- **Estandarizado:** SOAP sigue un protocolo formal definido por el W3C.
 - **XML:** Usa XML para formatear los mensajes de solicitud y respuesta.
 - **WSDL (Web Services Description Language):** Los servicios web SOAP se describen mediante WSDL.
 - **Extensibilidad:** Permite la extensión del protocolo con características de seguridad y transacciones.
 - **Basado en contratos:** El cliente y el servidor se comunican a través de un contrato definido por el WSDL.
 - **Más rígido:** Comparado con REST, SOAP es más estricto en cuanto a formato y estructura de mensajes.
-

5. WebSocket

WebSocket es un protocolo que permite la comunicación bidireccional en tiempo real entre el cliente (navegador) y el servidor sobre un solo canal TCP.

Características:

- **Comunicación en tiempo real:** Permite la transmisión de mensajes desde y hacia el servidor sin necesidad de múltiples solicitudes HTTP.
- **Conexión persistente:** Mantiene una conexión abierta entre el cliente y el servidor, lo que reduce la sobrecarga de abrir y cerrar conexiones repetidamente.
- **Bajo latencia:** Al ser bidireccional, permite la transferencia de mensajes con baja latencia.

- **Escalabilidad:** Ideal para aplicaciones que requieren comunicaciones en tiempo real como chats, juegos online o actualizaciones de datos en vivo.
 - **Soporte para varios protocolos:** Puede transportar no solo datos JSON, sino también texto y binarios.
-

6. RESTful Web Services

REST (Representational State Transfer) es un estilo arquitectónico que se utiliza para desarrollar servicios web ligeros que se comunican a través de HTTP. Los **RESTful Web Services** son servicios que implementan el estilo REST.

Características:

- **Uso de HTTP:** Aprovecha los verbos HTTP estándar (GET, POST, PUT, DELETE) para realizar operaciones sobre recursos.
- **Recursos:** Los recursos son identificados mediante URIs (Uniform Resource Identifier).
- **Formato de respuesta flexible:** Los datos pueden ser devueltos en varios formatos, como JSON, XML, HTML, etc., siendo JSON el más común.
- **Sin estado (stateless):** Cada solicitud del cliente al servidor debe contener toda la información necesaria para procesarla, ya que el servidor no almacena el estado del cliente entre solicitudes.
- **Escalable y simple:** REST es ligero y sencillo, ideal para aplicaciones web y móviles donde la eficiencia y la velocidad son cruciales.

Repositorio de GIT que evidencia consumo de 3 APIs:

<https://github.com/jhoyosR/consumoAPIJEE.git>