



UNIVERSIDAD BANCARIA DE MÉXICO

"Constancia Unidad y trabajo"

INGENIERÍA EN SISTEMAS COMPUTACIONALES

RECONOCIMIENTO DE VALIDEZ OFICIAL DE ESTUDIOS DE LA SECRETARÍA DE EDUCACIÓN PÚBLICA No. 2022241 DE FECHA 13 DE SEPTIEMBRE DE 2002.

**Desarrollo de Aplicaciones
para Internet Avanzado I**

MTI. Sthefany Villarreal Jiménez

Octavo Cuatrimestre

**DOCUMENTACIÓN
SITIO PROYECTO EXAMEN**

FERRUZCA RAUL

MARZO 28 2025



Índice

1. Introducción -----	5
2. Tecnologías Utilizadas -----	7
2.1 HTML (HyperText Markup Language) -----	7
2.2 PHP (Hypertext Preprocessor) -----	8
2.3 MySQL (Sistema de Gestión de Bases de Datos) -----	9
2.4 Servidor Local MAMP -----	10
2.5 Bootstrap (Framework de Diseño) -----	10
2.6 FPDF (Generación de Reportes en PDF) -----	11
2.7 Chart.js (Generación de Gráficos) -----	12
3. Manual de Usuario -----	13
3.1 Index -----	13
3.2 Login -----	15
3.3 Visitas -----	17
3.4 Administrador -----	19
3.5 Vendedor -----	59
3.6 Cliente -----	72
4. Conclusión -----	84

Índice de imágenes

Ejemplo html propia autoría	7
Ejemplo Php propia autoría	8
Ejemplo Tabla propia autoría	9
Ejemplo Bootstrap propia autoría	10
Ejemplo fpdf propia autoría	11
Ejemplo Gráfica propia autoría	12
Index propia autoría	13
Login propia autoría	15
Pagina Visitas propia autoría	17
DashboardAdmin propia autoría	19
Gestion Usuarios propia autoría	21
Agregar Usuario propia autoría	23
Editar Usuario propia autoría	25
Gestión Horarios propia autoría	27
Editar Horario propia autoría	29
Gestión Promociones propia autoría	31
Editar Promoción propia autoría	33
Gestión del Reglamento propia autoría	35
Editar Regla propia autoría	37
Gestión de Instalaciones propia autoría	39
Editar Instalación propia autoría	41
Gestión de Competencias Acuáticas propia autoría	43
Editar Competencia propia autoría	45
Gestión de Rehabilitación propia autoría	47
Editar Sesión de Rehabilitación propia autoría	49
Visitas propia autoría	52
PDF propia autoría	54
Gráfica propia autoría	56
Dashboard Vendedor propia autoría	59

Registrar Nuevo Usuario Natación propia autoría	62
Consulta de Horarios propia autoría	64
Crear Nueva Promoción propia autoría	66
Reglamento propia autoría	68
Crear Nueva Competencia	70
Dashboard Cliente propia autoría	72
Vista Horarios propia autoría	74
Inscribirse en Competencias propia autoría	76
Agendar Rehabilitación propia autoría	78
Promociones Activas propia autoría	80
Reglamento propia autoría	82

1. Introducción

En la actualidad, el uso de herramientas tecnológicas en instituciones deportivas se ha convertido en un factor clave para la mejora de la gestión, organización y atención al público. La digitalización de procesos no solo permite automatizar tareas rutinarias, sino también centralizar la información, optimizar tiempos de respuesta, mejorar la experiencia del usuario y garantizar un mayor control administrativo. En este contexto, nace el proyecto del Sistema de Gestión del Centro de Natación Innovex, una plataforma web desarrollada con tecnologías como PHP, MySQL, Bootstrap, JavaScript y FPDF, con el propósito de brindar una solución eficaz, ordenada y segura a las operaciones del centro acuático.

Este sistema fue diseñado para atender múltiples áreas funcionales del centro, permitiendo desde la administración de usuarios, horarios, competencias y promociones, hasta la gestión de sesiones de rehabilitación, visitas registradas, visualización de reglamento interno y generación de reportes y gráficas estadísticas. Cada módulo fue creado de forma modular y adaptativa, facilitando su comprensión y uso por parte de los distintos tipos de usuarios registrados en el sistema.

Una característica importante de este proyecto es su enfoque en la jerarquización de usuarios mediante niveles de acceso, permitiendo así diferenciar roles y asignar funciones específicas: el administrador posee control total del sistema, el vendedor gestiona usuarios y promociones, el cliente puede interactuar con el sistema mediante consultas, inscripciones y solicitudes, y el visitante puede acceder de forma rápida a ciertas funciones informativas. Este enfoque mejora la seguridad, evita accesos no autorizados y personaliza la experiencia según el perfil del usuario.

Asimismo, se cuidaron aspectos visuales, estructurales y funcionales del sistema para que cada módulo sea intuitivo, accesible y coherente con el entorno gráfico institucional del Centro de Natación Innovex. Las tablas dinámicas, formularios estilizados, botones con iconos y ventanas emergentes permiten una interacción moderna y amigable, además de ofrecer una experiencia fluida tanto en computadoras como en dispositivos móviles.

El presente documento expone el desarrollo completo del sistema, explicando cada uno de sus módulos, funciones, estructuras de base de datos, diseño de interfaz, flujo de usuarios y funcionalidades específicas. La finalidad es ofrecer una visión completa del proyecto, sus alcances, fundamentos técnicos y beneficios que aporta a la institución, posicionando al Centro de Natación Innovex como una organización moderna, tecnológica y enfocada en la excelencia administrativa y deportiva.

2. Tecnologías Utilizadas

En este apartado, se detallan las herramientas utilizadas para la construcción del sistema, su propósito y ejemplos prácticos de implementación.

2.1 HTML (HyperText Markup Language)

📌 Descripción

HTML (Lenguaje de Marcado de Hipertexto) es el lenguaje fundamental para la creación de páginas web. Es el encargado de definir la estructura de una página mediante etiquetas (`<html>`, `<body>`, `<table>`, `<form>`, `<button>`, etc.).

El navegador interpreta el código HTML y lo muestra de manera visual, permitiendo la interacción con el usuario.

📌 Uso en el Proyecto

- **Estructuración de la interfaz de usuario**, incluyendo formularios de pedidos, tablas de productos y reportes.
- **Integración con CSS y Bootstrap** para mejorar la presentación y el diseño responsive.
- **Definición de elementos interactivos**, como botones y formularios para la gestión de pedidos.

📌 Ejemplo de Código HTML

```
<form action="procesar_pedido.php" method="POST">
    <label>Nombre del Cliente:</label>
    <input type="text" name="cliente" required>
    <button type="submit">Registrar Pedido</button>
</form>
```

Ejemplo html propia autoría

2.2 PHP (Hypertext Preprocessor)

Descripción

PHP es un lenguaje de programación del lado del servidor que permite **generar contenido dinámico**, interactuar con bases de datos y manejar sesiones de usuario.

PHP se ejecuta en el servidor y envía la información procesada al navegador del usuario en forma de HTML.

Uso en el Proyecto

- **Procesamiento de formularios y validación de datos** enviados desde HTML.
- **Manejo de sesiones y autenticación de usuarios** mediante `$_SESSION`.
- **Conexión con MySQL** para almacenar y recuperar información.
- **Generación de reportes y exportación de datos a PDF** con la biblioteca FPDF.

Ejemplo de Código PHP

```
<?php
$servername = "localhost";
$username = "root";
$password = "root";
$database = "sistema_ventas";

$conn = new mysqli($servername, $username, $password, $database);

if ($conn->connect_error) {
    die("Error de conexión: " . $conn->connect_error);
}
?>
```

Ejemplo Php propia autoría

2.3 MySQL (Sistema de Gestión de Bases de Datos)

Descripción

MySQL es un **sistema de gestión de bases de datos relacional (RDBMS)** que permite almacenar y gestionar grandes volúmenes de datos de manera estructurada.

Funciona mediante el lenguaje SQL (**Structured Query Language**) para realizar operaciones como inserción, actualización y consulta de datos.

Uso en el Proyecto

- **Almacenamiento de información** sobre pedidos, ventas, empleados y usuarios.
- **Ejecución de consultas SQL** para mostrar datos en la aplicación.
- **Optimización de datos mediante índices** para mejorar la velocidad de respuesta.

Ejemplo de Creación de Tabla de Ventas

```
CREATE TABLE ventas (
    id INT AUTO_INCREMENT PRIMARY KEY,
    folio VARCHAR(50) UNIQUE NOT NULL,
    empleado_id INT NOT NULL,
    fecha DATETIME DEFAULT CURRENT_TIMESTAMP,
    total DECIMAL(10,2) NOT NULL,
    metodo_pago VARCHAR(50) NOT NULL,
    estado ENUM('Pagado', 'Cancelado') DEFAULT 'Pagado'
);
```

Ejemplo Tabla propia autoría

2.4 Servidor Local MAMP

📌 Descripción

MAMP (Mac Apache MySQL PHP) es un entorno de desarrollo que permite ejecutar aplicaciones web **sin necesidad de un servidor en línea**.

Incluye **Apache (servidor web)**, **MySQL (base de datos)** y **PHP (procesamiento del servidor)**.

📌 Uso en el Proyecto

- **Ejecución local de archivos PHP.**
- **Administración de la base de datos MySQL** mediante phpMyAdmin.
- **Simulación de un servidor en localhost** para pruebas antes de la implementación.

📌 Pasos para Configurar MAMP

1. Descargar MAMP desde <https://www.mamp.info>.
2. Instalar MAMP y abrir la aplicación.
3. Iniciar Apache y MySQL desde la interfaz de MAMP.
4. Acceder a `http://localhost:8888` en el navegador para probar el servidor.

2.5 Bootstrap (Framework de Diseño)

📌 Descripción

Bootstrap es un **framework CSS** que facilita la creación de sitios web **responsivos y atractivos**, sin necesidad de escribir código CSS manualmente.

📌 Uso en el Proyecto

- **Diseño de formularios, botones y tablas** con estilos predefinidos.
- **Creación de una interfaz visualmente atractiva.**
- **Compatibilidad con dispositivos móviles y pantallas grandes.**

📌 Ejemplo de Código con Bootstrap

```
<button class="btn btn-primary">Registrar Pedido</button>
```

Ejemplo Bootstrap propia autoría

2.6 FPDF (Generación de Reportes en PDF)

📌 Descripción

FPDF es una **biblioteca PHP** que permite generar archivos PDF de forma programática.

Permite la creación de reportes exportables con tablas y gráficos.

📌 Uso en el Proyecto

- **Generación de reportes de ventas** en formato PDF.
- **Exportación de información de la base de datos** en un archivo descargable.
- **Creación de documentos oficiales** con formato predefinido.

📌 Ejemplo de Generación de PDF

```
require '../fpdf/fpdf.php';
$pdf = new FPDF();
$pdf->AddPage();
$pdf->SetFont('Arial', 'B', 16);
$pdf->Cell(40, 10, 'Reporte de Ventas');
$pdf->Output();
```

Ejemplo fpdf propia autoría

2.7 Chart.js (Generación de Gráficos)

📌 Descripción

Chart.js es una **biblioteca de JavaScript** para la creación de gráficos dinámicos.

📌 Uso en el Proyecto

- **Visualización de estadísticas de ventas** en tiempo real.
- **Gráficos de barras y líneas** con datos de la base de datos.

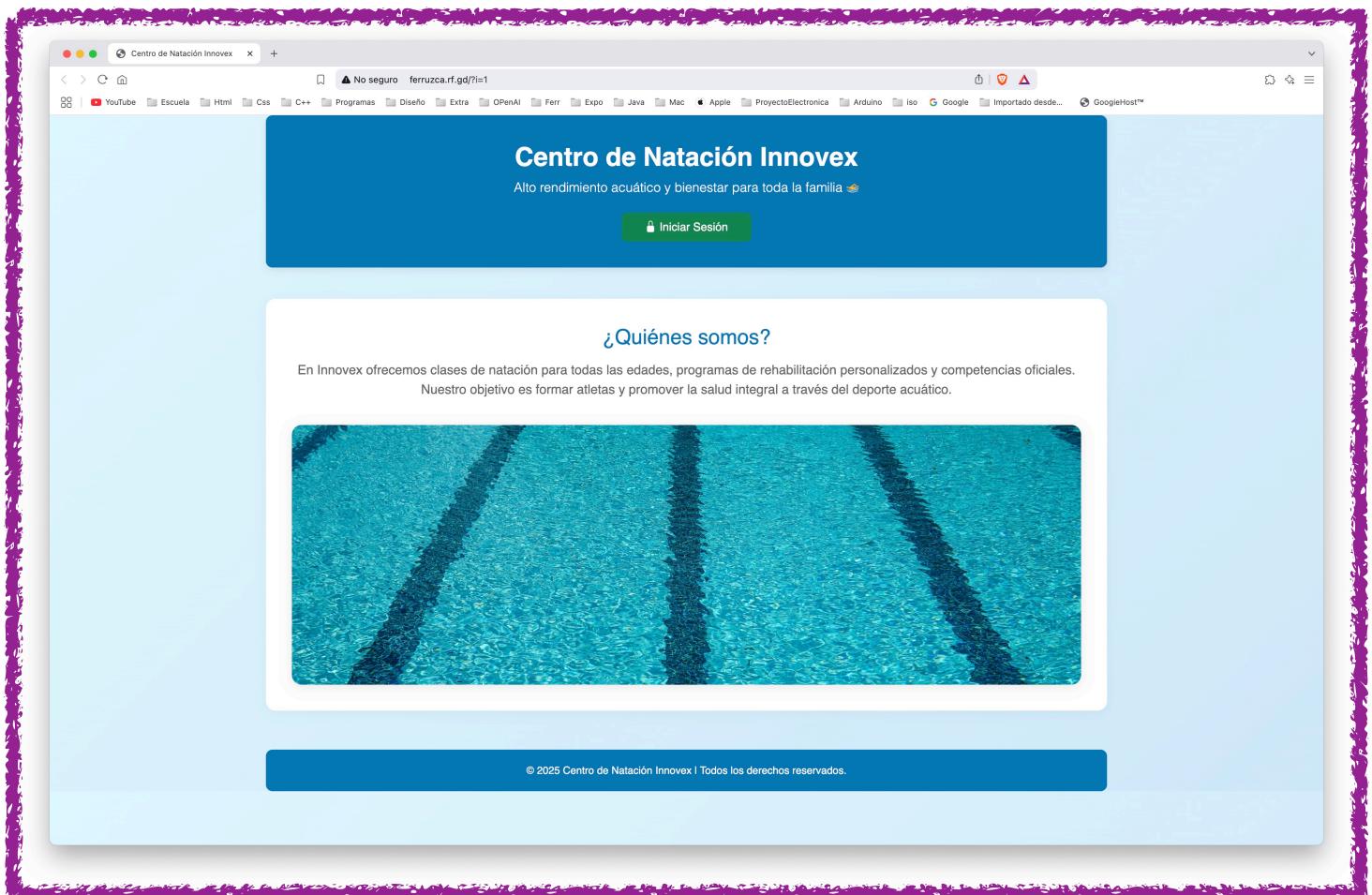
📌 Ejemplo de Código

```
<canvas id="graficoVentas"></canvas>
<script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
```

Ejemplo Gráfica propia autoría

3. Manual de Usuario

3.1 Index



Index propia autoría

Distribución Visual

- **Encabezado:** Contiene el nombre del centro "**Centro de Natación Innovex**" con un subtítulo que refuerza la misión del centro.
- **Botón "Iniciar Sesión":** Centrado y estilizado en verde con Bootstrap (`btn btn-success`), destinado a redirigir al formulario de login.
- **Sección "¿Quiénes somos?":** Explica brevemente la filosofía y servicios del centro, acompañada de una imagen de una piscina como fondo visual descriptivo.
- **Imagen central:** Representación de un carril de alberca olímpica que da identidad visual al sitio.
- **Pie de página:** Contiene derechos reservados con el año dinámico (© 2025).



Funcionalidad

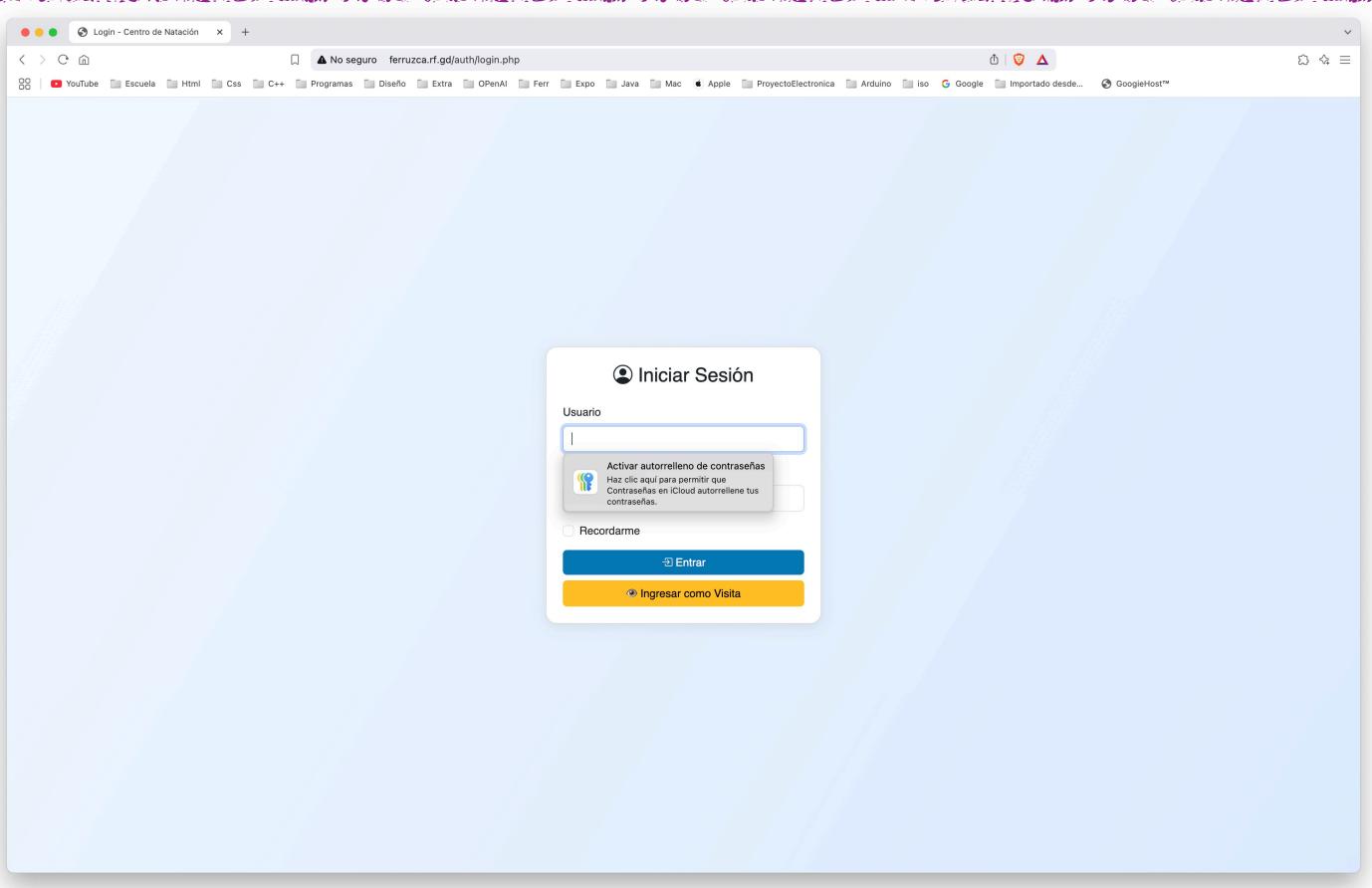
- Al cargar esta página, el usuario no necesita estar autenticado.
- El botón "Iniciar Sesión" redirige a `auth/login.php`.
- Esta vista se construye en HTML con diseño adaptado usando **Bootstrap 5** y colores suaves que evocan un entorno acuático.
- El contenido es completamente público, pensado para presentación institucional del sitio.



Tres Datos Importantes

1.  **Lenguajes y tecnologías usadas:** HTML5, PHP (para pie de página dinámico y rutas), CSS (vía Bootstrap), y responsive design adaptado para móviles.
2.  **Seguridad desde la raíz:** Aunque esta vista es pública, el sistema utiliza `session_start()` en páginas restringidas, separando acceso público del privado desde el login.
3.  **Modularidad y reutilización:** Este archivo se encuentra separado del sistema principal en la raíz o carpeta `/public` y se integra con las rutas como `auth/login.php`, lo que demuestra una arquitectura organizada y modular del proyecto.

3.2 Login



Login propia autoría

Pantalla de Login - Acceso Seguro al Sistema

Distribución Visual

- **Título "Iniciar Sesión"** centrado en la parte superior del formulario, con ícono de usuario.
- **Campo de Usuario** con auto-completado y soporte para gestores de contraseñas (como iCloud Keychain).
- **Campo de Contraseña** oculto (no visible en la imagen, pero programado).
- **Checkbox "Recordarme"** para mantener la sesión iniciada si el usuario lo desea.
- **Botón "Entrar"** con ícono de login y diseño en Bootstrap (`btn btn-primary`).
- **Botón "Ingresar como Visita"** en color amarillo con ícono, que permite acceder directamente a la sección pública sin autenticación.



Funcionalidad

- Al hacer clic en “**Entrar**”, se envían los datos mediante **POST** al archivo **validar_login.php** para su validación con la base de datos.
- Si se selecciona “**Ingresar como Visita**”, se redirige directamente a **visita/visita.php**, **ocultando** el formulario gracias a una validación JavaScript o PHP previa.
- El sistema distingue entre niveles de usuario mediante **session_start()** y variables de sesión como **\$_SESSION['nivel']**.
- Los datos de login se validan contra la tabla **usuarios** usando conexión PDO (según mejoras del proyecto).



Tres Datos Importantes

1.  **Programación PHP + Seguridad:** Validación del usuario mediante consulta PDO con contraseña encriptada usando **password_verify()** para mayor seguridad.
2.  **Control de sesión y roles:** Al iniciar sesión correctamente, se establece la sesión con **\$_SESSION['nombre'], \$_SESSION['nivel']**, y se redirige dinámicamente según el rol (admin, gerente, vendedor, etc.).
3.  **Validación de visitas:** Si se accede como “Visita”, se omite completamente el uso de la tabla **usuarios** y se accede a una vista restringida sin validaciones, ideal para marketing o pruebas públicas.

3.3 Visitas

The screenshot shows a web browser window with the following content:

- Header:** "Centro de Natación Innovex" and "Consulta general para visitantes".
- Section 1 (Instalaciones):** Contains three images: an indoor swimming pool, a swimmer underwater, and a person in a locker room.
- Section 2 (Horarios):** Shows operating hours: Lunes a Viernes: 6:00 AM - 9:00 PM; Sábado: 7:00 AM - 3:00 PM; Domingo: Cerrado.
- Section 3 (Reglamento):** A numbered list of five rules:
 - El uso de gorra es obligatorio en todo momento.
 - Está prohibido correr en las áreas húmedas.
 - Respetar las indicaciones de los instructores y personal del centro.
 - No se permite el ingreso con alimentos o bebidas.
 - Respetar el aforo máximo en cada área.
- Footer:** A yellow button labeled "← Volver al Inicio" (Back to Home).

Página Visitas propia autoría

Distribución Visual General

- **Encabezado principal:** "Centro de Natación Innovex" con subtítulo "**Consulta general para visitantes**".
- **Sección de Instalaciones:** Contiene tres imágenes que representan distintos espacios del centro (alberca techada, nadador submarino y vestidores).
- **Sección de Horarios:** Tabla visual con días y horarios de operación. Domingo aparece como cerrado.
- **Reglamento:** Lista numerada con las normas básicas de conducta en el centro.
- **Botón "Volver al Inicio":** Botón amarillo tipo Bootstrap (`btn btn-warning`) con ícono, que permite regresar a la página principal.



Funcionalidad

- Esta vista está disponible **solo si el usuario selecciona "Ingresar como Visita"** desde el login, sin necesidad de sesión activa.
- No accede a la base de datos, es una vista **estática pero profesional**, ideal para brindar información general al público.
- Usa **Bootstrap para estructura y estilo**, especialmente en cards, tablas y botones.
- Todas las imágenes tienen diseño responsive (**img-fluid, rounded**), para una experiencia limpia en móvil y escritorio.



Tres Datos Importantes de Programación

 **Ruta controlada:** El archivo se encuentra en **visita/visita.php** y es accesible directamente únicamente si se accede como visitante. Desde el login, se usa una condición tipo:

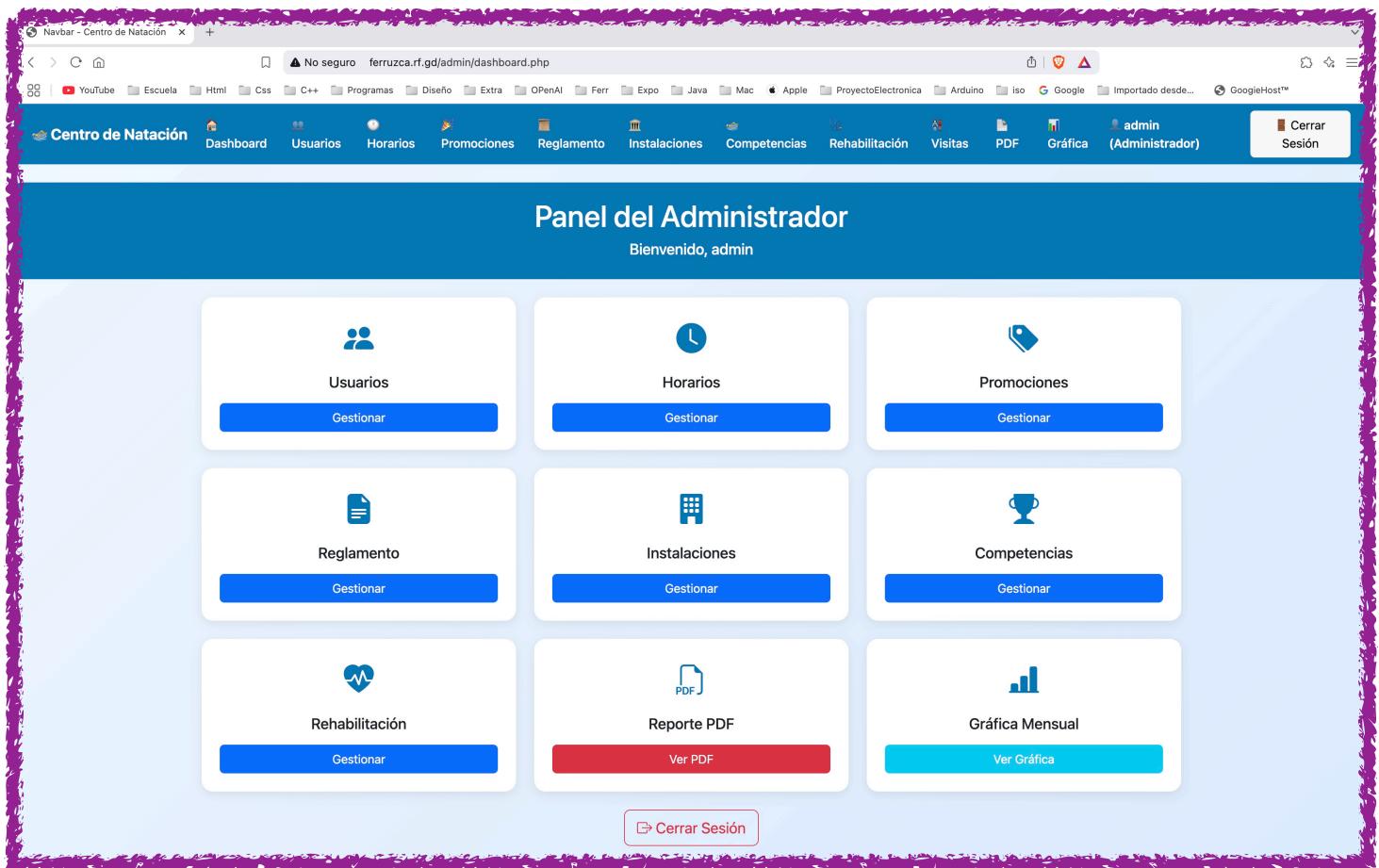
```
if ($_POST['tipo_usuario'] === 'visita') {
    header("Location: visita/visita.php");
}
```

 **Modularidad con includes:** Aunque esta vista es pública, puede compartir estilos o navbar con otras páginas a través de:

```
include('../includes/navbar.php');
```

 **Diseño sin base de datos:** Esta vista no requiere consultas SQL, pero se mantiene dentro del ecosistema del sistema principal, permitiendo futuras extensiones como **formularios de contacto**, promociones dinámicas o mapas integrados.

3.4 Administrador



DashboardAdmin propia autoría

💻 Panel del Administrador - Control Total del Sistema

🧭 Distribución Visual

- **🌐 Encabezado** con el título "**Panel del Administrador**" y mensaje de bienvenida personalizado: "*Bienvenido, admin*".
- **📍 Menú de navegación superior (`navbar_admin.php`)** con acceso rápido a todas las secciones del sistema:
 - Dashboard, Usuarios, Horarios, Promociones, Reglamento, Instalaciones, Competencias, Rehabilitación, Visitas, PDF y Gráfica.
- **🧩 Tarjetas de gestión:** Distribuidas en una cuadrícula visual (probablemente con `row` y `col-md-4` de Bootstrap) para acceder a cada módulo.
- **📄 Botón de PDF y 📈 Gráfica mensual** están destacados con colores distintos para enfatizar su función de reporte.
- **🔴 Botón "Cerrar Sesión"** estilizado en rojo para indicar una acción de cierre de seguridad.

Funcionalidad

- Cada botón "Gestionar" redirige a un CRUD correspondiente, donde el administrador puede realizar acciones de **alta, baja lógica, modificación y consulta**.
- El sistema valida el **nivel del usuario (nivel = 1)** para restringir este panel exclusivamente a administradores.
- Uso de **íconos SVG o FontAwesome** para facilitar la comprensión visual del contenido.
- Este **dashboard.php** es el **centro de control** del sistema. Está construido con estructura modular y rutas organizadas por carpeta (**admin/usuarios/**, **admin/promociones/**, etc.).

Tres Datos Importantes de Programación

Control de acceso con sesión y nivel:

```
session_start();

if ($_SESSION['nivel'] != 1) {
    header("Location: ../auth/login.php");
    exit();
}
```

Asegura que solo los administradores vean esta vista.

 **Organización de rutas:** Cada módulo apunta a un archivo o subcarpeta específica, por ejemplo:

- **usuarios/gestionar.php**
- **pdf/ventas_pdf.php**
- **graficas/ventas_grafico.php**

Integración de reportes con FPDF y gráficas:

- El botón "Ver PDF" abre un reporte generado con **FPDF**.
- El botón "Ver Gráfica" carga visualizaciones dinámicas, posiblemente con **Chart.js** o gráficos generados desde PHP con datos SQL.

Rol	Usuario	Password
Administrador	admin	admin123

The screenshot shows a web application interface for managing users. At the top, there's a navigation bar with links like 'Dashboard', 'Usuarios', 'Horarios', etc. Below the navigation is a title 'Gestión de Usuarios'. Underneath the title are two buttons: 'Volver al Panel' and 'Agregar Usuario'. The main content is a table with the following data:

ID	Nombre	Usuario	Nivel	Estado	Acciones
1	Raúl Ferruzca	admin	Administrador	Activo	
2	Juan Pérez	vendedor1	Vendedor	Activo	
3	Maria López	cliente1	Cliente	Activo	

Gestion Usuarios propia autoría

Distribución Visual

- **Título centrado:** "Gestión de Usuarios" dentro del contenido principal de la página.
- **Botón “Volver al Panel”** y **Botón “Agregar Usuario”**, ambos estilizados con Bootstrap (`btn btn-outline-primary` y `btn btn-primary`) ubicados en la parte superior de la tabla.
- **Tabla de usuarios:** Presenta columnas ordenadas por:
 - ID
 - Nombre completo
 - Usuario
 - Nivel (rol del sistema)
 - Estado (activo/inactivo)
 - Acciones (editar / eliminar)
- Etiquetas verdes de estado con texto “Activo” (`badge bg-success`).
- ícono de edición (lápiz) y ícono de eliminación (cruz roja) en cada fila.

Funcionalidad

- Al cargar esta vista, se ejecuta una consulta a la base de datos **usuarios** usando **PDO** para listar todos los registros activos.
- El botón “**Agregar Usuario**” redirige a un formulario (**agregar_usuario.php**) donde se registra un nuevo usuario.
- El botón de lápiz permite **modificar nombre, usuario, contraseña y nivel**.
- El botón de eliminar realiza una **baja lógica**, cambiando el estado del usuario a “**Inactivo**” (no se elimina físicamente).

Esta vista solo está disponible si el usuario tiene nivel 1 (administrador), con control de sesión:

```
if ($_SESSION['nivel'] != 1) {
    header("Location: ../auth/login.php");
}
```

Tres Datos Importantes de Programación

Validación y seguridad:

El archivo **gestionar_usuarios.php** restringe el acceso por medio de sesiones.

Las acciones de edición y eliminación incluyen validaciones contra inyecciones SQL.

Consulta de usuarios activos:

```
$stmt = $pdo->prepare("SELECT * FROM usuarios WHERE estado = 1");
$stmt->execute();
$usuarios = $stmt->fetchAll();
```

Solo se muestran los usuarios activos en esta tabla.

Edición y eliminación modular:

- Los botones enlazan con archivos como **editar_usuario.php?id=3** o **eliminar_usuario.php?id=3**.
- La eliminación no borra el registro, solo actualiza el campo **estado = 0** para mantener la integridad de los datos históricos.

The screenshot shows a web application interface for user registration. The main title is 'Registrar Nuevo Usuario'. The form includes fields for 'Nombre completo', 'Usuario', 'Contraseña', and 'Nivel de acceso' (a dropdown menu). There are two buttons at the bottom: 'Volver' (Back) and 'Registrar' (Register).

Agregar Usuario propia autoría



Registrar Nuevo Usuario - Alta de Cuentas en el Sistema



Distribución Visual

- **◆ Encabezado de formulario** con fondo azul y texto blanco: “Registrar Nuevo Usuario”.
- **📄 Campos del formulario** alineados verticalmente:
 - Nombre completo
 - Usuario
 - Contraseña
 - Nivel de acceso (select desplegable)
- **⬅ Botón “Volver”** para regresar sin guardar cambios.
- **✓ Botón “Registrar”** de color verde (`btn btn-success`), al final del formulario.



Funcionalidad

- Este formulario permite al administrador crear nuevos usuarios que se insertarán en la tabla **usuarios**.
- El **nivel de acceso** define el rol:
 - 1 → Administrador
 - 2 → Vendedor
 - 3 → Cliente
 - (Puede incluir también: 4 → Gerente)
- Al hacer clic en **Registrar**, los datos se envían mediante POST a un archivo como **insertar_usuario.php** o son procesados directamente en este archivo.
- El campo de contraseña se cifra usando **password_hash()** antes de insertarse en la base de datos.

Tres Datos Importantes de Programación

Hashing de contraseña seguro en PHP:

```
$password_hashed = password_hash($_POST['password'],
PASSWORD_DEFAULT);
```

Esto garantiza que la contraseña no se almacena en texto plano en la base de datos.

Inserción con PDO preparada:

```
$stmt = $pdo->prepare("INSERT INTO usuarios (nombre, usuario,
password, nivel, estado) VALUES (?, ?, ?, ?, 1)");
$stmt->execute([$nombre, $usuario, $password_hashed, $nivel]);
```

Esta sintaxis protege contra inyecciones SQL y define estado activo por defecto.

Validaciones obligatorias antes de insertar:

- Validación de campos vacíos.
- Verificación de unicidad del usuario con una consulta SELECT previa.
- Control de errores con mensajes tipo alerta (**alert alert-danger / alert alert-success**).

Editar Usuario propia autoría



Editar Usuario - Actualización de Datos del Sistema



Distribución Visual

- **Título principal:** “Editar Usuario” con ícono de lápiz.
- **Formulario** centrado, con estilo limpio y campos bien definidos:
 - Campo de texto para **Nombre completo**
 - Menú desplegable para seleccionar el **Nivel** del usuario (Administrador, Vendedor, Cliente...)
- Botón “Actualizar” (`btn btn-primary`) para guardar los cambios.
- Botón gris “Volver” para regresar sin modificar datos.

 **Funcionalidad**

- Esta interfaz permite editar información específica de un usuario existente. En este caso, el sistema carga los datos desde la base usando el `id` recibido por **GET**:

```
editar_usuario.php?id=1
```

- Al cargar la página, se consulta el usuario correspondiente con una sentencia **SQL SELECT**.
- Al hacer clic en **Actualizar**, el sistema valida los campos y actualiza los registros mediante un **UPDATE**.

 **Tres Datos Importantes de Programación**
 **Consulta segura con PDO:**

```
$stmt = $pdo->prepare("SELECT * FROM usuarios WHERE id = ?");

$stmt->execute([$GET['id']]);
$usuario = $stmt->fetch();
```

Esto carga los datos existentes de forma segura.

 **Actualización con validación:**

```
$stmt = $pdo->prepare("UPDATE usuarios SET nombre = ?, nivel = ?
WHERE id = ?");

$stmt->execute([$nombre, $nivel, $id]);
```

Esta consulta se ejecuta tras validar que los campos no estén vacíos.

 **Modularidad en control de roles:** El menú desplegable permite seleccionar entre niveles de usuario definidos por el sistema:

- 1 = Administrador
- 2 = Vendedor
- 3 = Cliente
- 4 = Gerente (opcional en el sistema actual)

The screenshot shows a web-based administration system for a swimming center. The top navigation bar includes links for YouTube, Escuela, Html, Css, C++, Programas, Diseño, Extra, OpenAI, Ferr, Expo, Java, Mac, Apple, ProyectoElectronica, Arduino, iso, Google, Importado desde..., and InfinityFree. The main menu has items like Centro de Natación, Dashboard, Usuarios, Horarios, Promociones, Reglamento, Instalaciones, Competencias, Rehabilitación, Visitas, PDF, Gráfica, and an Admin (Administrador) section where 'admin' is logged in.

The central content area is titled 'Gestión de Horarios' (Schedule Management). It features a 'Agregar Nuevo Horario' (Add New Schedule) form with fields for 'Día' (Day), 'Hora Inicio' (Start Time), 'Hora Fin' (End Time), and 'Nombre de la actividad' (Activity Name), along with a green 'Agregar' (Add) button. Below this is a table titled 'Horarios Registrados' (Registered Schedules) showing the following data:

Día	Hora Inicio	Hora Fin	Actividad	Acciones
Lunes	08:00:00	09:00:00	Natación básica	
Martes	10:00:00	11:00:00	Rehabilitación física	
Miércoles	09:00:00	10:00:00	Competencia Juvenil	

Gestión Horarios propia autoría

JUL 17 Gestión de Horarios - Control y Administración de Actividades

Distribución Visual

- **Encabezado principal:** “Gestión de Horarios” al centro de la vista.
- **Formulario para agregar nuevo horario:**
 - Campo de día (texto libre: ej. Lunes)
 - Campo de hora de inicio (`type="time"`)
 - Campo de hora de fin (`type="time"`)
 - Campo de nombre de la actividad
 - Botón verde “Agregar”
- **BACK** Botón de retorno a panel principal con estilo `btn-outline-primary`.
- **Tabla de horarios registrados:**
 - Columnas: Día, Hora Inicio, Hora Fin, Actividad, Acciones.
 - Acciones con íconos: Editar, Eliminar.

 **Funcionalidad**

- Este módulo permite al administrador **crear, consultar, editar y eliminar horarios** de actividades como natación, rehabilitación y competencias.
- Al presionar el botón "Aregar", el sistema guarda los datos en la tabla **horarios** mediante una sentencia **INSERT con PDO**.
- Los botones de editar y eliminar permiten modificar los horarios o eliminarlos lógicamente de la base.
- Al eliminar un horario se aplica una **eliminación lógica**, manteniéndolo oculto de la vista sin borrarlo físicamente.
- Este archivo probablemente se llama **gestionar_horarios.php** y está ubicado en la carpeta **admin/**.

 **Tres Datos Importantes de Programación**
 **Inserción en base de datos con PDO:**

```
$stmt = $pdo->prepare("INSERT INTO horarios (dia, hora_inicio, hora_fin, actividad) VALUES (?, ?, ?, ?)");
$stmt->execute([$dia, $hora_inicio, $hora_fin, $actividad]);
```

Garantiza seguridad y evita SQL Injection.

 **Validación de campos vacíos y de hora** antes de insertar:

```
if (empty($dia) || empty($hora_inicio) || empty($hora_fin) || empty($actividad)) {
    echo "<div class='alert alert-danger'>Todos los campos son obligatorios</div>";
}
```

 **Separación lógica de controladores:** Las funciones para editar o eliminar registros probablemente están en archivos separados como:

- **editar_horario.php?id=3**
- **eliminar_horario.php?id=3** Esto permite mantener limpio el código y una buena arquitectura MVC.

Editar Horario propia autoría

⌚ Editar Horario - Modificación de Actividades Programadas

🧭 Distribución Visual

- **✓ Encabezado principal:** “Editar Horario” con ícono de check-box
- **📋 Formulario de edición centrado**, con los siguientes campos:
 - Día (ej. Lunes)
 - Hora de inicio (`type="time"`)
 - Hora de fin (`type="time"`)
 - Actividad (texto libre)
- **● Botón “Guardar Cambios”** (`btn btn-primary`) para confirmar la edición.
- **● Botón “Volver”** en gris para regresar al panel anterior sin hacer cambios.

 **Funcionalidad**

- Al acceder a esta vista, el **ID** del horario se pasa por la URL:

editar_horario.php?id=1

- El sistema obtiene los datos con una consulta **SELECT** usando **PDO** y los carga en los campos del formulario.
- Al hacer clic en **Guardar Cambios**, se actualiza el registro en la tabla **horarios** mediante una sentencia **UPDATE**.

 **Tres Datos Importantes de Programación**
 **Carga previa del horario:**

```
$stmt = $pdo->prepare("SELECT * FROM horarios WHERE id = ?");

$stmt->execute([$GET['id']]);
$horario = $stmt->fetch();
```

Esto permite mostrar los datos actuales en el formulario.

 **Actualización protegida:**

```
$stmt = $pdo->prepare("UPDATE horarios SET dia = ?, hora_inicio = ?, hora_fin = ?, actividad = ? WHERE id = ?");

$stmt->execute([$dia, $hora_inicio, $hora_fin, $actividad, $id]);
```

Se asegura que solo se modifiquen los campos necesarios.

 **Validación previa:**

- No se permite guardar si algún campo está vacío.
- Se valida que **la hora de inicio sea menor que la hora de fin**, para evitar conflictos.

The screenshot shows a web application interface for managing promotions. At the top, there's a navigation bar with links like 'Dashboard', 'Usuarios', 'Horarios', 'Promociones', 'Reglamento', 'Instalaciones', 'Competencias', 'Rehabilitación', 'Visitas', 'PDF', 'Gráfica', and 'admin (Administrador)'. A 'Cerrar Sesión' button is also present.

The main content area has a title 'Gestión de Promociones'. Below it is a blue header for 'Agregar Nueva Promoción' containing fields for 'Título de la promoción' (Title of the promotion), 'Descripción' (Description), 'Fecha de inicio:' (Start date) with a dd/mm/aaaa format, and 'Fecha de fin:' (End date) with a dd/mm/aaaa format. There's also a green 'Agregar' (Add) button.

Below this is a section titled 'Promociones Registradas' (Registered Promotions) with a table:

Título	Descripción	Inicio	Fin	Acciones
Combo Familiar	Inscripción para 3 con precio especial.	2024-05-01	2025-05-31	
Mes del Niño	Primera clase gratis para menores de 12 años.	2024-04-15	2024-05-15	
Descuento Primavera	20% de descuento en inscripción.	2024-04-01	2024-04-30	

Gestión Promociones propia autoría

Gestión de Promociones - Control de Descuentos y Ofertas

Distribución Visual

- Encabezado principal centrado: “Gestión de Promociones”.
- Formulario para agregar nueva promoción:
 - Campo: Título de la promoción
 - Campo: Descripción (multilínea)
 - Campo: Fecha de inicio (`type="date"`)
 - Campo: Fecha de fin (`type="date"`)
 - Botón verde "Agregar"
- Botón “Volver al Panel” ubicado arriba del formulario, estilo `btn-outline-primary`.
- Tabla de promociones registradas:
 - Columnas: Título, Descripción, Inicio, Fin, Acciones.
 - Íconos de edición () y eliminación () en cada fila para modificar o eliminar promociones.

Funcionalidad

- Esta vista permite al administrador registrar, consultar, editar y eliminar promociones que se mostrarán públicamente.
- El botón **Agregar** guarda la promoción con validaciones previas (que ningún campo esté vacío y que la fecha de fin sea posterior a la de inicio).
- Las promociones se guardan en la tabla **promociones** y se muestran en tiempo real en la tabla inferior.
- Las acciones de edición y eliminación están ligadas a **editar_promocion.php?id=...** y **eliminar_promocion.php?id=...** respectivamente.
- El diseño responsive y la tabla bien estructurada permiten una fácil navegación.

Tres Datos Importantes de Programación

Validación de fechas y campos:

```
if (empty($titulo) || empty($descripcion) || empty($fecha_inicio) || empty($fecha_fin)) {
    echo "<div class='alert alert-danger'>Todos los campos son obligatorios.</div>";
} elseif ($fecha_inicio > $fecha_fin) {
    echo "<div class='alert alert-warning'>La fecha de fin debe ser posterior a la de inicio.</div>";
}
```

Inserción en la base de datos con PDO:

```
$stmt = $pdo->prepare("INSERT INTO promociones (titulo, descripcion, fecha_inicio, fecha_fin) VALUES (?, ?, ?, ?, ?)");
$stmt->execute([$titulo, $descripcion, $fecha_inicio, $fecha_fin]);
```

Arquitectura limpia con modularidad:

- **gestionar_promociones.php**: visualización y formulario.
- **editar_promocion.php**: actualización de promociones.
- **eliminar_promocion.php**: eliminación lógica o física.
- Uso de íconos e interfaz tipo card para mejorar UX.

Editar Promoción propia autoría

Editar Promoción - Modificación de Ofertas del Centro

Distribución Visual

- **Título principal:** “Editar Promoción” con ícono de etiqueta, centrado en la parte superior del formulario.
- **Formulario de edición** con los siguientes campos:
 - Título de la promoción (campo de texto)
 - Descripción (campo multilinea)
 - Fecha de inicio (`type="date"`)
 - Fecha de fin (`type="date"`)
- **Botón "Guardar Cambios"** para confirmar la modificación.
- **Botón "Volver"** para regresar sin modificar.

 **Funcionalidad**

- Esta vista permite **editar promociones existentes** mediante el **ID** recibido por **GET**:

`editar_promocion.php?id=3`

- Al cargar, se ejecuta una consulta **SELECT** para precargar los datos en el formulario.
- Al presionar **Guardar Cambios**, el sistema actualiza los campos en la tabla **promociones** usando una sentencia **UPDATE**.
- Se aplican validaciones para que ningún campo quede vacío y la fecha final no sea anterior a la inicial.

 **Tres Datos Importantes de Programación**
 **Consulta inicial del registro:**

```
$stmt = $pdo->prepare("SELECT * FROM promociones WHERE id = ?");

$stmt->execute([$GET['id']]);
$promo = $stmt->fetch();
```

Esto permite mostrar los datos actuales antes de editarlos.

 **Actualización con validación:**

```
$stmt = $pdo->prepare("UPDATE promociones SET titulo = ?, descripcion = ?, fecha_inicio = ?, fecha_fin = ? WHERE id = ?");

$stmt->execute([$titulo, $descripcion, $fecha_inicio, $fecha_fin, $id]);
```

Se usa **PDO** para mayor seguridad y claridad.

 **Validación lógica de fechas y campos:**

- Se valida que la descripción no supere cierto número de caracteres.
- Que la fecha de inicio no sea posterior a la fecha final.
- Que el título no esté repetido si se maneja clave única.

The screenshot shows a web browser window with the URL ferruzca.rf.gd/admin/gestionar_reglamento.php. The page title is "Gestión del Reglamento". The top navigation bar has links for Dashboard, Usuarios, Horarios, Promociones, Reglamento (the current page), Instalaciones, Competencias, Rehabilitación, Visitas, PDF, Gráfica, and a user account for "admin (Administrador)". A "Cerrar Sesión" (Logout) button is also visible.

Agregar Nueva Regla

Escribe la nueva regla...

+ Agregar Regla

Reglas Registradas

Regla	Acciones
Respetar los horarios asignados.	
Uso obligatorio de gorra y sandalias.	
Prohibido correr en el área de alberca.	

Gestión del Reglamento propia autoría

Gestión del Reglamento - Normas del Centro de Natación

Distribución Visual

- **Encabezado principal:** “Gestión del Reglamento”, centrado en la parte superior.
- **Formulario para agregar nueva regla:**
 - Campo de texto para escribir la regla (multilínea).
 - Botón verde “**Agregar Regla**”.
- Botón "Volver al Panel" en estilo **btn-outline-primary**.
- **Tabla inferior con las reglas registradas:**
 - Columnas: Regla | Acciones.
 - Botones de edición () y eliminación () en cada fila.
 - Estilo visual limpio y amigable, usando Bootstrap para tablas y botones.

Funcionalidad

- El módulo permite **crear, editar y eliminar reglas** que rigen el comportamiento en el centro de natación.
- Las reglas se almacenan en la tabla **reglamento**, y pueden modificarse en tiempo real.
- Al hacer clic en **Agregar Regla**, el texto se inserta mediante una consulta **INSERT**.
- La tabla muestra las reglas registradas, junto con opciones para editar o eliminar usando el **ID** de cada regla.
- La vista es accesible solo para administradores (**nivel = 1**) y forma parte de **admin/gestionar_reglamento.php**.

Tres Datos Importantes de Programación

Inserción segura en base de datos:

```
$stmt = $pdo->prepare("INSERT INTO reglamento (regla) VALUES (?)");
$stmt->execute([$regla]);
```

Esto protege contra inyección SQL y mantiene los datos ordenados.

Edición modular:

- Al presionar el ícono de lápiz, se accede a **editar_regla.php?id=3**, donde se carga el contenido para su modificación.
- El botón de eliminar usa **eliminar_regla.php?id=3**, y puede aplicar eliminación lógica o física según configuración.

Validaciones previas:

- No se permite insertar campos vacíos.
- Se evita duplicar reglas ya existentes con una consulta **SELECT** previa o restricción en la base:
 $\text{UNIQUE}(\text{regla})$

The screenshot shows a web browser window with the URL ferruzca.rf.gd/admin/editar_regla.php?id=1. The page title is "Centro de Natación". The main content is a modal dialog titled "Editar Regla" (Edit Rule). Inside the dialog, there is a text area labeled "Texto de la Regla" (Text of the Rule) containing the text "Respetar los horarios asignados." (Respect the assigned schedules.). Below the text area are two buttons: a blue "Guardar Cambios" (Save Changes) button with a checkmark icon, and a grey "Volver" (Return) button.

Editar Regla propia autoría



Editar Regla - Modificación de Normas



Distribución Visual

-  **Encabezado centrado** con ícono de documento: “Editar Regla”.
-  **Campo de texto multilinea** que carga el contenido actual de la regla seleccionada.
-  **Botón “Guardar Cambios”** para confirmar la modificación.
-  **Botón “Volver”** para regresar sin modificar.

 **Funcionalidad**

- Al ingresar a esta vista, el **ID** de la regla se pasa mediante la URL:

`editar_regle.php?id=1`

- Se ejecuta una consulta **SELECT** con **PDO** para obtener la regla y mostrarla en el **textarea**.
- Al hacer clic en **Guardar Cambios**, se actualiza la base de datos con un **UPDATE**.
- Las reglas actualizadas se reflejan en la vista principal del reglamento sin necesidad de recargar manualmente.

 **Tres Datos Importantes de Programación**
 **Consulta para precargar datos:**

```
$stmt = $pdo->prepare("SELECT * FROM reglamento WHERE id = ?");

$stmt->execute([$_GET['id']]);
$regla = $stmt->fetch();
```

Muestra automáticamente el contenido existente para su edición.

 **Actualización en la base de datos:**

```
$stmt = $pdo->prepare("UPDATE reglamento SET regla = ? WHERE id = ?");

$stmt->execute([$nueva_regle, $id]);
```

Se asegura la persistencia y consistencia del dato editado.

 **Validaciones previas:**

- No se permite guardar si el campo está vacío.
- Se puede evitar duplicados consultando si el nuevo texto ya existe antes del UPDATE.

The screenshot shows a web browser window with the URL ferruzca.rf.gd/admin/gestionar_instalaciones.php. The page title is "Gestión de Instalaciones". At the top, there is a blue header bar with the text "Agregar Nueva Instalación". Below it, there are input fields for "Nombre de la instalación" and "Descripción...", and a file upload section for "Imagen" with a "Seleccionar archivo" button. A green "Registrar" button is at the bottom. Below this, a table titled "Instalaciones Registradas" lists three entries:

Nombre	Descripción	Imagen	Acciones
Alberca Olímpica	Piscina profesional de 50 metros con calefacción.		
Gimnasio Acuático	Área equipada con maquinaria de entrenamiento acuático.		
Área de Rehabilitación	Zona especializada para fisioterapia en el agua.		

Gestión de Instalaciones propia autoría

Gestión de Instalaciones - Control Visual del Espacio Físico

Distribución Visual

- **Encabezado principal:** “Gestión de Instalaciones”.
- **Formulario para agregar nueva instalación:**
 - Campo: Nombre de la instalación
 - Campo: Descripción
 - Campo: Carga de imagen (`input type="file"`)
 - Botón verde “Registrar”
- **Botón “Volver al Panel”** con estilo `btn-outline-primary`.
- **Tabla de instalaciones registradas:**
 - Columnas: Nombre, Descripción, Imagen, Acciones
 - Cada fila muestra una instalación con su imagen, descripción y botones (editar) y (eliminar).

 **Funcionalidad**

- Este módulo permite **crear, editar y eliminar** instalaciones del centro de forma visual y con imágenes asociadas.
- La imagen se sube al servidor y se asocia al registro mediante el campo **imagen** en la base de datos.
- Se valida que todos los campos estén completos antes de registrar.
- Las imágenes se almacenan probablemente en una carpeta como **/uploads/instalaciones/** y se insertan como **VARCHAR** en la tabla.

 **Tres Datos Importantes de Programación**
 **Carga y almacenamiento de imágenes:**

```
$imagen = $_FILES['imagen']['name'];

$ruta = 'uploads/instalaciones/' . $imagen;
move_uploaded_file($_FILES['imagen']['tmp_name'], $ruta);
```

Esta lógica guarda la imagen en el servidor y registra la ruta en la base.

 **Inserción del registro con imagen:**

```
$stmt = $pdo->prepare("INSERT INTO instalaciones (nombre,
descripcion, imagen) VALUES (?, ?, ?)");
$stmt->execute([$nombre, $descripcion, $imagen]);
```

 **Edición modular y eliminación:**

- editar_instalacion.php?id=2** carga los datos para edición.
- eliminar_instalacion.php?id=2** borra el registro y opcionalmente la imagen del servidor:

```
unlink('uploads/instalaciones/' . $nombre_imagen);
```

The screenshot shows a web-based administration interface for a swimming center. The main navigation bar includes links for Dashboard, Usuarios, Horarios, Promociones, Reglamento, Instalaciones, Competencias, Rehabilitación, Visitas, PDF, Gráfica, and a user account for 'admin (Administrador)'. A 'Cerrar Sesión' (Logout) button is also present. The central content area is titled 'Editar Instalación' (Edit Installation). It contains a form with the following fields:

- Nombre:** Gimnasio Acuático (Acuatic Gym)
- Descripción:** Área equipada con maquinaria de entrenamiento acuático.
- Imagen actual:** A small thumbnail image showing an aquatic gym with various training equipment.
- Cambiar imagen (opcional):** Buttons for 'Seleccionar archivo' (Select file) and 'Sin archivos seleccionados' (No files selected).

At the bottom of the form are two buttons: a green 'Guardar Cambios' (Save Changes) button with a checkmark icon, and a grey 'Cancelar' (Cancel) button.

Editar Instalación propia autoría



Editar Instalación - Modificación de Espacios Físicos



Distribución Visual

- **Encabezado principal:** “Editar Instalación” con ícono de edificio.
- **Formulario cargado con los datos actuales:**
 - Campo: Nombre (editable)
 - Campo: Descripción (editable)
 - Imagen actual mostrada en miniatura.
 - Campo para **subir una nueva imagen** (opcional).
- Botón verde “**Guardar Cambios**” para confirmar la edición.
- Botón gris “**Cancelar**” para regresar sin guardar cambios.

 **Funcionalidad**

- Esta vista permite **modificar el nombre, descripción o imagen** de una instalación ya registrada.
- La imagen actual se mantiene a menos que el usuario suba una nueva.
- El sistema valida si se ha subido una nueva imagen:
 - Si **sí**, reemplaza la imagen anterior (y opcionalmente la elimina del servidor).
 - Si **no**, conserva la anterior.
- El formulario es enviado mediante POST al mismo archivo o a uno como **actualizar_instalacion.php**.

 **Tres Datos Importantes de Programación**
 **Manejo condicional de imagen:**

```
if (!empty($_FILES['imagen']['name'])) {
    $nuevaImagen = $_FILES['imagen']['name'];
    move_uploaded_file($_FILES['imagen']['tmp_name'], "uploads/
instalaciones/$nuevaImagen");
    $sql = "UPDATE instalaciones SET nombre = ?, descripcion = ?,
imagen = ? WHERE id = ?";
} else {
    $sql = "UPDATE instalaciones SET nombre = ?, descripcion = ?
WHERE id = ?";
}
```

Esto evita errores si no se cambia la imagen.

 **Precarga de datos con GET:**

```
$stmt = $pdo->prepare("SELECT * FROM instalaciones WHERE id = ?");
$stmt->execute([$GET['id']]);
$instalacion = $stmt->fetch();
```

Muestra los datos actuales antes de editar.

 **Opción de limpieza del servidor:** Si se reemplaza una imagen, puede ejecutarse:

```
unlink("uploads/instalaciones/" . $imagenAnterior);
```

para eliminar la imagen vieja y liberar espacio

The screenshot shows a web browser window with the URL ferruzca.rf.gd/admin/gestionar_competencias.php. The page title is "Gestión de Competencias Acuáticas". The main content area has a blue header "Registrar Nueva Competencia" containing fields for "Nombre del evento", "Fecha" (dd/mm/aaaa), and "Descripción...". A green "Registrar" button is at the bottom. Below this is a table titled "Competencias Registradas" with columns "Nombre", "Fecha", "Descripción", and "Acciones" (with edit and delete icons). The table contains three rows: "Copa Interna" (2024-07-10, Anual entre alumnos), "Carrera Juvenil" (2024-06-01, Estilo mariposa y pecho), and "Torneo Infantil" (2024-05-12, Estilo libre para menores de 12 años).

Gestión de Competencias Acuáticas propia autoría

Gestión de Competencias Acuáticas - Organización de Eventos

Distribución Visual

- **Encabezado central:** “Gestión de Competencias Acuáticas”.
- **Formulario para registrar nueva competencia:**
 - Campo: Nombre del evento
 - Campo: Fecha (`type="date"`)
 - Campo: Descripción
 - Botón verde “**Registrar**”
- Botón “Volver al Panel” con estilo Bootstrap `btn-outline-primary`.
- **Tabla de competencias registradas:**
 - Columnas: Nombre, Fecha, Descripción, Acciones.
 - Íconos (editar) y (eliminar) permiten gestionar cada evento.

Funcionalidad

- Permite al administrador **registrar, consultar, modificar o eliminar** competencias.
- Los eventos se almacenan en la tabla **competencias** con campos como **id, nombre, fecha, descripcion**.
- Las fechas se validan antes de ser insertadas, y la tabla se actualiza automáticamente.
- Las acciones de editar y eliminar redirigen a archivos separados por **ID**, usando **GET**.

Tres Datos Importantes de Programación

Validación de campos antes de insertar:

```
if (empty($nombre) || empty($fecha) || empty($descripcion)) {
    echo "<div class='alert alert-danger'>Todos los campos son obligatorios.</div>";
}
```

Inserción con PDO:

```
$stmt = $pdo->prepare("INSERT INTO competencias (nombre, fecha,
descripcion) VALUES (?, ?, ?)");
$stmt->execute([$nombre, $fecha, $descripcion]);
```

Esto garantiza una conexión segura y eficiente.

Arquitectura modular:

- **gestionar_competencias.php**: vista principal.
- **editar_competencia.php?id=2**: carga la competencia para modificar.
- **eliminar_competencia.php?id=2**: elimina la competencia seleccionada.

The screenshot shows a web browser window with a purple decorative border. The address bar indicates the URL is http://ferruzca.rf.gd/admin/editar_competencia.php?id=3. The page title is 'Navbar - Centro de Natación'. The main navigation menu includes links like 'Dashboard', 'Usuarios', 'Horarios', 'Promociones', 'Reglamento', 'Instalaciones', 'Competencias', 'Rehabilitación', 'Visitas', 'PDF', 'Gráfica', and 'admin (Administrador)'. A 'Cerrar Sesión' (Logout) link is also visible. The central content area displays a modal dialog with a trophy icon and the title 'Editar Competencia'. It contains three input fields: 'Nombre del Evento' (Event Name) with the value 'Copa Interna', 'Fecha' (Date) set to '10/07/2024', and 'Descripción' (Description) with the text 'Evento anual entre alumnos del centro.'. Below the fields are two buttons: a green 'Guardar Cambios' (Save Changes) button with a checkmark icon, and a grey 'Cancelar' (Cancel) button.

Editar Competencia propia autoría

Editar Competencia - Actualización de Evento Acuático

Distribución Visual

- **Encabezado centrado:** "Editar Competencia" acompañado del ícono de trofeo.
- **Formulario precargado** con los datos actuales del evento:
 - Campo: Nombre del evento
 - Campo: Fecha (`type="date"`)
 - Campo: Descripción (multilínea)
- **Botón verde "Guardar Cambios"** para confirmar.
- **Botón gris "Cancelar"** para regresar sin modificar.

 **Funcionalidad**

- Al acceder a esta vista, se obtiene el **ID** del evento desde la **URL**:

```
editar_competencia.php?id=3
```

- El sistema precarga los datos mediante una consulta **SELECT**.
- El usuario puede modificar nombre, fecha y descripción del evento.
- Al guardar, se realiza una consulta **UPDATE** que actualiza los datos en la base.
- La edición es controlada para evitar duplicados o fechas inválidas.

 **Tres Datos Importantes de Programación**
 **Precarga con PDO y seguridad:**

```
$stmt = $pdo->prepare("SELECT * FROM competencias WHERE id = ?");

$stmt->execute([$_GET['id']]);
$competencia = $stmt->fetch();
```

Carga los datos actuales del evento de forma segura.

 **Actualización de la competencia:**

```
$stmt = $pdo->prepare("UPDATE competencias SET nombre = ?, fecha = ?, descripcion = ? WHERE id = ?");

$stmt->execute([$nombre, $fecha, $descripcion, $id]);
```

Asegura que solo se modifiquen los campos deseados.

 **Validación de campos obligatorios:**

- Verifica que ningún campo esté vacío.
- Se puede agregar validación de formato de fecha o de longitud de texto:

```
if (strlen($descripcion) > 255) {
    echo "La descripción es demasiado larga.";
}
```

Gestión de Rehabilitación de Nadadores

Registrar Nueva Sesión

Paciente:

Fecha:

Tipo de terapia:

Observaciones:

Paciente	Fecha	Terapia	Observaciones	Acciones
Lucía Díaz	2024-03-25	Rehabilitación articular	Dolor leve controlado.	
Pedro Morales	2024-03-22	Hidroterapia de espalda	Mejora en movilidad.	
Ana Torres	2024-03-20	Hidroterapia de rodilla	Recuperación postoperatoria.	

Gestión de Rehabilitación propia autoría

Gestión de Rehabilitación - Registro y Control Terapéutico

Distribución Visual

- **Encabezado principal:** “Gestión de Rehabilitación de Nadadores”.
- **Formulario de registro de sesión:**
 - **Paciente:** `select` con usuarios registrados (relación con tabla `usuarios` o `clientes`)
 - **Fecha:** campo `type="date"`
 - **Tipo de terapia:** menú desplegable (hidroterapia, rehabilitación articular, etc.)
 - **Observaciones:** campo de texto largo
 - Botón verde “**Registrar**”
- Botón “**Volver al Panel**” con estilo Bootstrap.
- **Tabla de sesiones registradas:**
 - Columnas: Paciente, Fecha, Terapia, Observaciones, Acciones.
 - Botones (editar) y (eliminar) por sesión.

Funcionalidad

- El sistema permite **registrar, visualizar, editar y eliminar sesiones de rehabilitación acuática**.
- La información se guarda en una tabla **rehabilitacion**, con campos como:
id, paciente_id, fecha, terapia, observaciones
- El **paciente** se relaciona con el **ID** del usuario o cliente de la base.
- Las sesiones se muestran automáticamente debajo del formulario.
- Cada acción se dirige a scripts **editar_sesion.php** y **eliminar_sesion.php**.

Tres Datos Importantes de Programación

Relación paciente-usuario:

```
$stmt = $pdo->prepare("SELECT id, nombre FROM usuarios WHERE nivel = 3");
```

Se filtran solo los usuarios con rol de **cliente** para evitar confusión.



Registro seguro con PDO:

```
$stmt = $pdo->prepare("INSERT INTO rehabilitacion (paciente_id, fecha, terapia, observaciones) VALUES (?, ?, ?, ?)");
$stmt->execute([$paciente_id, $fecha, $terapia, $observaciones]);
```

Garantiza que los datos sean almacenados de forma protegida.



Carga dinámica de sesiones:

```
$stmt = $pdo->query("SELECT r.*, u.nombre FROM rehabilitacion r JOIN usuarios u ON r.paciente_id = u.id ORDER BY fecha DESC");
```

Esto permite mostrar los datos con nombre completo del paciente gracias al **JOIN**.

Navbar - Centro de Natación

No seguro ferruzca.rf.gd/admin/editar_rehabilitacion.php?id=3

Centro de Natación Dashboard Usuarios Horarios Promociones Reglamento Instalaciones Competencias Rehabilitación Visitas PDF Gráfica admin (Administrador) Cerrar Sesión

Editar Sesión de Rehabilitación

Nombre del Paciente
Lucía Díaz

Fecha
25/03/2024

Tipo de Terapia
Rehabilitación articular

Observaciones
Dolor leve controlado.

[← Volver](#) [Actualizar](#)

Editar Sesión de Rehabilitación propia autoría

editar Sesión de Rehabilitación - Control y Seguimiento Terapéutico

Distribución Visual

- **Encabezado:** “Editar Sesión de Rehabilitación”, centrado y con estilo visual consistente.
- **Formulario con datos precargados:**
 - **Nombre del Paciente** (no editable, campo `readonly`)
 - **Fecha de la sesión** (`type="date"`)
 - **Tipo de Terapia** (campo de texto o `select`)
 - **Observaciones** (campo de texto multilínea)
- Botón verde “Actualizar”
- Botón gris “Volver” para regresar sin guardar cambios.

#	Nombre	Correo	Fecha de Registro
1	Pedro Morales	pedro@gmail.com	30/03/2025 08:46
2	Sandra Díaz	sandra@aol.com	30/03/2025 08:46
3	Luis Vega	luis@msn.com	30/03/2025 08:46
4	Valeria Núñez	valeria@outlook.com	30/03/2025 08:46
5	Mario Torres	mario@yahoo.com	30/03/2025 08:46

Funcionalidad

- Esta vista permite **modificar los detalles** de una sesión de rehabilitación previamente registrada.
- El **ID** de la sesión se pasa por la **URL**:
editar_rehabilitacion.php?id=3
- Al cargar, se ejecuta una consulta para traer los datos actuales.
- Al guardar, se actualiza el registro en la tabla **rehabilitacion**.
- Se valida que la fecha sea válida, el tipo de terapia esté definido y las observaciones no estén vacías.

Tres Datos Importantes de Programación

Precarga de datos con **GET** y **PDO**:

```
$stmt = $pdo->prepare("SELECT r.*, u.nombre FROM rehabilitacion r
JOIN usuarios u ON r.paciente_id = u.id WHERE r.id = ?");

$stmt->execute([$_GET['id']]);
$datos = $stmt->fetch();
```

Esto permite mostrar el nombre del paciente sin editarla.

 **Actualización segura:**

```
$stmt = $pdo->prepare("UPDATE rehabilitacion SET fecha = ?, terapia = ?, observaciones = ? WHERE id = ?");  
$stmt->execute([$fecha, $terapia, $observaciones, $id]);
```

Se mantiene la integridad del paciente vinculando solo los datos editables.

 **Validación de campos antes del update:**

Fecha no vacía ni futura.

Terapia seleccionada de un listado válido.

Observaciones con longitud aceptable:

```
if (strlen($observaciones) > 255) {  
    echo "Observación demasiado extensa.";  
}
```

Listado de Visitas Registradas

#	Nombre	Correo	Fecha de Registro
1	Pedro Morales	pedro@gmail.com	30/03/2025 08:46
2	Sandra Díaz	sandra@aol.com	30/03/2025 08:46
3	Luis Vega	luis@msn.com	30/03/2025 08:46
4	Valeria Núñez	valeria@outlook.com	30/03/2025 08:46
5	Mario Torres	mario@yahoo.com	30/03/2025 08:46

Visitas propia autoría

Listado de Visitas Registradas - Control de Acceso de Invitados

Distribución Visual

- 📌 **Encabezado principal:** “Listado de Visitas Registradas”, centrado y claro.
- 📋 **Tabla estructurada** con los siguientes campos:
 - #: Número consecutivo (ID o índice)
 - Nombre** del visitante
 - Correo electrónico** del visitante
 - Fecha de Registro** con hora exacta
- La tabla está contenida en un **div** con estilo Bootstrap, visualmente centrado y de lectura cómoda.

 **Funcionalidad**

- Muestra en tiempo real a todos los visitantes que accedieron al sistema como “visita”.
- La información proviene de una tabla **visitas** en la base de datos, probablemente con campos como:
`id, nombre, correo, fecha_registro`
- Cada registro es insertado automáticamente desde la vista pública o desde el login al entrar como visita.
- **No permite editar ni eliminar registros**, sólo visualización.

 **Tres Datos Importantes de Programación**
 **Consulta para mostrar visitas:**

```
$stmt = $pdo->query("SELECT * FROM visitas ORDER BY fecha_registro DESC");

while($row = $stmt->fetch()) {
    // renderizar tabla
}
```

Esta lógica garantiza que se muestren en orden cronológico descendente.

 **Registro automático con `timestamp`:**

Cuando una visita se registra, el sistema guarda automáticamente la fecha y hora:

```
$fecha = date('Y-m-d H:i:s');

$stmt = $pdo->prepare("INSERT INTO visitas (nombre, correo,
fecha_registro) VALUES (?, ?, ?)");
```

 **Validación en `frontend/backend`:**

- El nombre y correo del visitante se validan para evitar entradas vacías o incorrectas.
- Se puede usar `filter_var($correo, FILTER_VALIDATE_EMAIL)` para asegurar un email válido.

The screenshot shows a web browser window with a dark theme. The address bar indicates the URL is `ferruzca.rf.gd/reports/reporte_pdf.php`. The main content area displays a PDF document titled "Centro de Natación Innovex" with the subtitle "Reporte de Usuarios Inscritos". Below the title is a table with the following data:

ID	Nombre	Edad	Nivel	Inscripción
1	Carlos Rodriguez	10	Básico	2024-03-01
2	Andrea Ramírez	14	Intermedio	2024-03-05
3	Luis Méndez	18	Avanzado	2024-03-10

PDF propia autoría

Reporte PDF - Usuarios Inscritos en el Centro de Natación

Distribución Visual

-  **Encabezado centrado:**
 - **Título:** *Centro de Natación Innovex*
 - **Subtítulo:** *Reporte de Usuarios Inscritos*
-  **Tabla con formato profesional:**
 - Columnas: **ID, Nombre, Edad, Nivel, Inscripción**
 - La tabla está centrada y con bordes visibles, alineada para exportación directa.
-  Vista PDF generada automáticamente desde **reporte_pdf.php**.

 **Funcionalidad**

- Este reporte se genera dinámicamente desde PHP usando la librería **FPDF**.
- Toma datos desde la base de datos, directamente desde la tabla de usuarios o inscripciones.
- Se puede acceder vía:

`/reports/reporte_pdf.php`

- La salida del archivo es inmediata en el navegador con opción de descarga o impresión.

 **Tres Datos Importantes de Programación**
 **Configuración básica de FPDF:**

```
require('fpdf/fpdf.php');

$pdf = new FPDF();
$pdf->AddPage();
$pdf->SetFont('Arial','B',14);
$pdf->Cell(0,10,'Centro de Natación Innovex',0,1,'C');
```

 **Consulta a la base de datos con PDO:**

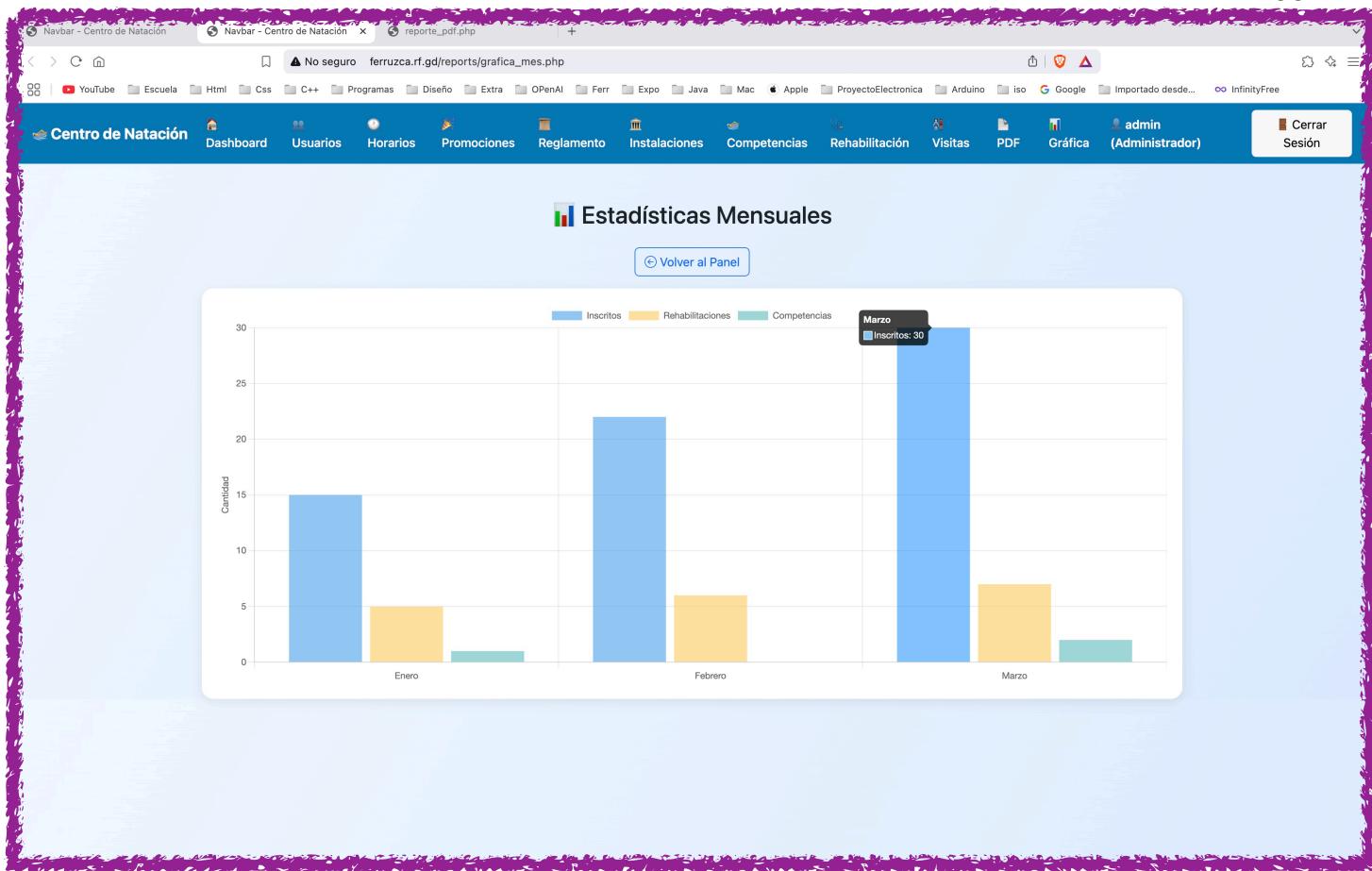
```
$stmt = $pdo->query("SELECT * FROM usuarios WHERE nivel_usuario = 3");

while ($row = $stmt->fetch()) {
    $pdf->Cell(40,10,$row['nombre'],1);
    // y así sucesivamente...
}
```

 **Salida del documento PDF:**

```
$pdf->Output('I', 'reporte_usuarios.pdf');
```

Esto abre el PDF directamente en el navegador sin necesidad de guardarlo.



Gráfica propia autoría

Estadísticas Mensuales - Gráfica de Actividades

Distribución Visual

- 📌 Título centrado: *Estadísticas Mensuales* con ícono de barra.
- ⬅ BACK Botón "Volver al Panel" con estilo Bootstrap.
- 📈 Gráfico de columnas con tres colores:
 - Azul: Inscritos
 - Amarillo: Rehabilitaciones
 - Aqua: Competencias
- 📅 Eje X con los meses: **Enero, Febrero, Marzo**
- 📏 Eje Y con el número de registros por actividad.



Funcionalidad

- Genera dinámicamente un **gráfico comparativo** mensual de las principales actividades:
 - Usuarios inscritos
 - Sesiones de rehabilitación registradas
 - Competencias organizadas
- Se visualiza mediante la biblioteca **Chart.js** o similar, renderizada con datos desde PHP.
- URL de acceso:

`/reports/grafica_mes.php`

📌 Tres Datos Importantes de Programación

📍 Consulta dinámica por mes:

```
SELECT MONTH(fecha) as mes, COUNT(*) as total FROM inscripciones
GROUP BY mes;
```

Repetido para **rehabilitacion** y **competencias**.

📦 Pasar datos de PHP a JavaScript:

```
echo json_encode($datos);
```

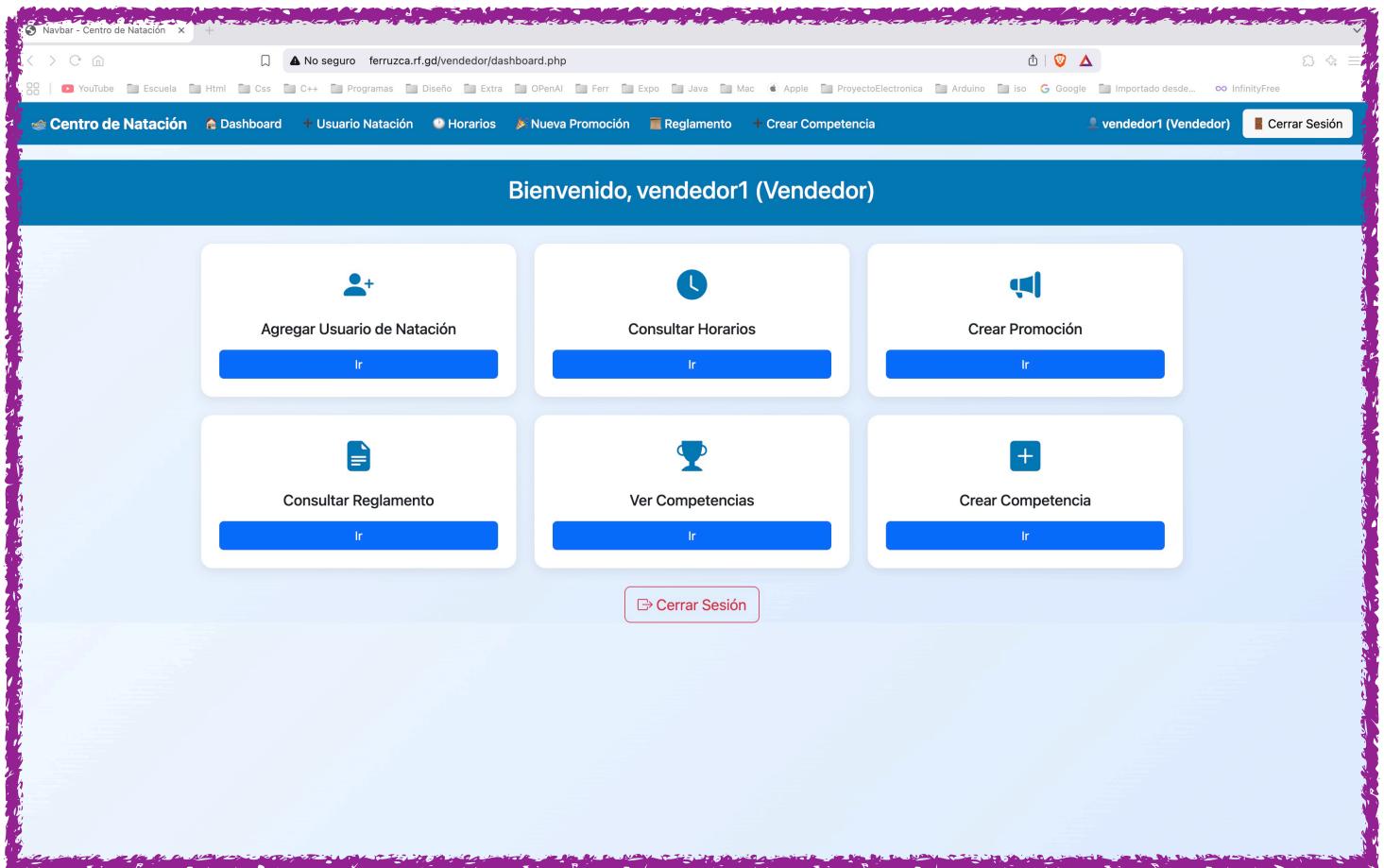
Luego se utiliza con **Chart.js** en:

```
const chartData = <?php echo $datos_json; ?>;
```

 **Configuración del gráfico:**

```
new Chart(ctx, {  
    type: 'bar',  
    data: {  
        labels: ['Enero', 'Febrero', 'Marzo'],  
        datasets: [  
            { label: 'Inscritos', backgroundColor: '#80d0ff', data: [15,  
22, 30] },  
            { label: 'Rehabilitaciones', backgroundColor: '#ffd86a', data:  
[5, 6, 7] },  
            { label: 'Competencias', backgroundColor: '#80ecec', data: [1,  
0, 2] }  
        ]  
    },  
    options: {  
        responsive: true,  
        scales: {  
            y: { beginAtZero: true }  
        }  
    }  
});
```

3.5 Vendedor



Dashboard Vendedor propia autoría

💻 Panel del Vendedor - Gestión Parcial del Sistema

🧭 Distribución Visual

🌐 Encabezado con el título "**Bienvenido, vendedor1 (Vendedor)**", personalizado según el usuario activo en sesión.

📍 Menú de navegación superior (`navbar_vendedor.php`) con accesos a las funciones disponibles para el rol de vendedor:

- Dashboard
- Usuario Natación
- Horarios
- Nueva Promoción
- Reglamento
- Crear Competencia

 Tarjetas de gestión distribuidas en una cuadrícula clara (probablemente utilizando `row` y `col-md-4` de Bootstrap) que enlazan con funcionalidades específicas mediante botones azules con íconos visuales SVG o FontAwesome:

1.  Agregar Usuario de Natación

2.  Consultar Horarios

3.  Crear Promoción

4.  Consultar Reglamento

5.  Ver Competencias

6.  Crear Competencia

 Botón "Cerrar Sesión" resaltado en rojo para indicar su funcionalidad de cierre seguro.

Funcionalidad

 Cada tarjeta redirige al **CRUD correspondiente** donde el vendedor puede registrar o consultar información, dependiendo del permiso otorgado por su nivel.

 Las secciones están restringidas solo a las funcionalidades que el **nivel 2 (Vendedor)** tiene asignadas.

 La interfaz está optimizada para la navegación rápida y eficiente desde cualquier dispositivo.

 Se integran íconos para cada módulo, reforzando la **experiencia visual del usuario**.

Tres Datos Importantes de Programación

Control de acceso con sesión y nivel:

```
session_start();
if ($_SESSION['nivel'] != 2) {
    header("Location: ../auth/login.php");
    exit();
}
```

 Garantiza que **solo usuarios con nivel 2** puedan ver y operar este panel.

 **Organización de rutas:** Cada acción está bien separada por módulos en carpetas como:

- `vendedor/gestionar_horarios.php`
- `vendedor/crear_promocion.php`
- `vendedor/crear_competencia.php`

 **Estructura modular del panel:** Utiliza `dashboard.php` con tarjetas visuales y código limpio, lo que permite **escalabilidad** y mantenibilidad del sistema.

Rol	Usuario	Password
Vendedor	vendedor1	vende123

Registrar Nuevo Usuario de Natación

Volver al panel

Datos del Usuario

Nombre completo
Ej: Juan Pérez

Edad
Edad

Nivel
-- Selecciona un nivel --

Registrar

Usuarios Registrados en Natación

#	Nombre	Edad	Nivel	Fecha de Inscripción
1	Luis Méndez	18	Avanzado	10/03/2024
2	Andrea Ramírez	14	Intermedio	05/03/2024
3	Carlos Rodríguez	10	Básico	01/03/2024

Registrar Nuevo Usuario Natación propia autoría

👤 Registro de Usuarios de Natación - Panel del Vendedor

🧭 Distribución Visual

- 🌐 Encabezado con el título "**Registrar Nuevo Usuario de Natación**", centrado y destacado visualmente.
- 📍 Menú de navegación (`navbar_vendedor.php`) activo, con botones visibles para todas las funciones del vendedor: Dashboard, Usuario Natación, Horarios, Nueva Promoción, Reglamento, Crear Competencia.
- 📘 Formulario de registro con el título "**Datos del Usuario**", secciones claras para:
 - 👉 **Nombre completo**
 - 👉 **Edad**
 - 👉 **Nivel de natación** (Básico, Intermedio, Avanzado)
 - 👉 **Registrar** con ícono fa-check

 Tabla inferior titulada "**Usuarios Registrados en Natación**", con las columnas:

- Nombre
- Edad
- Nivel
- Fecha de Inscripción

Colores y diseño responsive usando Bootstrap, con sección de datos debajo del formulario para **retroalimentación inmediata**.

Funcionalidad

-  Permite registrar nuevos usuarios en natación desde el rol de vendedor.
-  El campo "Nivel" se selecciona desde un dropdown con opciones limitadas.
-  Validaciones básicas para evitar campos vacíos.
-  Al registrar, los datos se guardan en la base de datos y se muestran en la tabla inferior.
-  Tabla actualizada en tiempo real tras cada inserción (mediante recarga o Ajax si se implementa).
-  Fecha de inscripción se genera automáticamente con la fecha actual del sistema (`date("Y-m-d")` en PHP).

Tres Datos Importantes de Programación

 Control de acceso con sesión para nivel de vendedor:

```
session_start();
if ($_SESSION['nivel'] != 2) {
    header("Location: ../auth/login.php");
    exit();
}
```

 Inserción de usuario con consulta SQL preparada (idealmente con PDO para seguridad):

```
$nombre = $_POST['nombre'];
$edad = $_POST['edad'];
$nivel = $_POST['nivel'];
$fecha = date("Y-m-d");
$sql = "INSERT INTO usuarios_natacion (nombre, edad, nivel,
fecha_inscripcion) VALUES (?, ?, ?, ?)";
```

 Estructura de archivos organizada en `vendedor/agregar_usuario_natacion.php`, permitiendo mantener el flujo exclusivo de vendedor sin invadir el panel del administrador.

Día	Hora Inicio	Hora Fin	Actividad
Lunes	08:00:00	09:00:00	Natación básica
Martes	10:00:00	11:00:00	Rehabilitación física
Miércoles	09:00:00	10:00:00	Competencia Juvenil

Consulta de Horarios propia autoría

⌚ Consulta de Horarios - Panel del Vendedor

🧭 Distribución Visual

🌐 Título centrado: "**Consulta de Horarios**"

📍 Menú superior activo desde **navbar_vendedor.php** con opciones clave:
Dashboard | Usuario Natación | Horarios | Nueva Promoción | Reglamento | Crear Competencia

📋 Sección principal:

- Título de la tabla: "**Horarios de Actividades**" con barra azul
- 📆 Tabla bien estructurada con columnas:
 - Día
 - Hora Inicio
 - Hora Fin
 - Actividad

 Diseño claro, responsive y con espacio amigable usando Bootstrap.

 Botón azul "Volver al Panel" ubicado en la parte superior izquierda para regresar al dashboard del vendedor.

Funcionalidad

 Este archivo (`ver_horarios.php`) permite al vendedor consultar los horarios registrados en el sistema.

 No permite edición, solo visualización (modo solo lectura).

 Los datos se obtienen mediante una consulta SQL a la tabla `horarios`.

 La información es dinámica, mostrando actividades como:

-  Natación básica
-  Rehabilitación física
-  Competencia Juvenil

 Esta vista está protegida por sesión del vendedor (`nivel = 2`).

Tres Datos Importantes de Programación

Validación de sesión:

```
session_start();
if ($_SESSION['nivel'] != 2) {
    header("Location: ../auth/login.php");
    exit();
}
```

Consulta SQL para mostrar horarios:

```
$sql = "SELECT * FROM horarios WHERE estado = 1 ORDER BY dia ASC";
$resultado = $pdo->query($sql);
```

 Ubicación del archivo: `/vendedor/ver_horarios.php`, respetando la estructura modular del sistema.

Crear Nueva Promoción

[Volver al Panel](#)

Formulario de Promoción

Título:

Descripción:

Inicio: dd/mm/aaaa Fin: dd/mm/aaaa

[Crear Promoción](#)

Promociones Registradas

#	Título	Descripción	Inicio	Fin
1	Combo Familiar	Inscripción para 3 con precio especial.	2024-05-01	2025-05-31
2	Mes del Niño	Primera clase gratis para menores de 12 años.	2024-04-15	2024-05-15
3	Descuento Primavera	20% de descuento en inscripción.	2024-04-01	2024-04-30

Crear Nueva Promoción propia autoría



Crear Nueva Promoción - Panel del Vendedor



Distribución Visual

📍 Encabezado centrado con el título "**Crear Nueva Promoción**"

◆ Barra azul con título interno: "**Formulario de Promoción**"



Formulario claro y ordenado con campos para:

- ✍ Título de la promoción
- 📝 Descripción
- 📅 Fecha de inicio (con ícono de calendario)
- 📅 Fecha de fin (con ícono de calendario)

✓ Botón verde con ícono ✓ "Crear Promoción" para enviar el formulario.

 Tabla inferior con barra celeste titulada "**Promociones Registradas**" que muestra:

- Número
- Título
- Descripción
- Fecha de Inicio
- Fecha de Fin

 Diseño basado en Bootstrap con estilo moderno, limpio y responsive.

Funcionalidad

 Esta vista permite a los **vendedores** (nivel = 2) registrar nuevas promociones para el centro de natación.

 Despues de enviar el formulario:

- Los datos se insertan en la tabla **promociones**.
- Se actualiza automáticamente la tabla inferior con las promociones activas.
- No se requiere recarga manual.

 El archivo **crear_promocion.php** está protegido por sesión:

```
session_start();
if ($_SESSION['nivel'] != 2) {
    header("Location: ../auth/login.php");
    exit();
}
```

Tres Datos Importantes de Programación

Inserción SQL con PDO:

```
$sql = "INSERT INTO promociones (titulo, descripcion, fecha_inicio,
fecha_fin) VALUES (?, ?, ?, ?, ?)";
$stmt = $pdo->prepare($sql);
$stmt->execute([$titulo, $descripcion, $inicio, $fin]);
```

 Ruta del archivo: /vendedor/crear_promocion.php

Listado dinámico de promociones:

```
$sql = "SELECT * FROM promociones WHERE estado = 1 ORDER BY
fecha_inicio ASC";
```

Reglamento propia autoría

Reglamento del Centro de Natación - Panel del Vendedor

Distribución Visual

-  Encabezado centrado con el título: "**Reglamento del Centro de Natación**"
-  Barra azul con título: "**Normas y Reglas Vigentes**"
-  Sección organizada en forma de lista desplegable estilo Bootstrap o tabla simple con fondo blanco y bordes claros.
-  Cada regla está acompañada con un ícono  de validación para enfatizar que está activa y vigente.
-  Botón superior "**Volver al Panel**" permite regresar al dashboard del vendedor.

Funcionalidad

 Este módulo es de solo lectura para los vendedores (nivel 2).

No pueden modificar, eliminar ni agregar nuevas reglas, únicamente visualizar.

 Las reglas se extraen desde la base de datos, generalmente de la tabla **reglamento** con un **estado = 1** (activas).

 Se muestran todas las reglas que los administradores han creado y aprobado para la vista pública.

Tres Datos Importantes de Programación

Consulta SQL:

```
$sql = "SELECT * FROM reglamento WHERE estado = 1 ORDER BY id ASC";
```

 Control de sesión para vendedores:

```
session_start();
if ($_SESSION['nivel'] != 2) {
    header("Location: ../auth/login.php");
    exit();
}
```

 Ruta del archivo: /vendedor/ver_reglamento.php

Crear Nueva Competencia

🏁 Crear Nueva Competencia - Gestión por Vendedor

Distribución Visual

- Encabezado principal centrado con el título "**Crear Nueva Competencia**"
- Sección del formulario destacada con el título: "**Formulario de Competencia**"

Campos:

- **Nombre del evento:** Entrada de texto (`<input>`)
- **Fecha:** Selector tipo fecha (`<input type="date">`)
- **Descripción:** Área de texto ampliada para detallar el evento.

Botón verde "**Registrar Competencia**" con ícono de check para enviar el formulario.

Botón superior "Volver al Panel" permite regresar al dashboard del vendedor.

 Tabla inferior con fondo celeste que muestra las **Competencias Registradas**, incluyendo:

- Número de orden
- Nombre del evento
- Fecha programada
- Descripción del evento

Funcionalidad

 Este módulo permite al **vendedor (nivel 2)** registrar nuevas competencias acuáticas.

 Los datos se almacenan en la tabla **competencias**, normalmente con columnas como:

`id, nombre, fecha, descripcion, estado`

 Al registrar, los valores se insertan mediante **INSERT INTO** y luego se muestran dinámicamente en la tabla inferior.

 La tabla solo muestra competencias **activas** (estado = 1) para mantener limpieza visual.

Tres Datos Importantes de Programación

 Validación por sesión de vendedor:

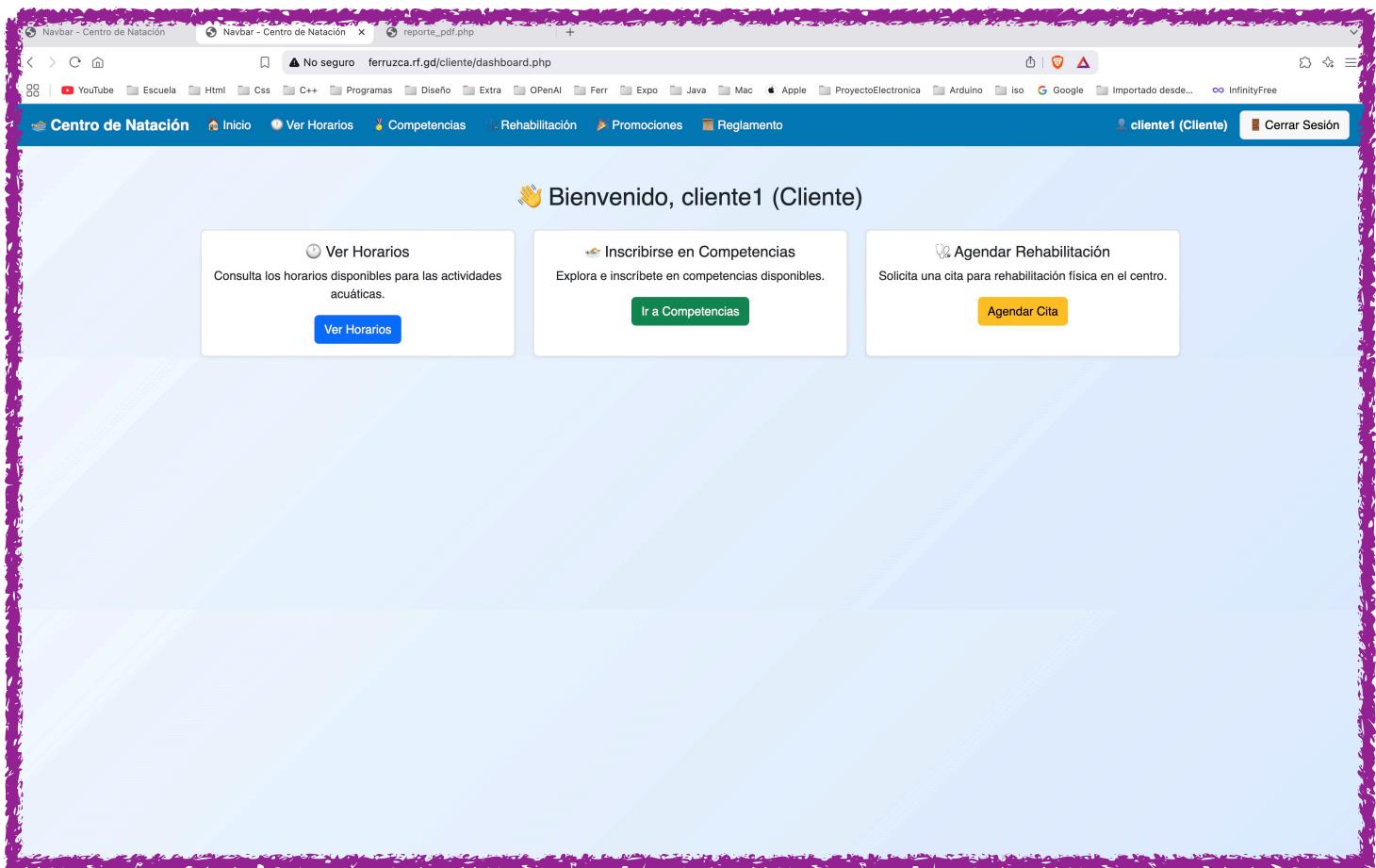
```
session_start();
if ($_SESSION['nivel'] != 2) {
    header("Location: ../auth/login.php");
    exit();
}
```

 Consulta para mostrar competencias:

```
$sql = "SELECT * FROM competencias WHERE estado = 1 ORDER BY fecha ASC";
```

 Ruta del archivo: `/vendedor/crear_competencia.php`

3.6 Cliente



Dashboard Cliente propia autoría



Panel del Cliente - Portal Personal del Usuario



Distribución Visual

- **Mensaje de bienvenida personalizado:**
 - "Bienvenido, cliente1 (Cliente)", con ícono de saludo y el rol del usuario.
- **Tres tarjetas principales con acciones:**
 - **Ver Horarios**
 - Texto: “Consulta los horarios disponibles para las actividades acuáticas.”
 - Botón azul con **Ver Horarios**
 - **Inscribirse en Competencias**
 - Texto: “Explora e inscríbete en competencias disponibles.”
 - Botón verde con **Ir a Competencias**

- **Agendar Rehabilitación**
 - Texto: “Solicita una cita para rehabilitación física en el centro.”
 - Botón amarillo con **Agendar Cita**
-  Botón de **Cerrar Sesión** visible en la parte superior derecha.

Funcionalidad

- Accesible solo para usuarios con nivel de acceso **Cliente** (nivel 3).
- Interfaz simplificada y centrada en tres acciones específicas.
- Las acciones redirigen a páginas internas:
 - **ver_horarios.php**
 - **inscribir_competencia.php**
 - **agendar_rehabilitacion.php**

Tres Datos Importantes de Programación

Control de sesión:

```
session_start();

if ($_SESSION['nivel'] != 3) {
    header("Location: ../auth/login.php");
    exit;
}
```

Diseño con Bootstrap:

Las tarjetas utilizan clases como **card**, **btn btn-primary**, **text-center**, y **shadow-sm** para una interfaz limpia y responsive.

Redirección dinámica por botón:

```
<a href="ver_horarios.php" class="btn btn-primary">Ver Horarios</a>
```

Esto mantiene una navegación clara sin necesidad de JavaScript adicional.

Rol	Usuario	Password
Cliente	cliente1	cliente123

Día	Hora de Inicio	Hora de Fin	Actividad
Lunes	08:00:00	09:00:00	Natación básica
Martes	10:00:00	11:00:00	Rehabilitación física
Miércoles	09:00:00	10:00:00	Competencia Juvenil

Vista Horarios propia autoría

Horarios Disponibles - Consulta para Clientes

Distribución Visual

- **Título centrado:** “Horarios Disponibles” con ícono de reloj
- **Tabla informativa** con 4 columnas:
 1. **Día**
 - Ejemplos: Lunes, Martes, Miércoles
 2. **Hora de Inicio**
 - Ejemplos: **08:00:00, 10:00:00**
 3. **Hora de Fin**
 - Ejemplos: **09:00:00, 11:00:00**
 4. **Actividad**
 - Actividades como *Natación básica, Rehabilitación física y Competencia Juvenil*
- La tabla se encuentra centrada y organizada con bordes visibles y un diseño limpio.



Funcionalidad

- Accesible desde el **panel del cliente**, al hacer clic en "Ver Horarios".
- La información se obtiene directamente desde la base de datos, mostrándose de manera dinámica.
- Los datos son **solo de lectura**, sin opciones de edición o eliminación para el cliente.



Tres Datos Importantes de Programación



Consulta desde MySQL vía PDO:

```
$sql = "SELECT dia, hora_inicio, hora_fin, actividad FROM horarios";
$stmt = $pdo->query($sql);
```



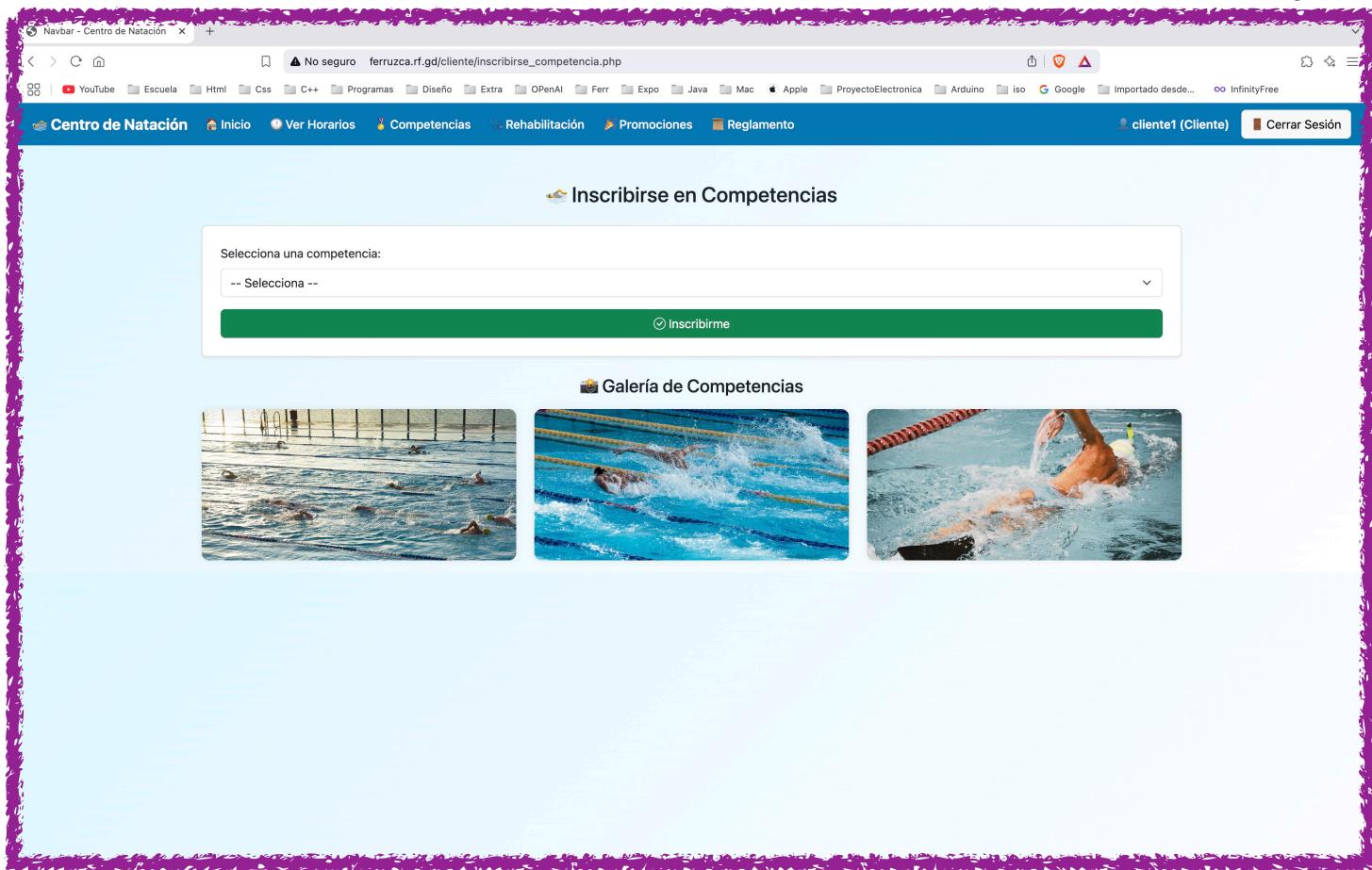
Impresión dinámica en tabla HTML:

```
while ($fila = $stmt->fetch()) {
    echo "<tr><td>{$fila['dia']}</td><td>{$fila['hora_inicio']}</td><td>{$fila['hora_fin']}</td><td>{$fila['actividad']}</td></tr>";
}
```



Diseño responsive con Bootstrap:

Uso de clases `table table-bordered table-striped text-center` para una presentación clara y profesional.



Inscribirse en Competencias propia autoría

Inscribirse en Competencias - Módulo del Cliente

Diseño de la Interfaz

- **Encabezado:**
Título centrado con ícono "🏊 Inscribirse en Competencias"
Estética limpia con Bootstrap y fondo azul claro.
- **Formulario principal:**
 - **Selector desplegable (<select>):**
Lista las competencias acuáticas disponibles desde la base de datos.
 - **Botón "Inscribirme":**
Botón verde grande (`btn btn-success btn-block`) que ejecuta la inscripción del usuario seleccionado a la competencia.
- **Galería visual:**
 - Título: *Galería de Competencias.*
 - Tres imágenes representativas de competencias en natación.
 - Las imágenes están distribuidas horizontalmente en tarjetas con bordes redondeados.

Funcionamiento Interno (Lógica del Backend)

Carga de Competencias (PDO + PHP):

```
$stmt = $pdo->query("SELECT id, nombre FROM competencias");

while ($row = $stmt->fetch()) {
    echo "<option value='{$row['id']}'>{$row['nombre']}</option>";
}
```

Inscripción a Competencia (POST):

El botón activa un **POST** con el `id` de la competencia seleccionada.

Se inserta en una tabla intermedia tipo `inscripciones_competencias`:

```
INSERT INTO inscripciones_competencias (cliente_id, competencia_id,
fecha_registro)
VALUES (?, ?, NOW())
```

Galería estática o dinámica (según implementación):

Las imágenes pueden estar directamente en **HTML** o cargarse desde una tabla `galeria_competencias`.

Experiencia de Usuario

- Flujo sencillo y directo: selección de competencia → clic → inscripción.
- Botón claro, accesible y con diseño responsive.
- Apoyo visual que refuerza el propósito de la sección.

The screenshot shows a web browser window with the URL ferruzca.rf.gd/cliente/agendar_rehabilitacion.php. The page title is "Centro de Natación". The main content is a form titled "Agendar Rehabilitación" with the following fields:

- Fecha: A date input field with placeholder "dd/mm/aaaa".
- Tipo de Terapia: A dropdown menu with placeholder "-- Selecciona --".
- Observaciones: A text area for notes.
- A large green button at the bottom right labeled "Agendar Cita" with a calendar icon.

Agendar Rehabilitación propia autoría

Agendar Rehabilitación - Panel del Cliente

Componentes de la Interfaz

- **Encabezado centrado:**
Título con ícono  "Agendar Rehabilitación" en un diseño limpio y centrado.
- **Formulario con campos:**
 -  **Fecha:** Campo de tipo fecha (`<input type="date">`) obligatorio.
 -  **Tipo de Terapia:** Menú desplegable que permite elegir entre varias terapias (ej. *Rehabilitación articular, Hidroterapia de espalda, etc.*).
 -  **Observaciones:** Campo de texto libre para anotar cualquier detalle relevante (dolor, diagnóstico, etc.).
- **Botón "Agendar Cita":**
Botón verde de gran tamaño con ícono de calendario (`btn btn-success btn-block`), que envía los datos del formulario.

Lógica de Funcionamiento (Backend PHP + PDO)

Carga dinámica de terapias:

```
$stmt = $pdo->query("SELECT id, nombre FROM terapias");

while ($row = $stmt->fetch()) {
    echo "<option value='{$row['id']}'>{$row['nombre']}

```

Registro de cita al presionar "Agendar Cita":

```
$sql = "INSERT INTO sesiones_rehabilitacion (cliente_id, fecha,
terapia_id, observaciones)

VALUES (?, ?, ?, ?)";

$stmt->execute([$cliente_id, $fecha, $terapia_id, $observaciones]);
```

Validaciones:

Campos requeridos: fecha, tipo de terapia.

El campo observaciones puede ser opcional pero con límite de caracteres (ej. `maxlength="255"`).

Experiencia del Usuario

- Intuitiva y centrada en la acción.
- El botón ocupa todo el ancho y se presenta claramente.
- Las terapias deben reflejarse desde la base de datos para evitar errores y permitir mantenimiento desde el panel de administrador.

The screenshot shows a web browser window with the URL ferruzca.rf.gd/cliente/ver_promociones.php. The page is titled "Promociones Activas". A blue-bordered card on the left contains the following information:

- Combo Familiar**
- Inscripción para 3 con precio especial.
- Válido del 2024-05-01 al 2025-05-31

Promociones Activas propia autoría

Promociones Activas - Vista del Cliente

Componentes Visuales

- **Título centrado con ícono festivo:**
"🎉 Promociones Activas" en la parte superior, alineado al centro con estilo limpio.
- **Tarjeta de promoción:**
 - **Título** en negritas y color azul: "Combo Familiar"
 - **Descripción:** "Inscripción para 3 con precio especial."
 - **Fechas de vigencia:** mostradas claramente en formato YYYY-MM-DD.
- **Diseño visual:**
 - Uso de **card** estilo Bootstrap con bordes suaves y sombreados.
 - Margen entre elementos para mejorar la lectura.

Funcionamiento (PHP + PDO)

Consulta de promociones vigentes:

```
$hoy = date('Y-m-d');

$sql = "SELECT * FROM promociones WHERE fecha_inicio <= ? AND
fecha_fin >= ?";
$stmt = $pdo->prepare($sql);
$stmt->execute([$hoy, $hoy]);
```

Presentación en tarjetas:

```
foreach ($stmt as $promo) {

    echo "<div class='card'>";
    echo "<h5>{$promo['titulo']}</h5>";
    echo "<p>{$promo['descripcion']}</p>";
    echo "<small>Válido del {$promo['fecha_inicio']} al
{$promo['fecha_fin']}</small>";
    echo "</div>";
}
```

Experiencia del Usuario

- Solo se muestran promociones vigentes en el día actual.
- Visualización clara sin necesidad de hacer clic.
- Ideal para fomentar la inscripción de clientes.

Reglamento del Centro de Natación

Normas Generales

- Respetar los horarios asignados.
- Uso obligatorio de gorra y sandalias.
- Prohibido correr en el área de alberca.

Reglamento propia autoría

Reglamento del Centro de Natación - Vista del Cliente

Componentes Visuales

- **Título centrado con ícono de pergamino:**
"Reglamento del Centro de Natación" con tipografía clara y amigable.
- **Panel de Normas Generales:**
 - Cada regla aparece con un ícono de verificación .
 - Fondo claro, diseño minimalista.
 - Agrupadas en una card organizada bajo un encabezado azul.

Funcionamiento (PHP + PDO)

Consulta de reglas activas:

```
$sql = "SELECT * FROM reglamento ORDER BY id ASC";
$stmt = $pdo->query($sql);
```



Despliegue en lista con iconos:

```
foreach ($stmt as $row) {
    echo "<li class='list-group-item'><i class='bi bi-check-circle'></i> {$row['texto']}</li>";
}
```

Experiencia del Usuario

- Lectura simple y directa.
- Estilo coherente con el resto del sistema.
- Muestra exactamente lo mismo que el administrador configuró en `gestionar_reglamento.php`.

4. Conclusión

El desarrollo del Sistema de Gestión para el Centro de Natación Innovex representa una solución tecnológica integral, moderna y funcional, diseñada para cubrir de manera eficiente las necesidades operativas, administrativas y de atención al cliente dentro de un entorno deportivo-acuático. Desde sus primeras fases, el proyecto fue concebido con el propósito de optimizar los procesos internos del centro, facilitando la gestión de usuarios, horarios, promociones, reglamentos, instalaciones, competencias, sesiones de rehabilitación y visitas, así como la generación de reportes y estadísticas gráficas.

A lo largo del desarrollo, se implementaron niveles de acceso personalizados para garantizar la seguridad y el control de funciones según el rol del usuario (Administrador, Vendedor, Cliente y Visita). Cada uno de estos roles fue cuidadosamente diseñado con vistas específicas, funcionalidades diferenciadas y una experiencia de usuario intuitiva y bien estructurada. El administrador cuenta con un panel completo de control que le permite supervisar, modificar y generar reportes de todas las áreas del sistema. El vendedor tiene acceso a herramientas específicas para inscribir usuarios, gestionar competencias y registrar promociones. Por su parte, el cliente puede consultar horarios, inscribirse a competencias, solicitar sesiones de rehabilitación y consultar promociones vigentes de manera ágil y clara.

Uno de los logros más destacados del sistema es su estructura modular y escalable, que facilita futuras ampliaciones, mantenimiento y personalización. Gracias al uso de tecnologías como PHP, MySQL, Bootstrap y FPDF, se logró un sistema robusto y visualmente agradable, adaptable a cualquier dispositivo.

Asimismo, se garantizó la integridad y seguridad de los datos mediante sesiones, validaciones por nivel y control de accesos.

Otro aspecto a resaltar es el enfoque visual del sistema, en el que se incorporaron íconos, tarjetas dinámicas, botones de acción destacados y tablas ordenadas con estilos modernos, mejorando la experiencia de navegación para cualquier tipo de usuario. Además, la implementación de reportes en PDF y gráficas estadísticas mensuales permite tomar decisiones basadas en datos reales, lo cual es fundamental para la administración estratégica del centro.

En conclusión, este sistema no solo cumple con las expectativas planteadas al inicio del proyecto, sino que también representa una base sólida para futuras mejoras, digitalización de procesos y expansión tecnológica del Centro de Natación Innovex. Este trabajo demuestra la importancia de integrar la tecnología con la gestión deportiva y administrativa, y refleja el compromiso con la innovación, la organización y la atención de calidad hacia los usuarios.