

Taller de Programación

Clase 15

Cristian Vallejos (cristian.vallejos@uach.cl)

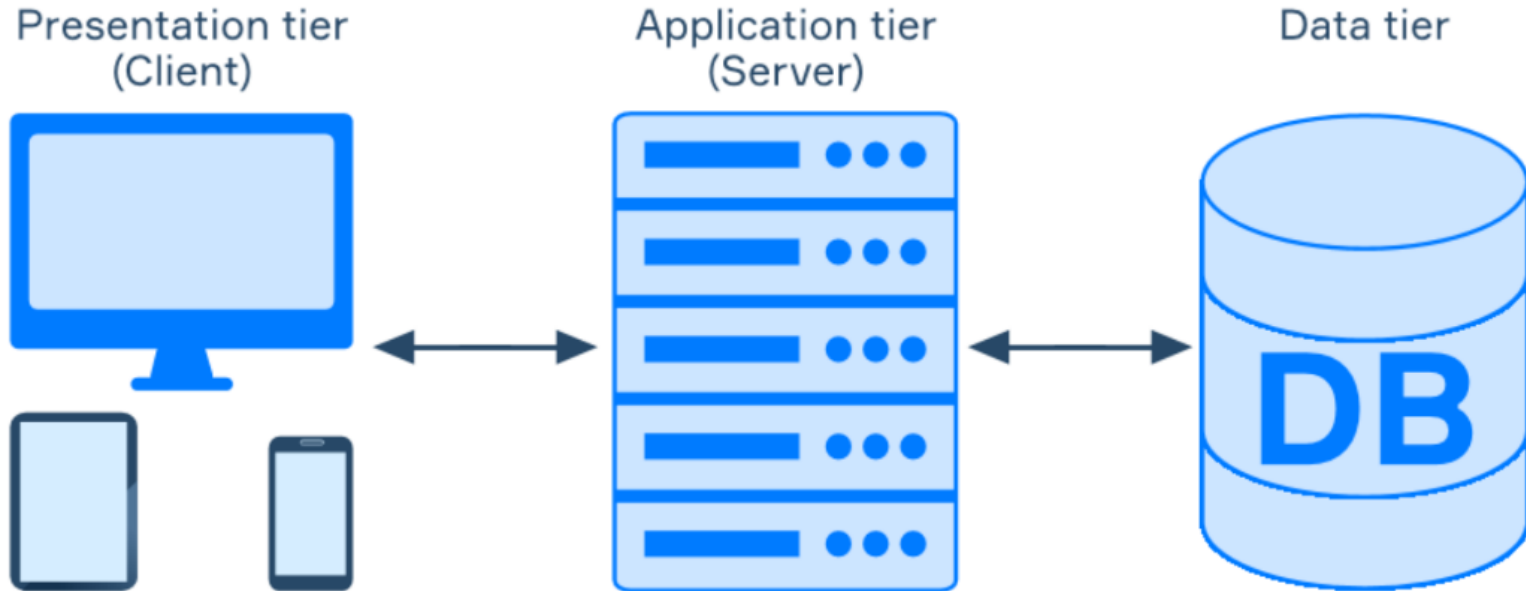
Joel Torres (joel.torres@uach.cl)

```
<meta http-equiv="X-UA-Compatible"
<meta name="viewport" content="width
<title>HTML</title>
```

Temario

- Backend con PHP
- Conexión a Bases de Datos
- Implementación a CRUD

Arquitectura de tres niveles



Backend con PHP

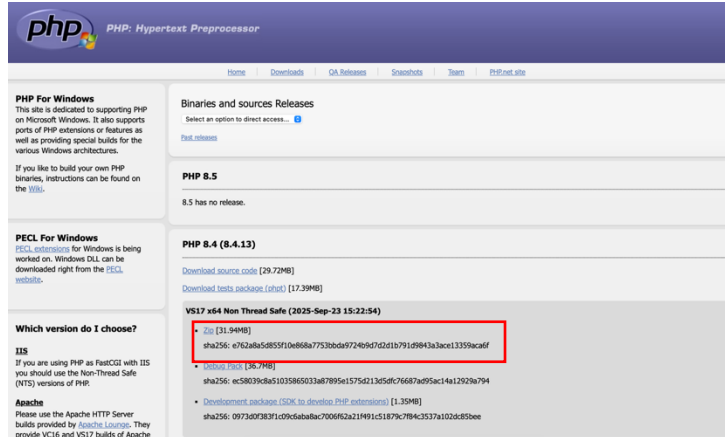
- **PHP** (*Hypertext Preprocessor*) es un lenguaje de programación **de código abierto** orientado al desarrollo de aplicaciones web. Fue creado por **Rasmus Lerdorf en 1994**, y es especialmente popular para el desarrollo **del lado del servidor (backend)**.
- PHP permite crear sitios **dinámicos**, es decir, páginas que **interactúan con bases de datos, reciben información de usuarios y generan contenido personalizado** en tiempo real.



Backend con PHP (Opción 1)

► Instalación Servidor WEB

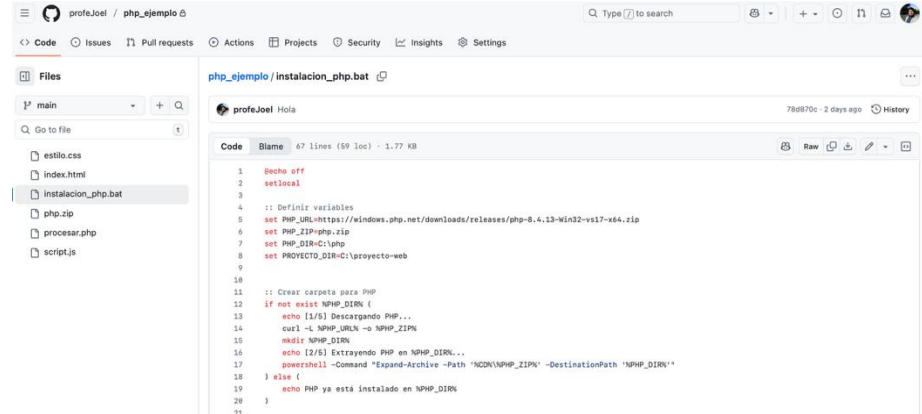
<https://windows.php.net/download>



The screenshot shows the official PHP website for Windows downloads. It features the PHP logo and navigation links like Home, Downloads, QA Releases, Snapshots, Team, and Feedback. The main content area is titled 'PHP For Windows' and 'Binaries and sources Releases'. It lists various PHP versions for Windows, including PHP 8.5, PHP 8.4 (8.4.13), and VS17 x64 Non Thread Safe (2025-Sep-23 15:22:54). A red box highlights the 'Zip' download option for the VS17 x64 Non Thread Safe version, showing its size (31.94MB) and SHA256 hash.

► Instalación por batch

https://github.com/profeJoel/php_ejemplo/blob/main/instalacion_php.bat



The screenshot shows a GitHub repository named 'profeJoel / php_ejemplo'. The file 'instalacion_php.bat' is selected, displaying its contents. The batch file is a script for installing PHP on Windows. It starts with 'echo off' and 'setlocal'. It defines variables for the PHP URL, ZIP file, directory, and project directory. It then uses 'curl' to download the PHP ZIP file and 'mkdir' to create the directory. Finally, it uses 'powershell' to extract the ZIP file to the specified directory. The script ends with 'echo PHP ya está instalado en NPHP_DIRN'.

Backend con PHP (Opción 2)

► Instalación XAMPP

<https://www.apachefriends.org>

► Panel de Control XAMPP



The screenshot shows the XAMPP website homepage. At the top is a navigation bar with links: Apache Friends, Download, Hosting, Community, About, and a search bar. Below the navigation bar is the XAMPP logo and the text "XAMPP Apache + MariaDB + PHP + Perl". A section titled "What is XAMPP?" describes it as the most popular PHP development environment. At the bottom, there are download buttons for Windows, Linux, and OS X, all for version 8.2.12 (PHP 8.2.12).

Apache Friends Download Hosting Community About Search.. Search EN

XAMPP Apache + MariaDB + PHP + Perl

What is XAMPP?

XAMPP is the most popular PHP development environment

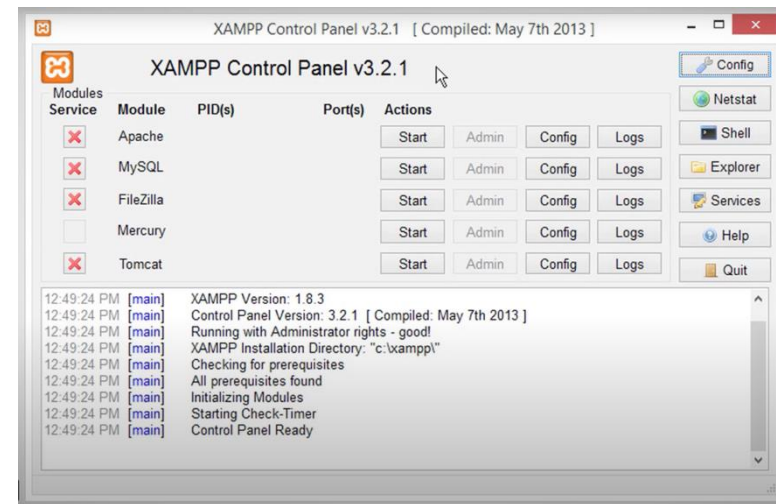
XAMPP is a completely free, easy to install Apache distribution containing MariaDB, PHP, and Perl. The XAMPP open source package has been set up to be incredibly easy to install and to use.

Download
Click here for other versions

XAMPP for Windows
8.2.12 (PHP 8.2.12)

XAMPP for Linux
8.2.12 (PHP 8.2.12)

XAMPP for OS X
8.2.4 (PHP 8.2.4)



The screenshot shows the XAMPP Control Panel v3.2.1 window. It has a title bar "XAMPP Control Panel v3.2.1 [Compiled: May 7th 2013]". The main area is divided into two sections. The top section is a table with columns: Modules, Service, Module, PID(s), Port(s), and Actions. The bottom section is a log window showing the status of the XAMPP installation.

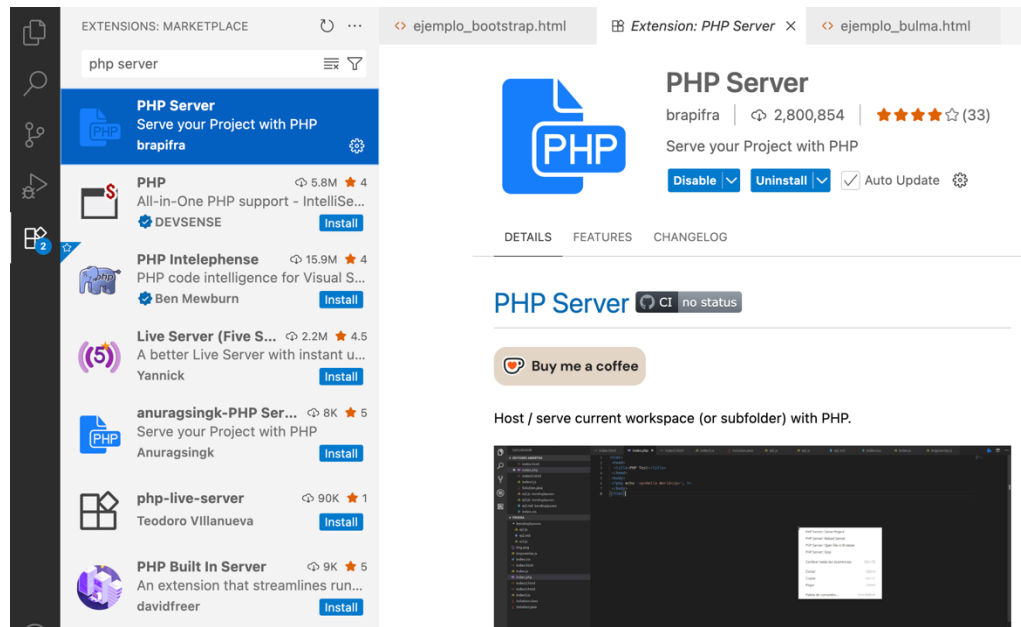
Modules	Service	Module	PID(s)	Port(s)	Actions
<input checked="" type="checkbox"/>	Apache	Apache			Start Admin Config Logs
<input checked="" type="checkbox"/>	MySQL	MySQL			Start Admin Config Logs
<input checked="" type="checkbox"/>	FileZilla	FileZilla			Start Admin Config Logs
<input type="checkbox"/>	Mercury	Mercury			Start Admin Config Logs
<input checked="" type="checkbox"/>	Tomcat	Tomcat			Start Admin Config Logs

Log window content:

```
12:49:24 PM [main] XAMPP Version: 1.8.3
12:49:24 PM [main] Control Panel Version: 3.2.1 [ Compiled: May 7th 2013 ]
12:49:24 PM [main] Running with Administrator rights - good!
12:49:24 PM [main] XAMPP Installation Directory: "c:\xampp\"
12:49:24 PM [main] Checking for prerequisites
12:49:24 PM [main] All prerequisites found
12:49:24 PM [main] Initializing Modules
12:49:24 PM [main] Starting Check-Timer
12:49:24 PM [main] Control Panel Ready
```

Backend con PHP

- Configuración Visual Studio Code
- Buscar en Extensiones
- Indicar PHP Server
- Instalar extensión PHP Server




► Ejemplo

https://github.com/profeJoel/php_ejemplo/blob/main/index.html

```
<> index.html >  html
1  <!DOCTYPE html>
2  <html lang="es">
3  <head>
4    <meta charset="UTF-8">
5    <title>Formulario de Contacto</title>
6    <link rel="stylesheet" href="estilo.css">
7  </head>
8  <body>
9    <h1>Formulario de Contacto</h1>
10   <form action="procesar.php" method="POST">
11     <label>Nombre: <input type="text" name="nombre" required></label><br>
12     <label>Email: <input type="email" name="email" required></label><br>
13     <label>Mensaje: <textarea name="mensaje" required></textarea></label><br>
14     <button type="submit">Enviar</button>
15   </form>
16
17   <script src="script.js"></script>
18 </body>
19 </html>
```


► Ejemplo

https://github.com/profeJoel/php_ejemplo/blob/main/estilo.css

```
# estilo.css >  form
1  body {
2      font-family: Arial, sans-serif;
3      padding: 20px;
4      background-color:  #f2f2f2;
5  }
6
7  form {
8      background-color:  #fff;
9      padding: 15px;
10     border-radius: 8px;
11     box-shadow: 0 0 10px  #ccc;
12     max-width: 400px;
13 }
```

► **Ejemplo** https://github.com/profeJoel/php_ejemplo/blob/main/script.js

JS script.js > ...

```
1 document.querySelector('form').addEventListener('submit', function () {  
2   console.log("Formulario enviado");  
3 });
```

► Ejemplo

https://github.com/profeJoel/php_ejemplo/blob/main/procesar.php

 procesar.php

```
1  <?php
2  if ($_SERVER["REQUEST_METHOD"] === "POST") {
3      $nombre = htmlspecialchars($_POST["nombre"]);
4      $email = htmlspecialchars($_POST["email"]);
5      $mensaje = htmlspecialchars($_POST["mensaje"]);
6
7      echo "<h1>Gracias por tu mensaje</h1>";
8      echo "<p><strong>Nombre:</strong> $nombre</p>";
9      echo "<p><strong>Email:</strong> $email</p>";
10     echo "<p><strong>Mensaje:</strong> $mensaje</p>";
11 } else {
12     echo "<p>Error: Debes enviar el formulario correctamente.</p>";
13 }
14 ?>
```

► Esquema:

db.sql

```
1  -- Crear base de datos
2  CREATE DATABASE IF NOT EXISTS formulario_contacto CHARACTER SET utf8mb4 COLLATE utf8mb4_general_ci;
3
4  USE formulario_contacto;
5
6  -- Crear tabla para almacenar los mensajes
7  CREATE TABLE mensajes (
8      id_mensaje INT AUTO_INCREMENT PRIMARY KEY,
9      nombre VARCHAR(100) NOT NULL,
10     email VARCHAR(100) NOT NULL,
11     mensaje TEXT NOT NULL,
12     fecha_envio TIMESTAMP DEFAULT CURRENT_TIMESTAMP
13 );
```

Conexión a Base de Datos

- Se debe especificar:
 - Host corresponde al servidor de Base de datos
 - User corresponde al usuario con acceso a la DBMS
 - Pass corresponde a la contraseña de dicho usuario.
 - Db corresponde al nombre de la base de datos que se requiere conectar

🐘 db.php

```
1  <?php
2  $host = "localhost";
3  $user = "root";
4  $pass = "";
5  $dbname = "formulario_contacto";
6
7  $conn = new mysqli($host, $user, $pass, $dbname);
8
9  if ($conn->connect_error) {
10     die("Error de conexión: " . $conn->connect_error);
11 }
12 ?>
```

Backend PHP + DB



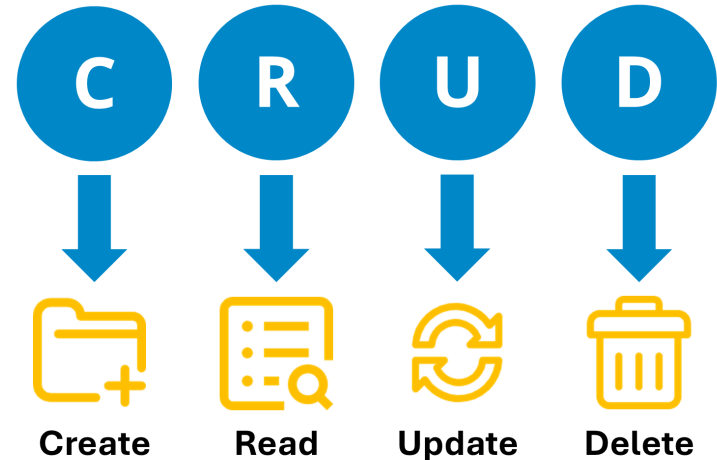
Backend PHP + DB

- Se modifica el código para guardar los datos.
 - Se seleccionan los datos a guardar
 - Se asocian a una variable
 - Se indica la consulta a utilizar.
 - Se realiza la función **execute()**

```
guardar.php
1  <?php
2  include 'db.php'; // Conexión a la base de datos
3
4  if ($_SERVER["REQUEST_METHOD"] === "POST") {
5      // Sanitizar los datos recibidos
6      $nombre = htmlspecialchars($_POST["nombre"]);
7      $email = htmlspecialchars($_POST["email"]);
8      $mensaje = htmlspecialchars($_POST["mensaje"]);
9
10     // Preparar la consulta para evitar inyección SQL
11     $stmt = $conn->prepare("INSERT INTO mensajes (nombre, email, mensaje) VALUES (?, ?, ?)");
12     $stmt->bind_param("sss", $nombre, $email, $mensaje);
13
14     if ($stmt->execute()) {
15         echo "<h1>Gracias por tu mensaje</h1>";
16         echo "<p><strong>Nombre:</strong> $nombre</p>";
17         echo "<p><strong>Email:</strong> $email</p>";
18         echo "<p><strong>Mensaje:</strong> $mensaje</p>";
19         echo "<p><em>Tu mensaje ha sido guardado correctamente en la base de datos.</em></p>";
20     } else {
21         echo "<p>Error al guardar el mensaje: " . $stmt->error . "</p>";
22     }
23
24     $stmt->close();
25     $conn->close();
26 } else {
27     echo "<p>Error: Debes enviar el formulario correctamente.</p>";
28 }
29 ?>
```

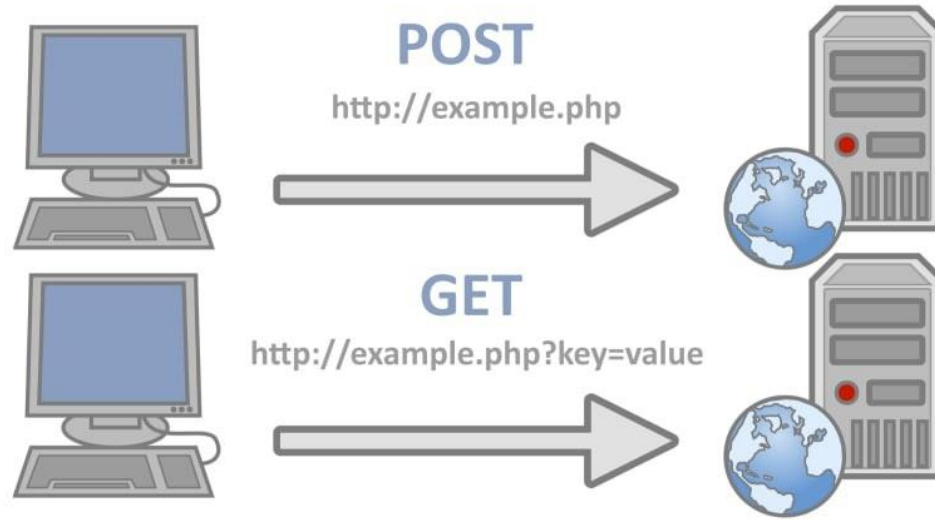
Implementación a CRUD

- ¿Qué es CRUD?
- **CRUD** es un acrónimo que representa las **cuatro operaciones básicas** que se pueden realizar sobre datos en una base de datos o sistema de almacenamiento.
- Estas operaciones permiten construir aplicaciones que interactúan con información almacenada, como sistemas web, móviles o de escritorio.



Verbos HTTP

- Los verbos HTTP son protocolos que lanzan diferentes acciones.



Implementación a CRUD

► Ejemplo DB: Clínica Dental

```
1      -- Creación de base de datos
2  •   CREATE DATABASE IF NOT EXISTS ClinicaDental;
3  •   USE ClinicaDental;
4
5      -- Tabla de pacientes
6  •   CREATE TABLE Paciente (
7          id_paciente INT PRIMARY KEY AUTO_INCREMENT,
8          nombre VARCHAR(100) NOT NULL,
9          rut VARCHAR(12) UNIQUE,
10         telefono VARCHAR(20),
11         email VARCHAR(100)
12     );
```

Implementación a CRUD

► Ejemplo Conexión DB:

crud_paciente >  db.php

```
1  <?php
2  $host = "localhost";
3  $user = "root";
4  $pass = "q1w2e3r4.";
5  $db = "ClinicaDental";
6
7  $conn = new mysqli($host, $user, $pass, $db);
8  if ($conn->connect_error) {
9      |    die("Conexión fallida: " . $conn->connect_error);
10 }
11 ?>
```

Implementación a CRUD

► Create = Insertar:

crud_paciente > insertar.php

```
1  <?php include 'db.php'; include 'header.php';
2  if ($_POST) {
3      $stmt = $conn->prepare("INSERT INTO Paciente (nombre, rut, telefono, email) VALUES (?, ?, ?, ?)");
4      $stmt->bind_param("ssss", $_POST['nombre'], $_POST['rut'], $_POST['telefono'], $_POST['email']);
5      $stmt->execute();
6      header("Location: index.php");
7  }
8  ?>
9  <h2>Nuevo Paciente</h2>
10 <form method="post">
11     <div class="mb-3">
12         <label>Nombre</label>
13         <input type="text" name="nombre" class="form-control" required>
14     </div>
15     <div class="mb-3">
16         <label>RUT</label>
17         <input type="text" name="rut" class="form-control">
18     </div>
19     <div class="mb-3">
20         <label>Teléfono</label>
21         <input type="text" name="telefono" class="form-control">
22     </div>
23     <div class="mb-3">
24         <label>Email</label>
25         <input type="email" name="email" class="form-control">
26     </div>
27     <button class="btn btn-primary">Guardar</button>
28 </form>
29 <?php include 'footer.php'; ?>
```

Implementación a CRUD

► Read = Recuperar información:

```
crud_paciente > index.php
1  <?php include 'db.php'; include 'header.php';
2  $result = $conn->query("SELECT * FROM Paciente ORDER BY id_paciente DESC");
3  ?>
4  <h2>Lista de Pacientes</h2>
5  <table class="table table-bordered">
6      <thead>
7          <tr>
8              <th>Nombre</th>
9              <th>RUT</th>
10             <th>Teléfono</th>
11             <th>Email</th>
12             <th>Acciones</th>
13         </tr>
14     </thead>
15     <tbody>
16     <?php while ($row = $result->fetch_assoc()): ?>
17         <tr>
18             <td><?= $row['nombre'] ?></td>
19             <td><?= $row['rut'] ?></td>
20             <td><?= $row['telefono'] ?></td>
21             <td><?= $row['email'] ?></td>
22             <td>
23                 <a href="detalle.php?id=<?= $row['id_paciente'] ?>" class="btn btn-info btn-sm">Ver</a>
24                 <a href="editar.php?id=<?= $row['id_paciente'] ?>" class="btn btn-warning btn-sm">Editar</a>
25                 <a href="eliminar.php?id=<?= $row['id_paciente'] ?>" class="btn btn-danger btn-sm" onclick="return confirm
26                     ('¿Seguro de eliminar?')">Eliminar</a>
27             </td>
28         </tr>
29     </tbody>
30 </table>
31 <?php include 'footer.php'; ?>
```

Implementación a CRUD

► Read = Recuperar Información en detalle:

crud_paciente > detalle.php

```
1  <?php include 'db.php'; include 'header.php';
2  $id = $_GET['id'];
3  $stmt = $conn->prepare("SELECT * FROM Paciente WHERE id_paciente = ?");
4  $stmt->bind_param("i", $id);
5  $stmt->execute();
6  $result = $stmt->get_result();
7  $p = $result->fetch_assoc();
8  ?>
9  <h2>Detalle del Paciente</h2>
10 <ul class="list-group">
11     <li class="list-group-item"><strong>Nombre:</strong> <?= $p['nombre'] ?></li>
12     <li class="list-group-item"><strong>RUT:</strong> <?= $p['rut'] ?></li>
13     <li class="list-group-item"><strong>Teléfono:</strong> <?= $p['telefono'] ?></li>
14     <li class="list-group-item"><strong>Email:</strong> <?= $p['email'] ?></li>
15 </ul>
16 <a href="index.php" class="btn btn-secondary mt-3">Volver</a>
17 <?php include 'footer.php'; ?>
```

Implementación a CRUD

► Update = Recuperar Información en detalle y Actualizar:

```
crud_paciente > editar.php
1  <?php include 'db.php'; include 'header.php';
2  $id = $_GET['id'];
3  $stmt = $conn->prepare("SELECT * FROM Paciente WHERE id_paciente = ?");
4  $stmt->bind_param("i", $id);
5  $stmt->execute();
6  $result = $stmt->get_result();
7  $p = $result->fetch_assoc();
8  if ($_POST) {
9      $stmt = $conn->prepare("UPDATE Paciente SET nombre=?, rut=?, telefono=?, email=? WHERE id_paciente=?");
10     $stmt->bind_param("ssssi", $_POST['nombre'], $_POST['rut'], $_POST['telefono'], $_POST['email'], $id);
11     $stmt->execute();
12     header("Location: index.php");
13 }
14 ?>
15 <h2>Editar Paciente</h2>
16 <form method="post">
17     <div class="mb-3">
18         <label>Nombre</label>
19         <input type="text" name="nombre" class="form-control" value="<?= $p['nombre'] ?>" required>
20     </div>
21     <div class="mb-3">
22         <label>RUT</label>
23         <input type="text" name="rut" class="form-control" value="<?= $p['rut'] ?>">
24     </div>
25     <div class="mb-3">
26         <label>Teléfono</label>
27         <input type="text" name="telefono" class="form-control" value="<?= $p['telefono'] ?>">
28     </div>
29     <div class="mb-3">
30         <label>Email</label>
31         <input type="email" name="email" class="form-control" value="<?= $p['email'] ?>">
32     </div>
33     <button class="btn btn-primary">Actualizar</button>
34 </form>
35 <?php include 'footer.php'; ?>
```

Implementación a CRUD

► Delete = Eliminar registro:

crud_paciente >  eliminar.php

```
1  <?php include 'db.php';
2  $id = $_GET['id'];
3  $stmt = $conn->prepare("DELETE FROM Paciente WHERE id_paciente = ?");
4  $stmt->bind_param("i", $id);
5  $stmt->execute();
6  header("Location: index.php");
```


- La **Municipalidad de Puerto Varas** desea implementar un sistema web simple que permita a la ciudadanía enviar consultas, sugerencias o reclamos a través de un formulario en línea. Esta herramienta tiene como objetivo mejorar la participación ciudadana y organizar las solicitudes recibidas por la unidad de atención al público.
- El sistema debe permitir:
 - **Registrar nuevos mensajes** enviados por la ciudadanía, incluyendo los siguientes datos:
 - Nombre completo del remitente
 - Correo electrónico
 - Tipo de mensaje (Consulta, Sugerencia, Reclamo)
 - Contenido del mensaje
 - **Listar todos los mensajes recibidos**, ordenados desde el más reciente al más antiguo.
 - **Ver en detalle cada mensaje**, con toda la información incluida en el registro.
 - **Editar un mensaje**, por ejemplo si el remitente comete un error al enviar su nombre o el tipo de mensaje.
 - **Eliminar mensajes**, en caso de spam o registros inválidos.

Taller de Programación

Clase 15

Cristian Vallejos (cristian.vallejos@uach.cl)

Joel Torres (joel.torres@uach.cl)

```
<meta http-equiv="X-UA-Compatible"
<meta name="viewport" content="width
<title>HTML</title>
```