

UNIVERSIDAD LUTERANA SALVADOREÑA



TITULO O TEMA: Formulario de roles para el personal de un
Hospital

NOMBRE DE LOS ESTUDIANTES:

Salvador Josué Cisneros Hernández - CH01137269

Héctor Alexis Gonzales Vásquez - GV01137328

Sebastián Emilio Vásquez Mejía - VM01137362

Miguel Ángel Nolasco Osorio – NO01137273

ASIGNATURA:

Base de Datos

DOCENTE:

Pedro Noble

CICLO/AÑO: I - 2023

Link del video: <https://youtu.be/skk-VOqUrDI>

INDICE

INTRODUCCIÓN.....	3
OBJETIVO GENERAL	3
OBJETIVOS ESPECÍFICOS	4
JUSTIFICACIÓN.....	4
PLANTEAMIENTO DEL PROBLEMA	4
TECNOLOGÍAS UTILIZADAS.....	5
DIAGRAMAS	8
CONCLUSIONES	9
ANEXOS	10

INTRODUCCIÓN

El presente proyecto tiene como objetivo desarrollar un sistema de gestión de roles del personal hospitalario que permita optimizar la organización de tareas y mejorar la atención a los pacientes en un centro hospitalario. La implementación de este sistema proporcionará una estructura eficiente para asignar y distribuir a los miembros del personal en diferentes áreas de trabajo de acuerdo con sus habilidades y responsabilidades.

El sistema de gestión de roles será una herramienta fundamental para garantizar que cada miembro del equipo médico y administrativo cumpla con las funciones asignadas de manera efectiva y en línea con las necesidades del hospital. Esto permitirá una mejor coordinación entre los diferentes departamentos y una distribución equitativa de las responsabilidades, lo que resultará en una atención más eficiente y una mejora general de los servicios hospitalarios.

Para la creación de este sistema, se utilizará el editor de código Visual Studio Code, reconocido por su versatilidad y amplias funcionalidades de desarrollo. Este entorno de programación brindará un marco adecuado para la implementación del programa, facilitando la creación de las funcionalidades requeridas y permitiendo una gestión eficiente del código fuente.

El sistema de gestión de roles del personal hospitalario buscará optimizar la asignación de tareas, asegurando que cada miembro del equipo desempeñe un papel adecuado y acorde con sus capacidades y conocimientos. Además, se enfocará en mejorar la comunicación interna y la colaboración entre los diferentes roles, fomentando un ambiente de trabajo cohesionado y eficaz.

En resumen, este proyecto tiene como finalidad desarrollar un sistema de gestión de roles del personal hospitalario que brinde una estructura organizativa sólida, utilizando Visual Studio Code como herramienta principal de desarrollo. Se espera que esta solución contribuya a la mejora de la eficiencia y calidad de los servicios hospitalarios, beneficiando tanto al personal como a los pacientes atendidos en el centro hospitalario.

OBJETIVO GENERAL

Distribuir los roles del personal hospitalario y controlar las labores que cada rol debe ejercer.

Link del video: <https://youtu.be/skk-VOqUrDI>

OBJETIVOS ESPECÍFICOS

- Mejorar la organización del personal médico

- Reducir el tiempo perdido para la atención a pacientes

JUSTIFICACIÓN

La implementación de un formulario de roles para el personal de un hospital puede ser de gran utilidad para mejorar la eficiencia y efectividad en la atención de los pacientes. Al contar con un formulario que identifique claramente las funciones y responsabilidades de cada miembro del personal del hospital, se puede reducir la duplicación de tareas, mejorar la coordinación entre los diferentes departamentos y equipos, y garantizar una atención más efectiva y eficiente a los pacientes.

Además, un formulario de roles también puede ayudar a mejorar la comunicación interna y externa del hospital, ya que proporciona una fuente de información clara y concisa sobre las funciones y responsabilidades del personal. También puede ser utilizado como una herramienta de capacitación para el personal nuevo o para aquellos que cambian de roles, asegurando que comprendan claramente sus responsabilidades y las expectativas del hospital. En resumen, la implementación de un formulario de roles para el personal de un hospital puede mejorar la eficiencia, efectividad y comunicación interna del hospital, lo que se traduce en una atención de calidad y segura para los pacientes.

PLANTEAMIENTO DEL PROBLEMA

El problema a resolver con la creación de un formulario de roles para el personal de un hospital es la falta de claridad y organización en la asignación de tareas y responsabilidades a los empleados. Muchas veces, los trabajadores pueden sentirse confundidos sobre lo que se espera de ellos o puede haber superposición de tareas entre diferentes miembros del personal. Esto puede resultar en una falta de eficiencia y productividad en el hospital, lo que a su vez puede afectar negativamente la calidad de la atención al paciente. La creación de un formulario de roles claro y conciso ayudaría a los empleados a comprender mejor sus responsabilidades y aseguraría que cada tarea sea asignada a la persona adecuada en el momento adecuado, mejorando así la eficiencia y la calidad de la atención al paciente en el hospital.

Link del video: <https://youtu.be/skk-VOqUrDI>

TECNOLOGÍAS UTILIZADAS

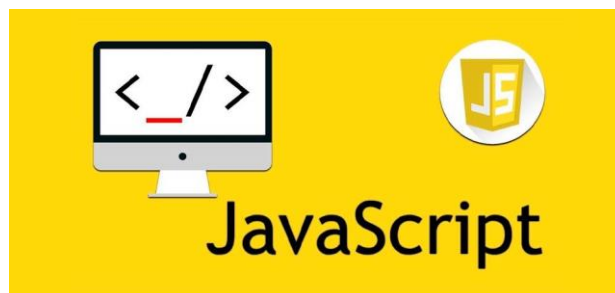
PHP



Se utilizó el lenguaje de programación PHP como base para el desarrollo del sistema de gestión de roles del personal hospitalario. PHP, acrónimo de "Hypertext Preprocessor", es un lenguaje de programación ampliamente utilizado en la creación de aplicaciones web dinámicas.

El uso de PHP permitió la implementación de características clave en el sistema, como la asignación de roles y responsabilidades a los miembros del personal, el seguimiento de su disponibilidad y la comunicación interna. Con PHP, fue posible crear una interfaz de usuario dinámica y amigable que facilita la interacción del personal hospitalario con el sistema.

JAVASCRIPT



Se utilizó el lenguaje de programación JavaScript para complementar el desarrollo del sistema de gestión de roles del personal hospitalario. JavaScript es un lenguaje de programación ampliamente utilizado en el desarrollo web y se ejecuta en el lado del cliente, lo que significa que se ejecuta en el navegador web del usuario final.

JavaScript se utilizó en este proyecto para agregar interactividad y dinamismo a la interfaz de usuario del sistema de gestión de roles. Con JavaScript, fue posible crear funcionalidades

Link del video: <https://youtu.be/skk-VOqUrDI>

como la validación de formularios, la manipulación y actualización en tiempo real de la información en la interfaz, y la respuesta instantánea a las acciones del usuario sin necesidad de recargar la página.

Además, JavaScript permitió la implementación de características como la gestión de eventos, que capturan las acciones del usuario, como hacer clic en un botón o seleccionar una opción de un menú desplegable, y responden a esos eventos ejecutando el código correspondiente. Esto resulta especialmente útil en un sistema de gestión de roles, donde es necesario responder de manera rápida y eficiente a las interacciones del usuario.

JQUERY



Se utilizó la biblioteca jQuery para complementar el desarrollo del sistema de gestión de roles del personal hospitalario. jQuery es una biblioteca de JavaScript que simplifica la manipulación y la interacción con el documento HTML, el manejo de eventos y la realización de peticiones AJAX de manera más concisa y eficiente.

jQuery se utilizó para agilizar y simplificar el desarrollo de funcionalidades interactivas en el sistema de gestión de roles. Proporciona una sintaxis sencilla y consistente que facilita la manipulación del DOM (Document Object Model) y la interacción con elementos HTML. Con jQuery, se pudo acceder fácilmente a los elementos de la interfaz de usuario, modificar su contenido, estilos y atributos, así como capturar eventos y responder a ellos de manera eficiente.

Link del video: <https://youtu.be/skk-VOqUrDI>

BOOTSTRAP



Se utilizó el framework Bootstrap para facilitar y agilizar el desarrollo del sistema de gestión de roles del personal hospitalario. Bootstrap es un framework de front-end que proporciona un conjunto de herramientas y estilos CSS predefinidos, así como componentes y utilidades de JavaScript, con el objetivo de crear interfaces web responsivas y visualmente atractivas de manera más eficiente.

Bootstrap se utilizó para establecer una base sólida y coherente en el diseño y la presentación de la interfaz de usuario del sistema de gestión de roles. Su amplia variedad de clases CSS predefinidas permitió aplicar estilos consistentes y profesionales a los elementos HTML, lo que agilizó el desarrollo y aseguró una apariencia visualmente atractiva en diferentes dispositivos y tamaños de pantalla.

MYSQL



Se utilizó MySQL como sistema de gestión de bases de datos para respaldar y almacenar la información relacionada con el sistema de gestión de roles del personal hospitalario. MySQL es un sistema de gestión de bases de datos relacional de código abierto que ofrece un rendimiento sólido, seguridad y escalabilidad, lo que lo convierte en una opción confiable para aplicaciones web.

MySQL se utilizó para crear y gestionar las tablas de base de datos que almacenan los datos

Link del video: <https://youtu.be/skk-VOqUrDI>

esenciales del sistema, como la información del personal hospitalario, los roles asignados, la disponibilidad y otras características relevantes. Con MySQL, se pudo definir la estructura de la base de datos, establecer relaciones entre las tablas y garantizar la integridad de los datos.

DIAGRAMAS

Diagrama de clases

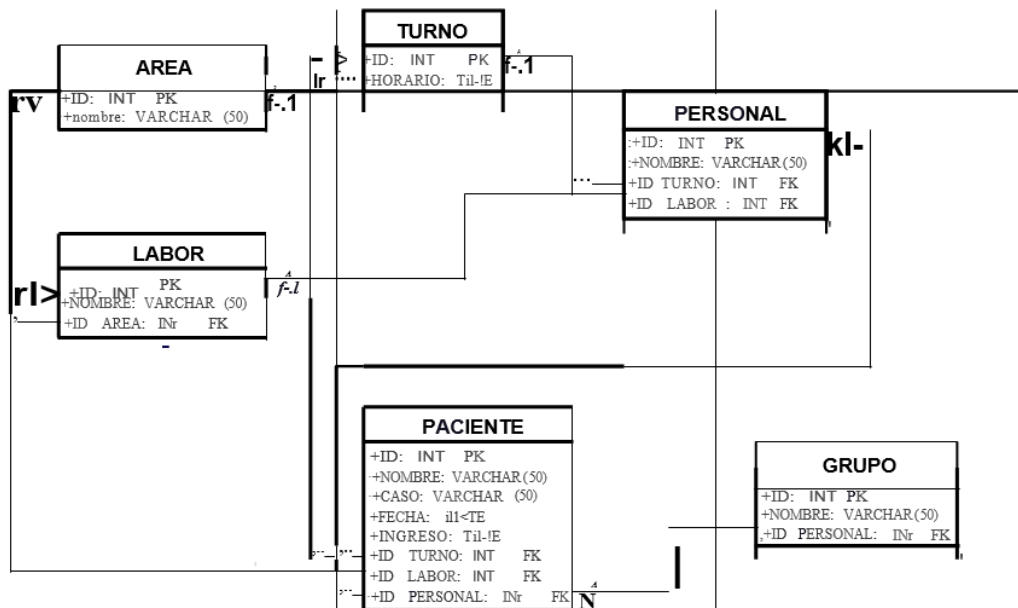
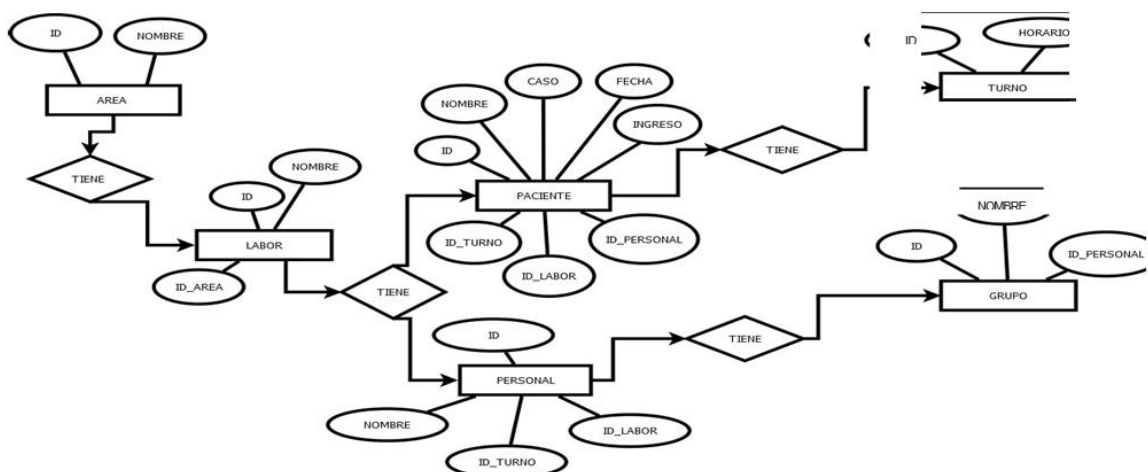


Diagrama de flujo.



Link del video: <https://youtu.be/skk-VOqUrDI>

CONCLUSIONES

- El desarrollo del sistema de gestión de roles del personal hospitalario utilizando tecnologías como PHP, JavaScript, jQuery, MySQL y Bootstrap ha sido fundamental para mejorar la eficiencia, el aprendizaje y la organización de los conocimientos adquiridos y utilizados para la creación de este sistema. Estas tecnologías han permitieron la creación de una interfaz de usuario dinámica, intuitiva y visualmente atractiva, que facilita la asignación de tareas y la comunicación interna entre el personal.

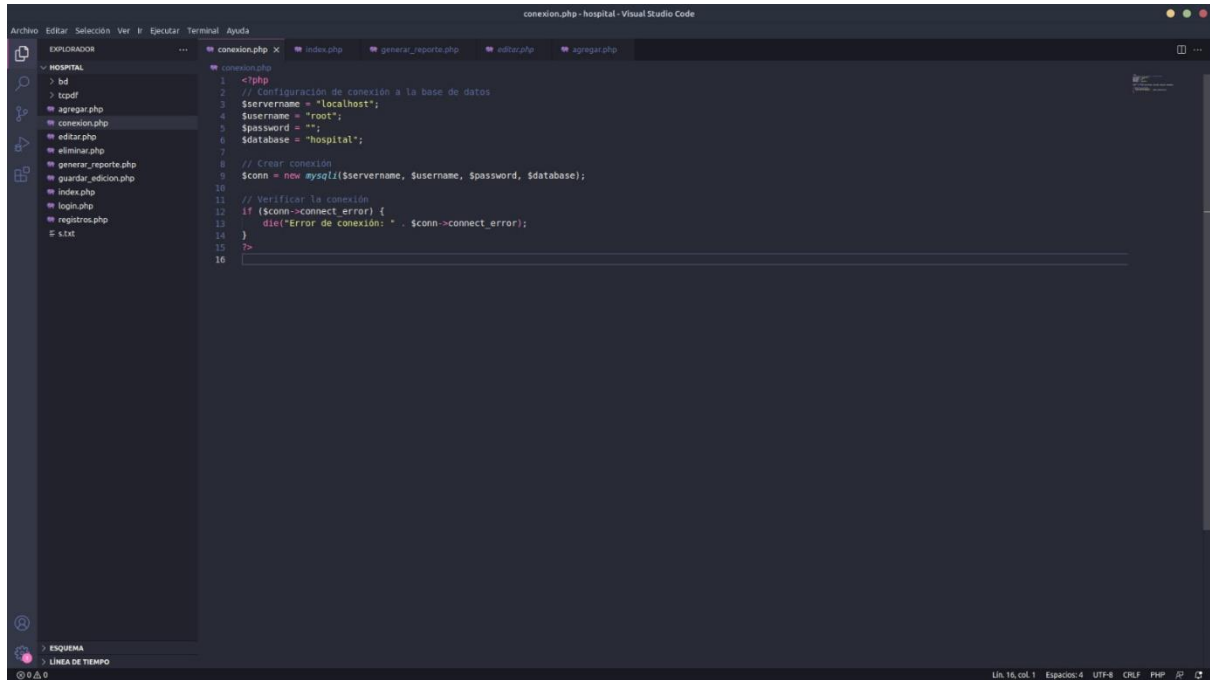
- La combinación de PHP como lenguaje de programación principal, junto con JavaScript y jQuery para la interactividad y Bootstrap para la apariencia y responsividad, ha resultado en un sistema completo y efectivo. El uso de estas tecnologías ha optimizado y facilitado el manejo y uso de este sistema creado.

Link del video: <https://youtu.be/skk-VOqUrDI>

ANEXOS

Manual del programador.

1. La conexión a la base de datos.



```
1 <?php
2 // Configuración de conexión a la base de datos
3 $servername = "localhost";
4 $username = "root";
5 $password = "";
6 $database = "hospital";
7
8 // Error conexión
9 $conn = new mysqli($servername, $username, $password, $database);
10
11 // Verificar la conexión
12 if ($conn->connect_error) {
13     die("Error de conexión: " . $conn->connect_error);
14 }
15 ?>
16
```

Primero se deben definir las variables necesarias para establecer la conexión a la base de datos. Estas variables incluyen el nombre del servidor (localhost en este caso), el nombre de usuario y contraseña para acceder al servidor de la base de datos (root y una cadena vacía respectivamente) y el nombre de la base de datos (hospital en este ejemplo).

Se crea una nueva instancia de la clase mysqli utilizando los parámetros de conexión definidos anteriormente. Esta clase proporciona métodos para interactuar con la base de datos MySQL.

Luego, se verifica si la conexión se ha establecido correctamente utilizando la propiedad connect_error de la instancia mysqli. Si hay un error de conexión, se muestra un mensaje de error y el script se detiene.

Si la conexión se establece exitosamente, se puede utilizar el objeto \$conn para realizar consultas y otras operaciones en la base de datos.

2. Formularios.

Link del video: <https://youtu.be/skk-VOqUrDI>

```
index.php - hospital - Visual Studio Code
Archivo Editar Selección Ver Ejecutar Terminal Ayuda
EXPLORADOR
  HOSPITAL
    bd
    tpdf
    agregar.php
    conexion.php
    editar.php
    eliminar.php
    genera_reporte.php
    guardar_edicion.php
    index.php
    login.php
    registros.php
    s.txt
  ESQUEMA
  LINEA DE TIEMPO
index.php
1 <?php
2 include 'conexion.php';
3 ?>
4 <!DOCTYPE html>
5 <html>
6 <head>
7 <title>CRUD - Agregar nuevo registro</title>
8 <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css">
9 <style>
10 .logout-btn {
11   position: fixed;
12   top: 10px;
13   right: 10px;
14 }
15 </style>
16 </head>
17 <body>
18 <div class="container">
19 <h2>Personal Hospitalario</h2>
20
21 <!-- Formulario para agregar nuevo registro -->
22 <h3>Agregar nuevo registro</h3>
23 <form method="GET" action="agregar.php">
24   <div class="mb-3">
25     <label form="nombre" class="form-label">Nombre:</label>
26     <input type="text" class="form-control" id="nombre" name="nombre" required>
27   </div>
28   <div class="mb-3">
29     <label form="turno" class="form-label">Turno:</label>
30     <select class="form-select" id="turno" name="turno" required>
31       <option value="">Seleccionar turno</option>
32     </select>
33     <!-- Obtener opciones de turno desde la base de datos -->
34     <pre>$conn = new mysqli($servername, $username, $password, $database);
35     $sql = "SELECT * FROM turno";
36     $result = $conn->query($sql);
37
38     if ($result->num_rows > 0) {
39       while ($row = $result->fetch_assoc()) {
40         echo "<option value='".$row['id']."'> ".$row['horario'] . "</option>";
41       }
42     }
43
44     $conn->close();
45     </pre>
46     </div>
47   <div class="mb-3">
48     <label form="area" class="form-label">Área:</label>
49     <select class="form-select" id="area" name="area" required>
50
```

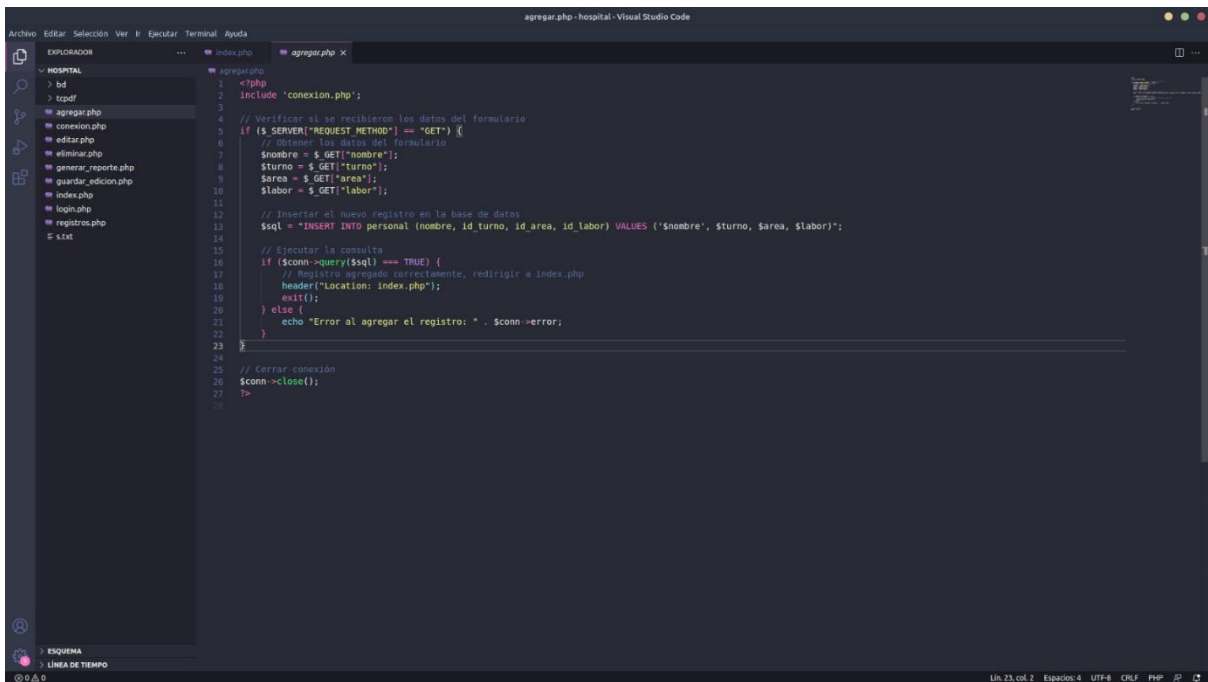
Este código muestra un formulario que permite agregar nuevos registros en un sistema de gestión de personal hospitalario, utilizando consultas a la base de datos para llenar los selectores con datos dinámicamente.

```
login.php - hospital - Visual Studio Code
Archivo Editar Selección Ver Ejecutar Terminal Ayuda
EXPLORADOR
  HOSPITAL
    bd
    tpdf
    agregar.php
    conexion.php
    editar.php
    eliminar.php
    genera_reporte.php
    guardar_edicion.php
    index.php
    login.php
    registros.php
    s.txt
  ESQUEMA
  LINEA DE TIEMPO
login.php
26 $username = $_POST["username"];
27 $password = $_POST["password"];
28
29 // Validar las credenciales
30 if (validateCredentials($username, $password)) {
31   // Inicio de sesión exitoso
32   $_SESSION["username"] = $username;
33   header("Location: index.php");
34   exit();
35 } else {
36   // Credenciales incorrectas
37   $error = "Nombre de usuario o contraseña incorrectos.";
38 }
39
40 ?>
41
42 <!DOCTYPE html>
43 <html>
44 <head>
45 <title>Iniciar sesión</title>
46 <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css">
47 <style>
48   body {
49     display: flex;
50     justify-content: center;
51     align-items: center;
52     height: 100vh;
53   }
54 </style>
55 </head>
56 <body>
57 <div class="container">
58 <h1 class="text-center">Iniciar sesión</h1>
59 <form method="POST" action="<?php echo htmlspecialchars($_SERVER["PHP_SELF"]); ?>">
60 <div class="mb-3">
61 <label form="username" class="form-label">Nombre de usuario:</label>
62 <input type="text" class="form-control" id="username" name="username" required>
63 </div>
64 <div class="mb-3">
65 <label form="password" class="form-label">Contraseña:</label>
66 <input type="password" class="form-control" id="password" name="password" required>
67 </div>
68 <button type="submit" class="btn btn-primary">Iniciar sesión</button>
69 </form>
70
71 <?php if (isset($error)) :>
72 <div class="alert alert-danger mt-3"><?php echo $error; ?></div>
73 <?php endif; ?>
74 </div>
75
```

Este código implementa un formulario de inicio de sesión que valida las credenciales ingresadas por el usuario y establece una sesión si son correctas.

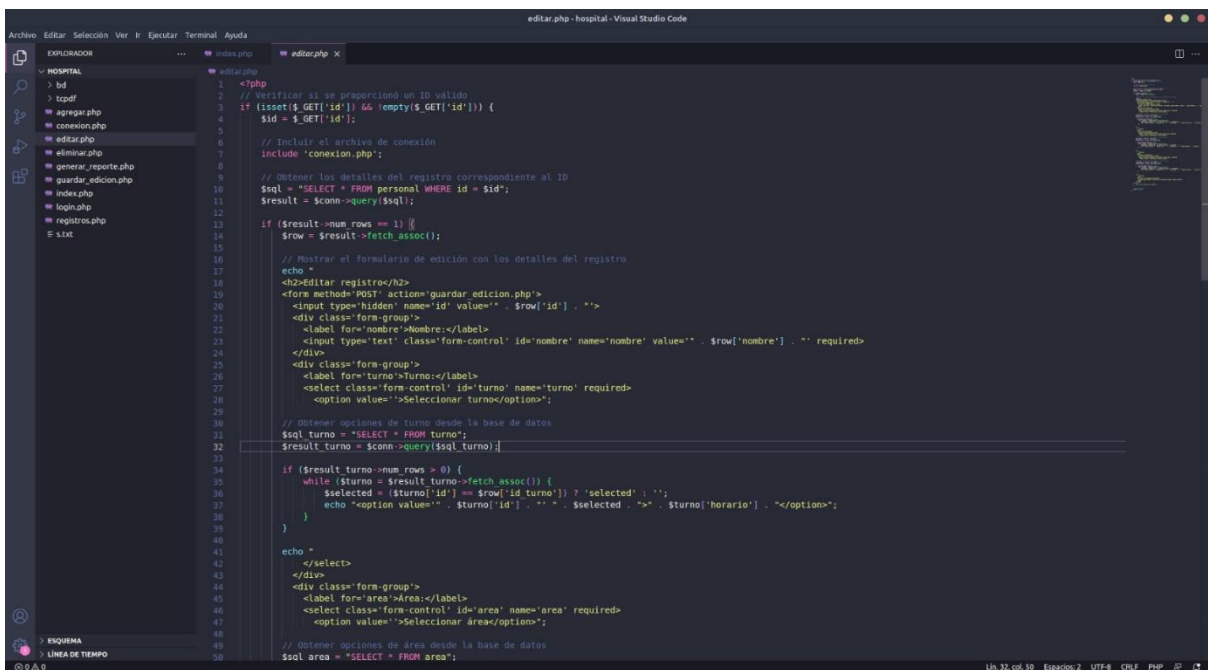
Link del video: <https://youtu.be/skk-VOqUrDI>

3. Registros.



```
agregar.php - hospital - Visual Studio Code
Archivo Editar Selección Ver Ejecutar Terminal Ayuda
EXPLORADOR
HOSPITAL
  > bd
  > tcpdf
  agregar.php
  conexion.php
  editar.php
  eliminar.php
  generar_reporte.php
  guardar_edicion.php
  index.php
  login.php
  registros.php
  s.txt
ESQUEMA
LINEA DE TIEMPO
agregar.php
1 <?php
2 include 'conexion.php';
3
4 // Verificar si se recibieron los datos del formulario
5 if ($SERVER["REQUEST_METHOD"] == "GET") {
6 // Obtener los datos del formulario
7 $nombre = $_GET['nombre'];
8 $turno = $_GET['turno'];
9 $area = $_GET['area'];
10 $labor = $_GET['labor'];
11
12 // Insertar el nuevo registro en la base de datos
13 $sql = "INSERT INTO personal (nombre, id_turno, id_area, id_labor) VALUES ('$nombre', $turno, $area, $labor)";
14
15 // Ejecutar la consulta
16 if ($conn->query($sql) === TRUE) {
17 // Registro agregado correctamente, redirigir a index.php
18 header("Location: index.php");
19 exit();
20 } else {
21 echo "Error al agregar el registro: " . $conn->error;
22 }
23
24
25 // Cerrar conexión
26 $conn->close();
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
Lin 23, col 2 Espacios: 4 UTF-8 CRLF PHP
```

Este código inserta un nuevo registro en la tabla "personal" de la base de datos utilizando los datos enviados desde un formulario anterior y maneja posibles errores durante el proceso de inserción.



```
editar.php - hospital - Visual Studio Code
Archivo Editar Selección Ver Ejecutar Terminal Ayuda
EXPLORADOR
HOSPITAL
  > bd
  > tcpdf
  agregar.php
  conexion.php
  editar.php
  eliminar.php
  generar_reporte.php
  guardar_edicion.php
  index.php
  login.php
  registros.php
  s.txt
ESQUEMA
LINEA DE TIEMPO
editar.php
1 <?php
2 // Verificar si se proporcionó un ID válido
3 if (isset($_GET['id']) && !empty($_GET['id'])) {
4 $id = $_GET['id'];
5
6 // Incluir el archivo de conexión
7 include 'conexion.php';
8
9 // Obtener los detalles del registro correspondiente al ID
10 $sql = "SELECT * FROM personal WHERE id = $id";
11 $result = $conn->query($sql);
12
13 if ($result->num_rows == 1) {
14 $row = $result->fetch_assoc();
15
16 // Mostrar el formulario de edición con los detalles del registro
17 echo "
18 <?>editar registro/>
19 <form method='POST' action='guardar_edicion.php'>
20 <input type='hidden' name='id' value='". $row['id']."' />
21 <div class='form-group'>
22 <label for='nombre'>Nombre:</label>
23 <input type='text' class='form-control' id='nombre' name='nombre' value='". $row['nombre']."' required>
24 </div>
25 <div class='form-group'>
26 <label for='turno'>Turno:</label>
27 <select class='form-control' id='turno' name='turno' required>
28 <option values="">Seleccionar turno/</options>
29
30 // Obtener opciones de turno desde la base de datos
31 $sql turno = "SELECT * FROM turno";
32 $result turno = $conn->query($sql turno);
33
34 if ($result turno->num_rows > 0) {
35 while ($turno = $result turno->fetch_assoc()) {
36 $selected = ($turno['id'] == $row['id turno']) ? 'selected' : '';
37 echo <option value="" . $turno['id'] . "' . $selected . "> . $turno['horario'] . "</option>;
38 }
39 }
40
41 echo "
42 </select>
43 </div>
44 <div class='form-group'>
45 <label for='area'>Área:</label>
46 <select class='form-control' id='area' name='area' required>
47 <option values="">Seleccionar área/</options>
48
49 // Obtener opciones de área desde la base de datos
50 $sql area = "SELECT * FROM area";
Lin 32, col 19 Espacios: 2 UTF-8 CRLF PHP
```

Este código muestra un formulario de edición con los detalles de un registro específico en la base de datos y permite al usuario modificar y guardar los cambios en el registro seleccionado.

Link del video: <https://youtu.be/skk-VOqUrDI>

```
eliminar.php - hospital - Visual Studio Code
Archivo Editar Selección Ver Ejecutar Terminal Ayuda
EXPLORADOR
> HOSPITAL
  > bd
  > tcpdf
  > agregar.php
  > conexion.php
  > editar.php
  > eliminar.php
  > generar_reporte.php
  > guardar_edicion.php
  > index.php
  > login.php
  > registros.php
  > s.txt
> ESQUEMA
> LINEA DE TIEMPO
eliminar.php x
1 <?php
2 // Verificar si se recibió el ID del registro a eliminar
3 if (isset($_GET['id'])) {
4     $id = $_GET['id'];
5
6     include 'conexion.php';
7
8     // Eliminar el registro de la base de datos
9     $sql = "DELETE FROM personal WHERE id = $id";
10
11     if ($conn->query($sql) === TRUE) {
12         // Registro eliminado correctamente, redirigir a index.php
13         header("Location: registros.php");
14         exit();
15     } else {
16         echo "Error al eliminar el registro: " . $conn->error;
17     }
18
19     // Cerrar conexión
20     $conn->close();
21 } else {
22     echo "ID del registro no especificado.";
23     exit();
24 }
25 ?>
26
```

Este código se encarga de eliminar un registro específico de la base de datos cuando se proporciona su ID. Después de la eliminación, el usuario es redirigido a una página que muestra la lista actualizada de registros.

```
generar_reporte.php - hospital - Visual Studio Code
Archivo Editar Selección Ver Ejecutar Terminal Ayuda
EXPLORADOR
> HOSPITAL
  > bd
  > tcpdf
  > agregar.php
  > conexion.php
  > editar.php
  > eliminar.php
  > generar_reporte.php
  > guardar_edicion.php
  > index.php
  > login.php
  > registros.php
  > s.txt
> ESQUEMA
> LINEA DE TIEMPO
generar_reporte.php x
1 <?php
2 // Incluir archivo de conexión
3 include 'conexion.php';
4
5 // Consulta para obtener los registros de médicos
6 $sql = "SELECT p.nombre, t.horario, a.nombre AS area, l.nombre AS labor FROM personal p
7         INNER JOIN turno t ON p.id_turno = t.id
8         INNER JOIN area a ON p.id_area = a.id
9         INNER JOIN labor l ON p.id_labor = l.id";
10 $result = $conn->query($sql);
11
12 if ($result->num_rows > 0) {
13     // Generar el informe en formato HTML
14     $reporte_html = "<h1>Informe de Médicos</h1>";
15     $reporte_html .= "<table class='table'><thead><tr><th>Nombre</th><th>Horario</th><th>Area</th><th>Labor</th></tr></thead><tbody>";
16     while ($row = $result->fetch_assoc()) {
17         $reporte_html .= "<tr><td>". $row["nombre"] . "</td><td>". $row["horario"] . "</td><td>". $row["area"] . "</td><td>". $row["labor"] . "</td></tr>";
18     }
19     $reporte_html .= "</tbody></table>";
20
21     // Generar archivo PDF utilizando una biblioteca como TCPDF o FPDF
22     // Aquí se muestra un ejemplo utilizando TCPDF
23
24     // Incluir la biblioteca TCPDF
25     require_once('tcpdf/tcpdf.php');
26
27     // Limpiar cualquier salida previa antes de generar el PDF
28     ob_end_clean();
29
30     // Crear una nueva instancia de TCPDF
31     $pdf = new TCPDF();
32
33     // Establecer algunas configuraciones del documento
34     $pdf->setCreator('Tu Nombre');
35     $pdf->setAuthor('Tu Nombre');
36     $pdf->setTitle('Informe de Médicos');
37     $pdf->setHeaderData('', 0, '', array(0, 0, 0), array(255, 255, 255));
38     $pdf->setHeaderFont(array('helvetica', '', 10));
39     $pdf->setFooterFont(array('helvetica', '', 8));
40     $pdf->setDefaultMonospacedFont('courier');
41     $pdf->setMargins(10, 10, 10);
42     $pdf->setAutoPageBreak(true, 10);
43     $pdf->setImageScale(1.25);
44     $pdf->setFontSubsetting(true);
45     $pdf->setFont('dejavusans', '', 10, '', true);
46
47     // Agregar una página
48     $pdf->addPage();
49
50     // Escribir el contenido HTML en el documento PDF

```

Este código consulta la base de datos para obtener los registros de médicos, genera un informe en formato PDF utilizando los datos obtenidos y la biblioteca TCPDF, y envía el archivo PDF al navegador para su descarga o visualización.

Link del video: <https://youtu.be/skk-VOqUrDI>

```
guardar_edicion.php - hospital - Visual Studio Code
Archivo Editar Selección Ver Ejecutar Terminal Ayuda
EXPLORADOR
> HOSPITAL
  > bd
  > tcpdf
  > agregar.php
  > conexion.php
  > editar.php
  > eliminar.php
  > genera_reporte.php
  > guardar_edicion.php
  > index.php
  > login.php
  > registros.php
  > s.txt
ESQUEMA
LINEA DE TIEMPO
guardar_edicion.php X
guardar_edicion.php
1 <?php
2 // Verificar si se recibieron los datos del formulario
3 if (isset($_POST['id']) && !empty($_POST['id']) && isset($_POST['nombre']) && !empty($_POST['nombre']) && isset($_POST['turno']) && !empty($_POST['turno']) && isset($_POST['area']) && !empty($_POST['area']) && isset($_POST['labor']) && !empty($_POST['labor'])) {
4     $id = $_POST['id'];
5     $nombre = $_POST['nombre'];
6     $turno = $_POST['turno'];
7     $area = $_POST['area'];
8     $labor = $_POST['labor'];
9
10    include 'conexion.php';
11
12    // Actualizar los datos en la base de datos
13    $sql = "UPDATE personal SET nombre = '$nombre', id_turno = '$turno', id_area = '$area', id_labor = '$labor' WHERE id = $id";
14    if ($conn->query($sql) == TRUE) {
15        // Redirigir al index.php con los datos actualizados
16        header("Location: registros.php?id=$id&nombre=$nombre&turno=$turno&area=$area&labor=$labor");
17        exit();
18    } else {
19        echo "Error al actualizar el registro: " . $conn->error;
20    }
21
22    // Cerrar conexión
23    $conn->close();
24 } else {
25    echo "No se recibieron los datos del formulario.";
26 }
27
28
```

Este código procesa los datos enviados desde un formulario de edición, actualiza los registros correspondientes en la base de datos y redirige al usuario a una página que muestra los datos actualizados.

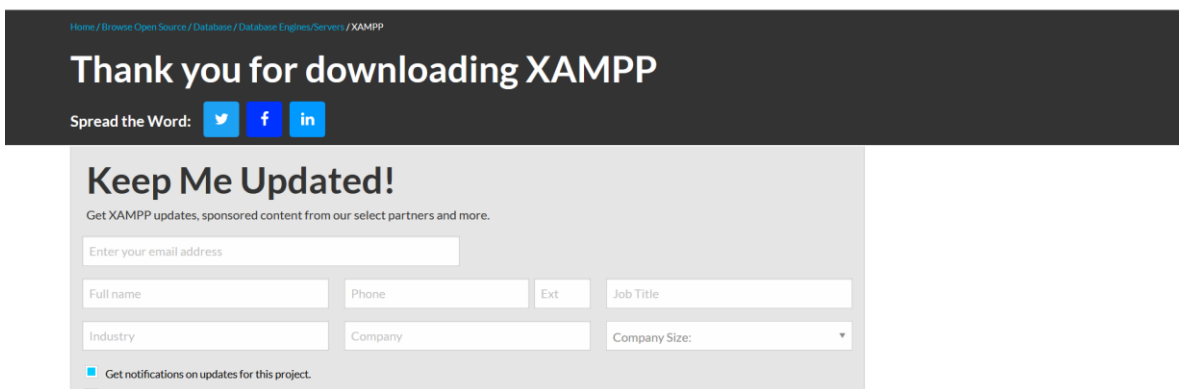
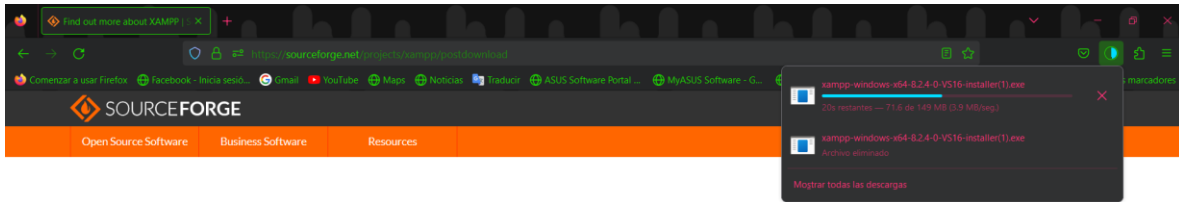
```
registros.php - hospital - Visual Studio Code
Archivo Editar Selección Ver Ejecutar Terminal Ayuda
EXPLORADOR
> HOSPITAL
  > bd
  > tcpdf
  > agregar.php
  > conexion.php
  > editar.php
  > eliminar.php
  > genera_reporte.php
  > guardar_edicion.php
  > index.php
  > login.php
  > registros.php
  > s.txt
ESQUEMA
LINEA DE TIEMPO
registros.php X
registros.php
1 <!DOCTYPE html>
2 <html>
3 <head>
4 <title>CRUD Personal Hospitalario</title>
5 <link rel="stylesheet" href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.css">
6 </head>
7 <body>
8 <div class="container">
9 <h1>CRUD Personal Hospitalario</h1>
10 <a href="index.php" class="btn btn-primary" style="position: absolute; top: 10px; left: 10px;">Regresar</a>
11
12 <!-- Mostrar registros existentes -->
13 <h2>Registros existentes</h2>
14 <?php
15 // Incluir archivo de configuración
16 include 'conexion.php';
17
18 // Consultar registros existentes
19 $sql = "SELECT personal.id, personal.nombre, personal.turno, personal.area AS area_nombre, personal.labor AS labor_nombre
20 FROM personal
21 INNER JOIN turno ON personal.id_turno = turno.id
22 INNER JOIN area ON personal.id_area = area.id
23 INNER JOIN labor ON personal.id_labor = labor.id";
24
25 $result = $conn->query($sql);
26
27 if ($result->num_rows > 0) {
28     echo <table class="table">
29 <tr>
30 <th>ID</th>
31 <th>Nombre</th>
32 <th>Turno</th>
33 <th>Area</th>
34 <th>Labor</th>
35 <th>Acciones</th>
36 </tr>
37
38 while ($row = $result->fetch_assoc()) {
39     echo <tr>
40 <td> $row['id'] </td>
41 <td> $row['nombre'] </td>
42 <td> $row['turno'] </td>
43 <td> $row['area_nombre'] </td>
44 <td> $row['labor_nombre'] </td>
45 <td>
46 <button type="button" class="btn btn-primary" data-bs-toggle="modal" data-bs-target="#editarModal" data-id="$row['id']">Editar</button>
47 <a href="eliminar.php?id=$row['id']" class="btn btn-danger">Eliminar</a>
48 <a href="genera_reporte.php?id=$row['id']" class="btn btn-info">Reporte</a>
49 </td>
50 </tr>
51 </table>
52
```

Este código genera una página web que muestra los registros existentes de personal hospitalario en una tabla y permite realizar acciones como editar, eliminar y generar informes para cada registro.

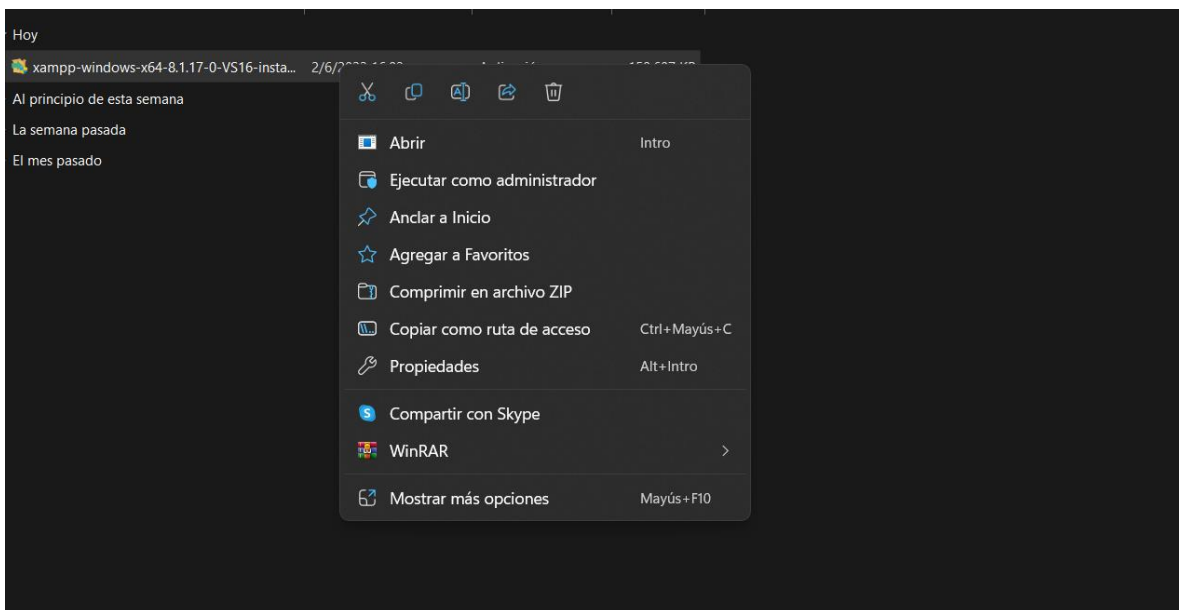
Link del video: <https://youtu.be/skk-VOqUrDI>

Instalación de XAMPP.

1. Descarga el archivo de instalación: Ve al sitio web oficial de XAMPP (<https://www.apachefriends.org/index.html>) y descarga la versión de XAMPP compatible con tu sistema operativo Windows.

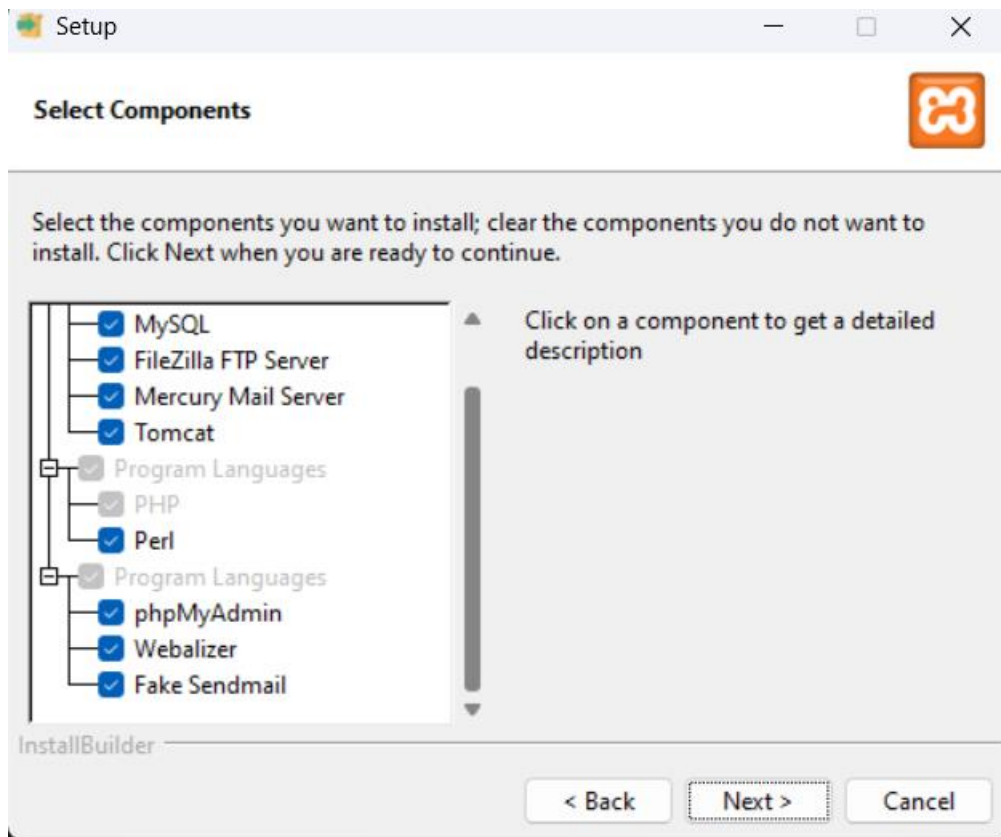


2. Ejecuta el archivo de instalación: Una vez que la descarga esté completa, haz doble clic en el archivo de instalación de XAMPP para ejecutarlo. Es posible que se te solicite permisos de administrador, así que asegúrate de aceptarlos si aparece una ventana emergente de control de cuentas de usuario.

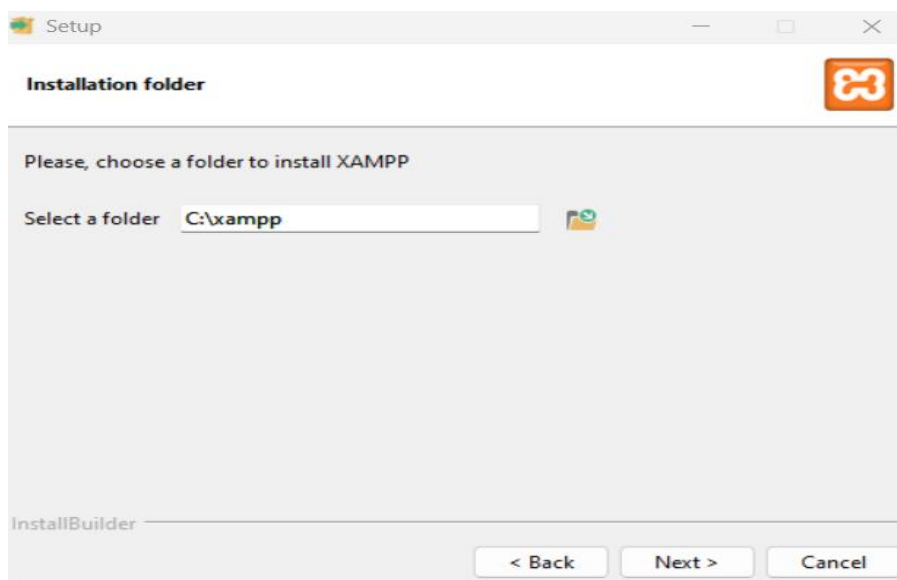


3. Selecciona los componentes a instalar: Durante el proceso de instalación, se te presentará una lista de componentes disponibles como Apache, MySQL, PHP, etc. Luego haz clic en "Next" (Siguiente).

Link del video: <https://youtu.be/skk-VOqUrDI>



4. Selecciona la ubicación de instalación: A continuación, se te pedirá que elijas una ubicación para la instalación de XAMPP. Puedes aceptar la ubicación predeterminada o elegir otra. Haz clic en "Next" (Siguiete) una vez que hayas hecho tu elección.

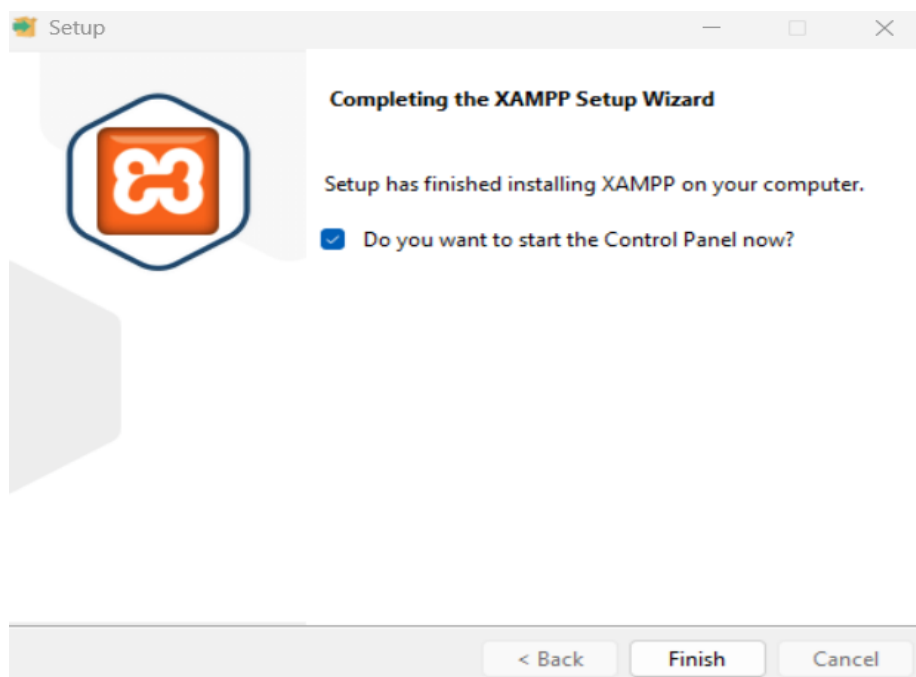


5. Completa la instalación: En esta etapa, se mostrará un resumen de la configuración que has seleccionado.

Link del video: <https://youtu.be/skk-VOqUrDI>

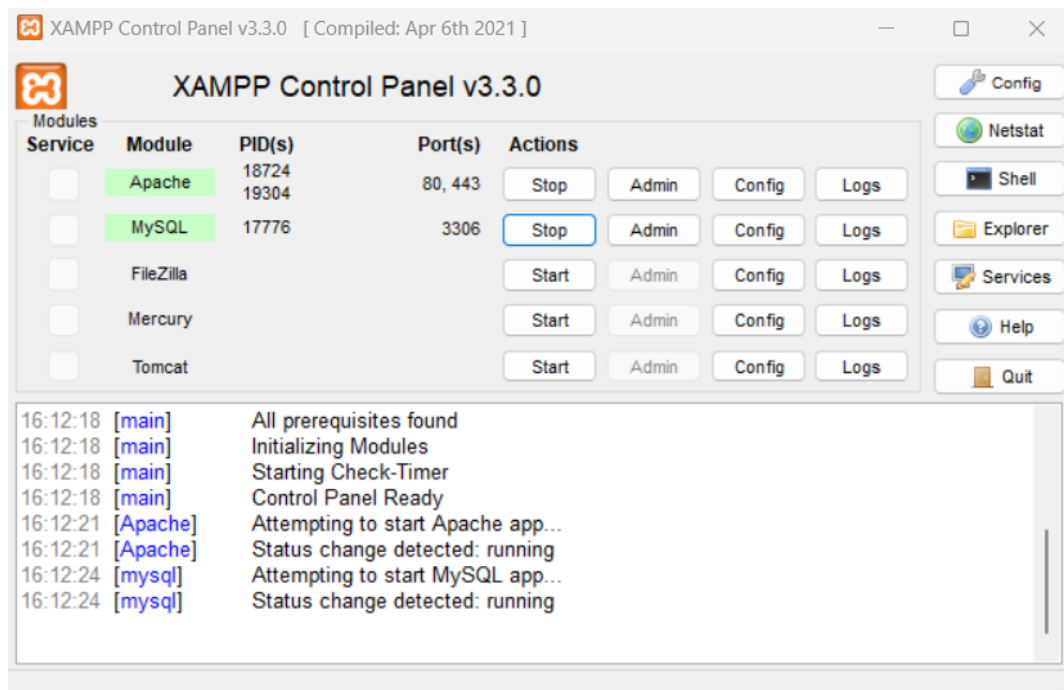


6. Finaliza la instalación: Una vez que la instalación esté completa, se te mostrará una pantalla de finalización. Asegúrate de marcar la opción ¿"Do you want to start the Control Panel now?" (¿Deseas iniciar el Panel de Control ahora?) y luego haz clic en "Finish" (Finalizar).

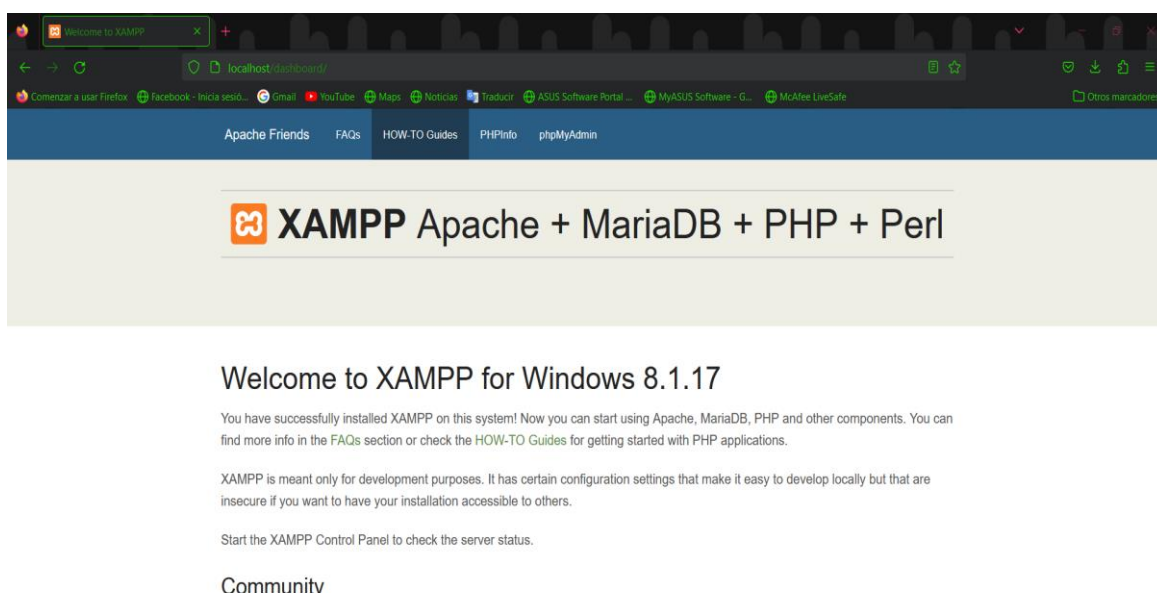


Link del video: <https://youtu.be/skk-VOqUrDI>

7. Inicia los servicios de XAMPP: Después de hacer clic en "Finish" (Finalizar), se abrirá el Panel de Control de XAMPP. Aquí puedes iniciar o detener los servicios de Apache, MySQL y otros componentes según sea necesario. Haz clic en los botones "Start" (Iniciar) junto a Apache y MySQL para iniciar los servicios.



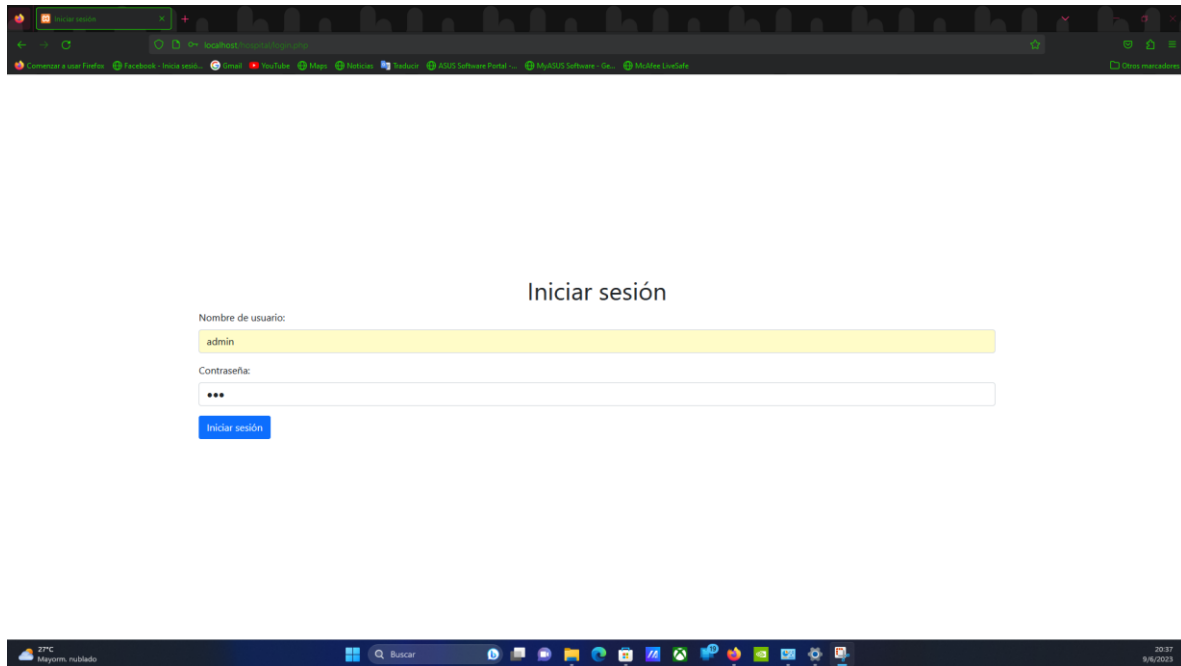
8. Una vez que los servicios estén en funcionamiento, podrás acceder a tus sitios web, abre un navegador web y escribe "localhost" en la barra de direcciones.



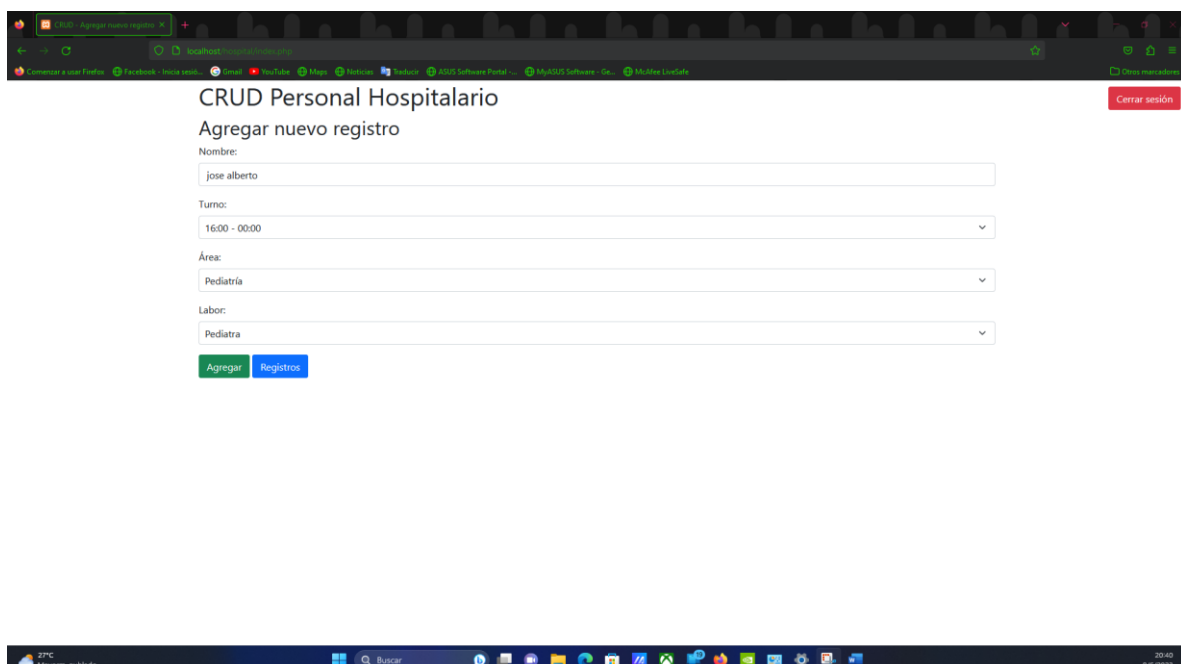
Link del video: <https://youtu.be/skk-VOqUrDI>

Manual de usuario.

1. Después de haber instalado xampp y poner la carpeta del proyecto en htdocs poner la siguiente dirección <http://localhost/hospital/login.php> e ingresar con las credenciales correspondientes

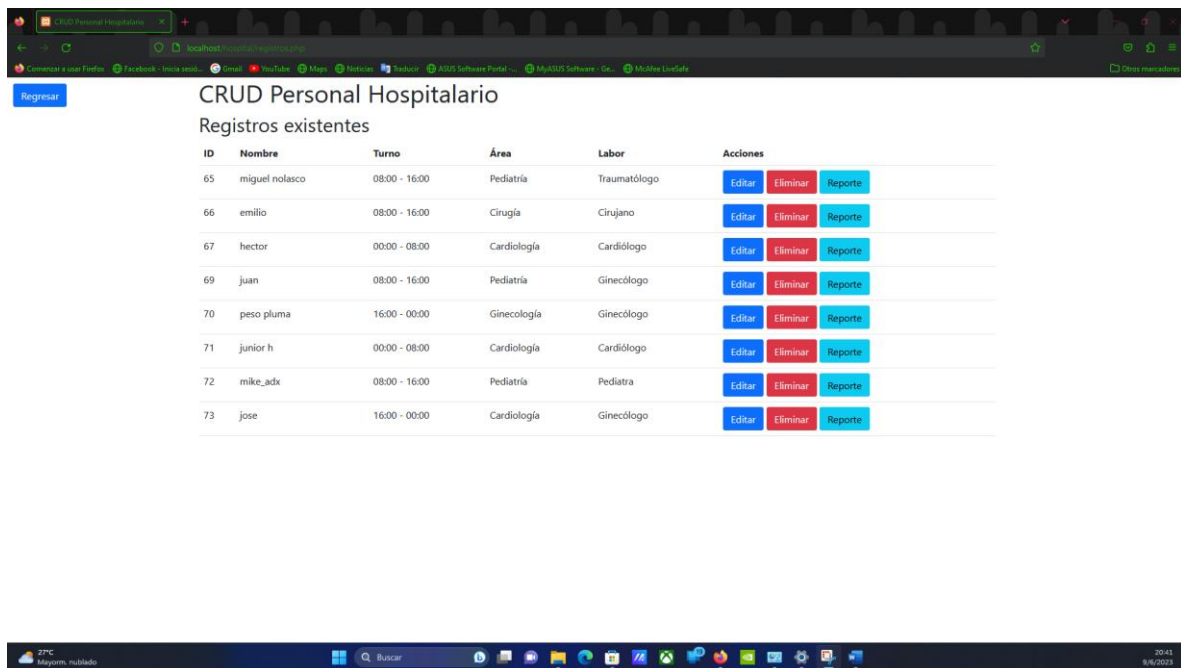


2. Nos llevara al panel principal en el cual esta el formulario para agregar al personal al dar clic al botón agregar



Link del video: <https://youtu.be/skk-VOqUrDI>


3. Al dar clic al botón registros mostrara todos los registros existentes



The screenshot shows a web browser window with the URL 'localhost:3000/registros'. The page title is 'CRUD Personal Hospitalario' and the main heading is 'Registros existentes'. Below the heading is a table with 8 columns: ID, Nombre, Turno, Área, Labor, and Acciones. The table contains 8 rows of data. Each row has three buttons in the 'Acciones' column: 'Editar' (blue), 'Eliminar' (red), and 'Reporte' (cyan). The browser's taskbar at the bottom shows the system tray with the date '1/6/2023' and time '20:41'.

ID	Nombre	Turno	Área	Labor	Acciones
65	miguel nolasco	08:00 - 16:00	Pediatría	Traumatólogo	Editar Eliminar Reporte
66	emilio	08:00 - 16:00	Cirugía	Cirujano	Editar Eliminar Reporte
67	hector	00:00 - 08:00	Cardiología	Cardiólogo	Editar Eliminar Reporte
69	juan	08:00 - 16:00	Pediatría	Ginecólogo	Editar Eliminar Reporte
70	peso pluma	16:00 - 00:00	Ginecología	Ginecólogo	Editar Eliminar Reporte
71	junior h	00:00 - 08:00	Cardiología	Cardiólogo	Editar Eliminar Reporte
72	mike_adx	08:00 - 16:00	Pediatría	Pediatra	Editar Eliminar Reporte
73	jose	16:00 - 00:00	Cardiología	Ginecólogo	Editar Eliminar Reporte

4. Al dar clic en editar nos saldrá una ventana emergente que dejara editar el nombre turno área o labor



The screenshot shows a modal window titled 'Editar registro'. It contains four input fields with the following values: 'Nombre: miguel nolasco', 'Turno: 08:00 - 16:00', 'Área: Pediatría', and 'Labor: Traumatólogo'. At the bottom of the modal is a blue button labeled 'Guardar cambios'.

Link del video: <https://youtu.be/skk-VOqUrDI>

5. Al dar clic en el botón eliminar se eliminará el registro, antes eran 73 registros y ahora solo 72

CRUD Personal Hospitalario

Registros existentes

ID	Nombre	Turno	Área	Labor	Acciones
65	miguel nolasco	08:00 - 16:00	Pediatría	Traumatólogo	Editar Eliminar Reporte
66	emilio	08:00 - 16:00	Cirugía	Cirujano	Editar Eliminar Reporte
67	hector	00:00 - 08:00	Cardiología	Cardiólogo	Editar Eliminar Reporte
69	juan	08:00 - 16:00	Pediatría	Ginecologo	Editar Eliminar Reporte
70	peso pluma	16:00 - 00:00	Ginecología	Ginecologo	Editar Eliminar Reporte
71	junior h	00:00 - 08:00	Cardiología	Cardiólogo	Editar Eliminar Reporte
72	mike_adx	08:00 - 16:00	Pediatría	Pediatra	Editar Eliminar Reporte

6. Al dar clic al botón de reportes generar un pdf con todo el personal sus turnos áreas y labor.

Informe de Médicos

Nombre	Horario	Área	Labor
miguel nolasco	08:00 - 16:00	Pediatría	Traumatólogo
emilio	08:00 - 16:00	Cirugía	Cirujano
hector	00:00 - 08:00	Cardiología	Cardiólogo
juan	08:00 - 16:00	Pediatría	Ginecologo
peso pluma	16:00 - 00:00	Ginecología	Ginecologo
junior h	00:00 - 08:00	Cardiología	Cardiólogo
mike_adx	08:00 - 16:00	Pediatría	Pediatra

Link del video: <https://youtu.be/skk-VOqUrDI>