**CRUD**

El CRUD es una de las bases fundamentales en el desarrollo de aplicaciones web, ya que permite gestionar datos de manera dinámica en una base de datos. Cada una de sus siglas representa una operación esencial:

**Create** (Crear): Agregar nuevos registros a la base de datos.

**Read** (Leer): Obtener y mostrar datos almacenados.

**Update** (Actualizar): Modificar datos existentes.

**Delete** (Eliminar): Borrar registros de la base de datos.

**Diferentes posibilidades del CRUD**

A medida que avanzas, puedes implementar el CRUD de distintas maneras según las necesidades del proyecto:

**CRUD Básico**

Formularios simples con PHP para manejar envíos de datos.

JavaScript con fetch() para hacer peticiones AJAX sin recargar la página.

* **CRUD** **con Validaciones**

Validar los datos en el backend (PHP) y en el frontend (JavaScript).

Mostrar errores en tiempo real sin necesidad de recargar.

* **CRUD con Autenticación**

Solo usuarios registrados pueden crear, leer, actualizar y eliminar datos.

* **CRUD con Relaciones entre Tablas**

Por ejemplo, en un sistema de productos y categorías, cada producto pertenece a una categoría.

* **CRUD con Paginación y Filtros**

Listar datos con paginación y filtrar por diferentes criterios.

* **CRUD con Archivos**

Subir imágenes o documentos asociados a los registros.

* **CRUD en una SPA (**Single Page Application**)**

Usando JavaScript (fetch y await) para hacer que las interacciones sean dinámicas sin recargar.

**Proyectos donde aplicar CRUD**

Puedes practicar CRUD en diferentes tipos de proyectos. Algunos interesantes para empezar son:

* **Gestor de Tareas (**To-Do List**)**

Los usuarios pueden agregar, editar, marcar como completadas y eliminar tareas.

* **Agenda de Contactos**

Permite almacenar nombres, correos y teléfonos con funciones de búsqueda y edición.

* **Gestión de Productos**

Agregar, listar, actualizar y eliminar productos con stock y categorías.

* **Sistema de Usuarios**

Registro, listado, edición y eliminación de usuarios con control de accesos.

* **Registro de Ventas**

Registrar compras, calcular totales y gestionar productos vendidos.

1. **CREATE:** Para procesar los datos e insertarlos en la base de datos
   1. **Lo primero es crear nuestra conexión a la base de datos y verificación de la conexión. Puede ser en un doc. aparte (A eso le llamamos una buena práctica)**
2. <?php
3. // 1️⃣ Conectar a la base de datos (XAMPP usa localhost, usuario 'root' y sin contraseña por defecto)
4. $servername = "localhost";
5. $username = "root";
6. $password = "";
7. $dbname = "registro\_ventas";
8. $conexion = new mysqli($servername, $username, $password, $dbname);
9. // 2️⃣ Verificar la conexión
10. if ($conexion->connect\_error) {
11. die("Error en la conexion: " . $conexion->connect\_error);
12. }

**1.2. Verificar que el formulario fue enviado por POST**

// 3️⃣ Validar que los datos vienen por POST

if ($\_SERVER["REQUEST\_METHOD"] == "POST")

**1.3. Verifico y limpio los datos**

// 4️⃣ Recibir y limpiar los datos del formulario

    $producto = trim($\_POST["producto"]); // El método trim(): Elimina espacios en blanco al inicio y final del nombre del producto.

    $cantidad = intval($\_POST["cantidad"]); // intval(): Convierte la cantidad en número entero.

    $precio = floatval($\_POST["precio"]); // floatval(): Convierte el precio en número decimal.

**1.4. Verificar que los campos no estén vacíos o tengan valores inválidos**

// 5️⃣ Verificar que los campos no estén vacíos

    if (!empty($producto) && $cantidad > 0 && $precio > 0)

📌 Se valida que:  
✅ producto no esté vacío.  
✅ cantidad sea mayor a 0.  
✅ precio sea mayor a 0.

**1.5. Preparar la consulta SQL**

 // 6️⃣ Preparar la consulta SQL

        $stmt = $conexion->prepare("INSERT INTO ventas (producto, cantidad, precio) VALUES (?, ?, ?)");  // Se insertan los valores con ? (evita inyecciones SQL).

        $stmt->bind\_param("sid", $producto, $cantidad, $precio); // "s"=string, "i"=int, "d"=double

**1. Se prepara la consulta con prepare(), definiendo los marcadores ?:**

$stmt = $conexion->prepare("INSERT INTO ventas (producto, cantidad, precio) VALUES (?, ?, ?)");

**2. Se asignan los valores con bind\_param(), especificando sus tipos de datos:**

$stmt->bind\_param("sid", $producto, $cantidad, $precio);

"s" → producto (string, "Laptop")

"i" → cantidad (integer, 2)

"d" → precio (double/float, 750.99)

**3. Se ejecuta la consulta con execute():**

$stmt->execute();