



CICLO: DAW
MÓDULO DE DESARROLLO WEB
ENTORNO SERVIDOR

INTERCAMBIO DE DATOS

Alumno:
Edward-Ionut, Bunoaica
Y1963355C

Los documentos, elementos gráficos, vídeos, transparencias y otros recursos didácticos incluidos en este contenido pueden contener imprecisiones técnicas o errores tipográficos. Periódicamente se realizan cambios en el contenido. Fomento Ocupacional FOC SL puede realizar en cualquier momento, sin previo aviso, mejoras y/o cambios en el contenido.

Es responsabilidad del usuario el cumplimiento de todas las leyes de derechos de autor aplicables. Ningún elemento de este contenido (documentos, elementos gráficos, vídeos, transparencias y otros recursos didácticos asociados), ni parte de este contenido puede ser reproducida, almacenada o introducida en un sistema de recuperación, ni transmitida de ninguna forma ni por ningún medio (ya sea electrónico, mecánico, por fotocopia, grabación o de otra manera), ni con ningún propósito, sin la previa autorización por escrito de Fomento Ocupacional FOC SL.

Este contenido está protegido por la ley de propiedad intelectual e industrial. Pertenecen a Fomento Ocupacional FOC SL los derechos de autor y los demás derechos de propiedad intelectual e industrial sobre este contenido.

Sin perjuicio de los casos en que la ley aplicable prohíbe la exclusión de la responsabilidad por daños, Fomento Ocupacional FOC SL no se responsabiliza en ningún caso de daños indirectos, sean cuales fueren su naturaleza u origen, que se deriven o de otro modo estén relacionados con el uso de este contenido.

© 2022 Fomento Ocupacional FOC SL todos los derechos reservados.

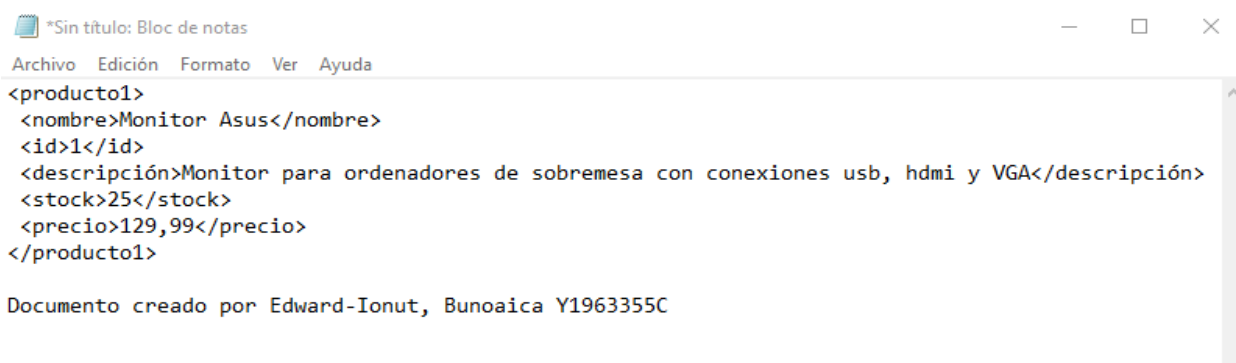
Contenido

1.	EJERCICIOS XML	2
2.	JSON	2
3.	HTTP	3
4.	SOAP	4
5.	REST	5

(Una vez realizado el informe, no olvidar actualizar esta tabla del índice **(F9 + Actualizar toda la tabla)**, con el fin de que se actualicen todos los epígrafes y números de página)

1. EJERCICIOS XML

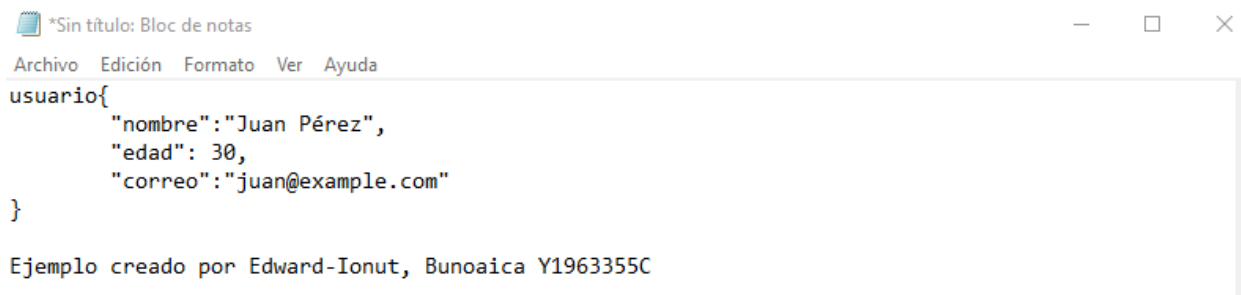
- Explica qué es XML y cuál es su función en intercambio de datos entre servicios
 - En resumidas cuentas, XML es un formato simple basado en texto que sirve para representar la información de manera estructurada
- Cuáles son las principales ventajas y desventajas de XML frente a otros formatos de intercambios de datos
 - Ventajas: Es fácil de procesar por los humanos, separa la información y el contenido de manera estructurada, diseñado para ser usado en cualquier lenguaje y alfabeto, tiene una estructura jerárquica
 - Desventajas: Se puede dar el caso de que la versión que usemos de XML esté desactualizada. Esto puede conllevar problemas, XML se enfoca en diversificarlo en todos los lenguajes posibles en vez de hacer algo universal, al manejar gran cantidad de datos puede ser bastante complicado de procesar.
- Escribe un ejemplo simple de documentos de XML que represente información de un producto



2. JSON

- ¿Qué es JSON y por qué es ampliamente utilizado en intercambios de datos entre servicios web?
 - JSON es un tipo de formato basado en texto igual que XML solo que este formato es legible tanto por humanos como por máquinas. JSON se usa mucho en servicios web porque los datos que maneja son muy ligeros y muy rápidos de procesar
- Compara JSON con XML en términos de legibilidad, eficiencia y compatibilidad
 - En términos de legibilidad XML representa la información en estructura de árbol, mientras que JSON lo recrea como pares clave-valor y lo representa como una especie de mapa
 - En términos de eficiencia podríamos decir que XML es mucho más flexible ya que admite datos de tipo complejo como los datos binarios y de tiempo, en caso de tener un rango más limitado como por ejemplo cadenas, números y objetos.
 - En el caso de compatibilidad los dos están muy iguales, ya que tanto XML y JSON son los dos ampliamente compatibles con la mayoría de lenguajes de programación usados hoy en día

- Convierte el fragmento de XML en JSON



*Sin título: Bloc de notas

Archivo Edición Formato Ver Ayuda

```
usuario{
  "nombre": "Juan Pérez",
  "edad": 30,
  "correo": "juan@example.com"
}
```

Ejemplo creado por Edward-Ionut, Bunoaica Y1963355C

3. HTTP

- Explica el funcionamiento del protocolo HTTP y su papel en la comunicación entre servicios web
 - El protocolo HTTP es un conjunto de reglas de comunicación para el proceso tipo cliente-servidor, el papel del protocolo HTTP es la de notificar la petición del servidor para que responda con otra petición HTTP para desplegar la página web deseada
- ¿Cuáles son los métodos HTTP más comunes y cuál es su propósito en el contexto de la API REST?
 - Los métodos más comunes de HTTP son POST, GET, PUT y DELETE. La API REST se refiere a un conjunto de atributos que comparte con HTTP para integrar aplicaciones y conexiones en arquitecturas de micro servicios. HTTP es un protocolo que usa un sistema "RESTful".
- ¿Qué diferencia hay entre una petición HTTP GET y una POST? Proporciona un ejemplo de cada una
 - Una solicitud GET es un método HTTP que sirve para obtener información sobre una API.

```

1  <html>
2  <head>
3  <title>Ejemplo de PHP</title>
4  </head>
5  <body>
6  <H1>Ejemplo de procesamiento de formularios</H1>
7
8  <FORM ACTION="parametros.php" METHOD="GET">
9  Introduzca su nombre:<INPUT TYPE="text" NAME="nombre"><
10 Introduzca sus apellidos:<INPUT TYPE="text" NAME="apell
11 <INPUT TYPE="submit" VALUE="Enviar">
12 </FORM>
13 </body>
14 </html>
```

- En el caso de una solicitud POST sirve para mandar una solicitud al servidor. En el caso de POST la información no se envía en la url como en la petición GET, es más discreto.

```
$productos = [
    ["nombre" => "Producto 1", "precio" => 10],
    ["nombre" => "Producto 2", "precio" => 15],
    ["nombre" => "Producto 3", "precio" => 20],
];

if ($_SERVER['REQUEST_METHOD'] === 'POST') {
    $producto = $productos[$_POST['producto_index']];
    $_SESSION['carrito'][] = $producto;
}

echo "<h1>Productos:</h1>";
foreach ($productos as $index => $producto) {
    echo "<p>{$producto['nombre']} - Precio: {$producto['precio']}</p>";
    echo "<form method='POST'>
        <input type='hidden' name='producto_index' value='{$index}'>
        <button type='submit'>Añadir al carrito</button>
    </form>";
}
```

4. SOAP

- ¿Qué es SOAP y en qué se diferencia de REST en términos de arquitectura y funcionalidad?
 - Principalmente la diferencia más notable es que SOAP solo admite el intercambio de datos XML, mientras que REST admite XML, JSON, texto plano y HTML. SOAP tiene un problema y es que al manejar solo los XML pueden ser muy grande y esto puede ralentizar las tareas, a REST no le pasa lo mismo ya que admite varios tipos de lenguajes.
- Explica el papel del protocolo de SOAP en los servicios WEB y menciona dos casos donde sería más adecuado que REST
 - El papel de SOAP en los servicios web es el de manejar datos muy delicados, realizar transacciones y compatibilidades con servicios empresariales.
 - En este caso SOAP es más necesario en los sistemas bancarios y en los sistemas de transacción de mensajes seguros.

- Escribe un ejemplo básico de un mensaje SOAP para solicitar información sobre un cliente
 - `<?xml version="1.0" encoding="UTF-8"?>`
 - `<soapenv:Envelope`
`xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"`
 - `xmlns:web="http://www.example.com/webservice">`
 - `<soapenv:Header/>`
 - `<soapenv:Body>`
 - `<web:GetCustomerInfo>`
 - `<web:CustomerID>12345</web:CustomerID>`
 - `</web:GetCustomerInfo>`
 - `</soapenv:Body>`
 - `</soapenv:Envelope>`

5. REST

- ¿Cuáles son los principios fundamentales de una API RESTful? Explica cada uno brevemente
 - En las API RESTful tenemos principalmente 5 principios a seguir.
 - INTERFAZ UNIFORME: esto es fundamental para cualquier API y se trata de interactuar con un servidor de forma totalmente uniforme y estándar.
 - API DE DESCANSO SIN ESTADO: esto significa que el servidor nunca almacene nada relacionado con cualquier tipo de sesión del cliente.
 - ALMACENABLE EN CACHE: cada respuesta enviada del servidor debe contener su propia información sobre la capacidad de almacenamiento en caché
 - PATRÓN CLIENTE-SERVIDOR: las API RESTful imponen el patrón cliente-servidor, esto refuerza la separación de intereses y que el cliente y servidor funcionen de forma independiente
 - SISTEMA EN CAPAS: las API RESTful quieren que la arquitectura que se utilice sea siempre una por capas jerárquicas, esto hace que cada capa solo tenga información sobre sí misma y nada más.
- ¿Qué significa que una API RESTful sea stateless? ¿Cuáles son sus beneficios?
 - El significado de que sea stateless es que cada solicitud es completamente autónoma. Esto hace que sea más escalable, fiable y sea más simple.
- Proporciona un ejemplo de una URL de API RESTful bien diseñada para obtener información sobre un usuario por su ID
 - Un muy buen ejemplo de esto es un servicio meteorológico en línea llamado OpenWeatherMap, este servicio usa una API RESTful, este servicio recibe solicitudes con parámetros de ubicación y devuelve los datos meteorológicos de la zona.

6. Lenguaje de definición de servicios: El estándar WSDL

- Explica que es WSDL y para qué se utiliza en los servicios web basados en SOAP
 - WSDL es un lenguaje basado en XML utilizado para describir los servicios web en la arquitectura SOAP. WSDL es fundamental porque describe como se invoca el servicio, como opera en el interior y define las reglas de comunicación principalmente.
- ¿Cuáles son las principales secciones de un WSDL y qué información contienen?
 - Las principales secciones de un archivo WSDL son <types>define los datos complejos, <message>define los mensajes, <portType>describe las operaciones disponibles en el servicio web, <binding>especifica como se transmiten los datos, <service>define la ubicación web y los puertos por los que se pueden acceder.
- Escribe un fragmento básico de WSDL para describir un servicio web que proporciona información meteorológica
 - En este ejemplo nos aparecen las etiquetas principales del WSDL mencionadas en el ejercicio anterior más algunas otras como <operation>, <input> y <output> que sirven para crear una operación, recibir datos y mostrar luego los resultados.