

Uso de JSON, PHP y Manejo de Datos

1. Introducción a JSON

JSON (JavaScript Object Notation) es un formato ligero para el intercambio de datos. Es fácil de leer y escribir para los humanos, y fácil de analizar y generar para las máquinas. Se utiliza ampliamente en aplicaciones web para enviar y recibir datos entre un cliente (navegador) y un servidor.

Ejemplo de un JSON simple:

```
{
  "nombre": "Juan",
  "edad": 25,
  "ciudad": "Medellín"
}
```

2. JSON en PHP

PHP tiene funciones integradas para trabajar con JSON, como `json_encode` y `json_decode`. Estas permiten convertir datos de un array a JSON y viceversa.

Ejemplo: Convertir un array de PHP a JSON.

```
<?php
$datos = ['nombre' => 'Juan', 'edad' => 25, 'ciudad' => 'Medellín'];
echo json_encode($datos);
?>
```

3. Conexión con Base de Datos y JSON

Para generar datos dinámicos en JSON, primero necesitas conectarte a una base de datos. Luego, puedes extraer los datos y convertirlos a formato JSON.

Ejemplo práctico:

```
<?php
$conn = new mysqli('localhost', 'usuario', 'contraseña', 'base_datos');
if ($conn->connect_error) die('Error de conexión: ' . $conn->connect_error);
$sql = 'SELECT * FROM tabla';
$result = $conn->query($sql);
$datos = [];
while ($row = $result->fetch_assoc()) {
    $datos[] = $row;
}
echo json_encode($datos);
```

```
$conn->close();  
?>
```

4. Consumir JSON desde HTML y JavaScript

Una vez que el backend genera un JSON, el frontend puede consumirlo utilizando JavaScript. Esto se logra con la función `fetch`.

Ejemplo práctico:

```
<!DOCTYPE html>  
<html lang='en'>  
<head>  
  <meta charset='UTF-8'>  
  <title>Consumir JSON</title>  
</head>  
<body>  
  <h1>Datos desde el Servidor</h1>  
  <table border='1'>  
    <thead>  
      <tr><th>ID</th><th>Nombre</th><th>Edad</th></tr>  
    </thead>  
    <tbody id='tabla-datos'></tbody>  
  </table>  
  <script>  
    fetch('datos.php')  
    .then(response => response.json())  
    .then(data => {  
      const tabla = document.getElementById('tabla-datos');  
      data.forEach(fila => {  
        const tr = document.createElement('tr');  
        tr.innerHTML =  
        `<td>${fila.id}</td><td>${fila.nombre}</td><td>${fila.edad}</td>`;  
        tabla.appendChild(tr);  
      });  
    });  
  </script>  
</body>  
</html>
```

5. Ejercicios Prácticos

1. Crea un archivo PHP que genere un JSON con datos estáticos y consúmelo en HTML.
2. Conecta PHP a una base de datos y genera un JSON con los datos de una tabla.
3. Usa JavaScript para mostrar los datos del JSON en una tabla HTML.

6. Desafíos

1. Filtra los datos desde el backend para que solo se incluyan usuarios mayores de 18 años.
2. Agrega un buscador en el frontend que permita filtrar los datos por nombre.
3. Crea un formulario para agregar nuevos registros a la base de datos.

Proyectos de Desarrollo Web ☒

Proyecto 1: Marketplace de Freelancers Técnicos

Contexto General

Desarrollar una plataforma digital que conecte profesionales técnicos con potenciales clientes, facilitando la oferta de servicios digitales mediante un sistema intuitivo y profesional.

Objetivo Principal

Crear un marketplace que permita a freelancers técnicos promocionar sus servicios, y a clientes encontrar profesionales calificados de manera sencilla y eficiente.

Modelo de Datos - Base de Datos

Entidades Principales:

1. Freelancers

- ID (Primary Key)
- Nombre Completo
- Email (Único)
- Especialidad
- Tarifa/Hora
- Portafolio URL
- Imagen Perfil
- Fecha Registro

2. Servicios

- ID Servicio
- ID Freelancer (Foreign Key)
- Nombre Servicio
- Descripción
- Categoría Tecnológica
- Tarifa
- Tiempo Entrega Estimado

3. Solicitudes

- ID Solicitud
- ID Freelancer
- Nombre Cliente
- Email Cliente
- Descripción Requerimiento
- Fecha Solicitud
- Estado (Pendiente/Revisado/Contactado)

Funcionalidades Requeridas

Registro de Freelancers

- Formulario completo de registro
- Validación de campos
- Subida opcional de imagen de perfil
- Generación de credenciales únicas

Catálogo de Servicios

- Listado de servicios
- Filtros por:
 - Categoría tecnológica
 - Rango de precios
 - Tiempo de entrega
- Sistema de calificaciones

Panel de Gestión

- Apartado para freelancers administrar perfiles
- Sección para gestionar servicios
- Visualización de solicitudes recibidas

Requisitos de Diseño

Estilo Visual

- Paleta de Colores:
 - Azul corporativo (#2C3E50)
 - Blanco neutral (#FFFFFF)
 - Gris secundario (#7F8C8D)
- Tipografía:
 - Fuente principal: Roboto
 - Fuente secundaria: Open Sans
- Diseño Responsivo
- Iconografía moderna
- Transiciones suaves

Datos a Mostrar en Interfaz

Perfil Freelancer

- Foto
- Nombre
- Especialidades
- Servicios ofertados
- Calificación promedio
- Tarifa por hora
- Portafolio

Listado Servicios

- Nombre servicio
- Descripción corta
- Categoría
- Precio
- Tiempo entrega
- Botón contacto

Consideraciones Técnicas

- Frontend: HTML5, CSS3
- Backend: PHP
- Base Datos: MySQL
- Formato Intercambio: JSON
- Validaciones: JavaScript

Entregables Esperados

1. Sitio web funcional
2. Base de datos normalizada
3. Interfaz responsive
4. Formularios con validación
5. Sistema básico de contacto

Proyecto 2: Gestor de Inventario para Tienda de Electrónica

(Continuará en el siguiente mensaje por límite de caracteres)

Proyecto 2: Gestor de Inventario para Tienda de Electrónica

Contexto General

Sistema de gestión para una tienda de electrónica que necesita controlar su inventario, ventas y generar reportes básicos. El sistema debe permitir un control eficiente del stock y facilitar el proceso de venta.

Objetivo Principal

Desarrollar una aplicación web que permita gestionar el inventario de productos electrónicos, registrar ventas y mantener un control actualizado del stock disponible.

Modelo de Datos - Base de Datos

Entidades Principales:

1. **Productos**
 - ID Producto (Primary Key)
 - Código SKU
 - Nombre
 - Descripción
 - Categoría
 - Precio Compra
 - Precio Venta
 - Stock Actual
 - Stock Mínimo
 - Imagen Producto
 - Estado (Activo/Agotado)
2. **Categorías**
 - ID Categoría
 - Nombre Categoría
 - Descripción
 - Estado
3. **Ventas**
 - ID Venta
 - Fecha Venta
 - Total Venta
 - Método Pago
 - Estado Venta
4. **Detalle_Ventas**
 - ID Detalle
 - ID Venta (Foreign Key)
 - ID Producto
 - Cantidad
 - Precio Unitario
 - Subtotal

Funcionalidades Requeridas

Gestión de Productos

- Registro de nuevos productos
- Actualización de stock
- Modificación de precios
- Alerta de stock bajo
- Búsqueda por código o nombre

Proceso de Venta

- Carrito de compras
- Selección múltiple de productos
- Cálculo automático de totales
- Registro de venta finalizada
- Actualización automática de stock

Panel de Control

- Dashboard con resumen:
 - Productos más vendidos
 - Stock bajo
 - Ventas del día
 - Total de ingresos

Requisitos de Diseño

Estilo Visual

- Paleta de Colores:
 - Verde principal (#2ECC71)
 - Negro textos (#333333)
 - Gris claro fondos (#F5F5F5)
 - Rojo alertas (#E74C3C)
- Elementos UI:
 - Cards para productos
 - Tablas responsivas
 - Botones destacados
 - Iconos intuitivos

Datos a Mostrar en Interfaz

Listado de Productos

- Imagen miniatura
- Código SKU
- Nombre producto
- Stock disponible
- Precio venta
- Estado
- Acciones (Editar/Eliminar)

Panel de Ventas

- Productos seleccionados
- Cantidades
- Precios unitarios
- Subtotales
- Total general

Consideraciones Técnicas

- Estructura modular
- Validaciones cliente/servidor
- Manejo de sesiones
- Reportes en JSON
- Control de acceso básico

Requerimientos Específicos

1. Interfaz Administrativa
 - Login simple
 - CRUD completo productos
 - Gestión de ventas
 - Reportes básicos
2. Proceso de Venta
 - Interfaz intuitiva
 - Búsqueda rápida
 - Cálculos automáticos
 - Confirmación de venta
3. Reportes Necesarios
 - Ventas diarias
 - Stock actual
 - Productos agotados
 - Productos más vendidos

Entregables Esperados

1. Sistema web funcional
2. Base de datos optimizada
3. Manual de usuario básico
4. Módulo de reportes
5. Interfaz responsiva

Notas Importantes

- Diseño mobile-first
- Optimización de consultas

- Manejo de errores
- Backups automáticos
- Logs de actividades

Recuerda que estas son las funcionalidades básicas solicitadas. Puedes expandir o mejorar características según tu criterio técnico, manteniendo la esencia del proyecto.