

UNIVERSIDAD MARIANO GALVEZ

FACULTAD DE INGENIERIA EN SISTEMAS

CURSO DE DESARROLLO WEB

Grupo 2.

MANUAL TECNICO DE TIMER PARA USO DE SALAS

Contenido

1.	Introducción.....	3
2.	Requisitos del Sistema	4
3.	Estructura del Sistema	4
4.	Instalación.....	4
	Paso 1: Configuración del Servidor.....	4
	Paso 2: Instalar CodeIgniter	5
	Paso 3: Configuración de la Base de Datos	5
5.	Descripción de Controladores.....	5
5.1	CRUD de Clientes (CodeIgniter).....	5
5.2	CRUD de Dashboard (CodeIgniter).....	6
5.3	CRUD de LOGIN (CodeIgniter)	6
	Controlador: ` application/controllers/Login.php `	6
5.4	CRUD de Paquetes (CodeIgniter)	7
	Controlador: ` application/controllers/Paquetes.php `	7
5.5	Cronómetro de Uso de Salas (CodeIgniter).....	8
	Controlador: ` application/controllers/Salas.php `	8
5.6	Cronómetro de Usuarios (CodeIgniter).....	9
	Controlador: ` application/controllers/Usuarios.php `	9
5.7	Cronómetro de Usuarios (CodeIgniter).....	10
	Controlador: ` application/controllers/Usuarios.php `	10
6.	Lógica del cronometro.....	10
7.	Mantenimiento y Actualización.....	13
8.	Solución de Problemas Comunes.....	13

1. Introducción

Este sistema permite la gestión de salas o ubicaciones para su uso, con un CRUD (Crear, Leer, Actualizar y Eliminar) para administrar salas, precios, clientes, usuarios y un cronómetro para calcular los tiempos de uso de las salas, todo utilizando el framework CodeIgniter.

2. Requisitos del Sistema

Requisitos de Software:

- Sistema Operativo: Windows, Linux, macOS
- Servidor Web: Apache
- PHP 7.4 o superior
- Base de Datos: SQL LITE
- CodeIgniter 4.x
- Composer (para la gestión de dependencias)
- Navegador: Chrome, Firefox o Edge (para la interfaz gráfica)

Requisitos de Hardware:

- Procesador: Intel i3 o superior
- RAM: 4GB mínimo
- Espacio en disco: 500MB para la instalación del sistema

3. Estructura del Sistema

El sistema se organiza en diferentes módulos que interactúan entre sí:

- Módulo de Salas: Permite gestionar la creación y administración de las salas o ubicaciones.
- Módulo de Paquetes: Administra los precios de las salas según la duración y servicios.
- Módulo de Clientes: Registra los clientes que usarán las salas.
- Módulo de Usuarios: Permite gestionar los roles y permisos de los usuarios del sistema.
- Cronómetro: Calcula el tiempo que una sala es usada para generar la facturación correspondiente.

4. Instalación

Paso 1: Configuración del Servidor

1. Instalar Apache
2. Instalar PHP 7.4 o superior.
3. Instalar SQL LITE

Paso 2: Instalar CodeIgniter

Descargar e instalar CodeIgniter 4:

```
composer create-project codeigniter4/appstarter gestion-salas
```

Paso 3: Configuración de la Base de Datos

En el archivo `Cronometroumg/application/config/database.php` de CodeIgniter, configurar la base de datos:

```
'database' => FCPATH . 'db/proyecto.db', // Nueva ruta relativa
```

Crear las tablas para las salas, paquetes, clientes, usuarios, etc.

Paso 5: Iniciar el Servidor

Ejecutar el servidor de con el servicio de Code Igniter, será necesario tener un servidor con Apache, ISS o la tecnología que uses agregando el proyecto a la ruta del servicio.

5. Descripción de Controladores.

5.1 CRUD de Clientes (CodeIgniter)

Este controlador maneja las operaciones CRUD de los clientes.

- Listar Clientes:
 - Método: GET
 - Ruta: /clientes
 - Función: `public function index() {...}`
 - Muestra todos los clientes.
- Crear Cliente:
 - Método: POST
 - Ruta: /clientes/create
 - Función: `public function create() {...}`
 - Muestra el formulario para crear un cliente.
- Guardar Cliente:
 - Método: POST
 - Ruta: /clientes/store
 - Función: `public function store() {...}`
 - Guarda un nuevo cliente en la base de datos.
- Editar Cliente:
 - Método: GET
 - Ruta: /clientes/edit/:id
 - Función: `public function edit($id) {...}`
 - Muestra el formulario para editar un cliente.

- Actualizar Cliente:
 - Método: POST
 - Ruta: /clientes/update/:id
 - Función: public function update(\$id) {...}
 - Actualiza un cliente en la base de datos.
- Eliminar Cliente:
 - Método: DELETE (o GET si no usas AJAX)
 - Ruta: /clientes/delete/:id
 - Función: public function delete(\$id) {...}
 - Elimina un cliente de la base de datos.

5.2 CRUD de Dashboard (CodeIgniter)

Este controlador gestiona las operaciones relacionadas con las salas, clientes, paquetes y estadísticas.

- Mostrar Dashboard:
 - Método: GET
 - Ruta: /dashboard
 - Función: public function index() {...}
 - Muestra la vista principal del dashboard con todas las salas.
- Actualizar Estado de Sala:
 - Método: POST
 - Ruta: /dashboard/actualizar_estado_sala
 - Función: public function actualizar_estado_sala() {...}
 - Actualiza el estado de una sala (ocupado o libre) en la base de datos.
- Ver Detalles de Sala:
 - Método: GET
 - Ruta: /dashboard/sala/:id
 - Función: public function sala(\$id) {...}
 - Muestra los detalles de una sala específica, junto con la lista de clientes y paquetes disponibles.
- Obtener Precio del Paquete:
 - Método: POST
 - Ruta: /dashboard/obtener_precio_paquete
 - Función: public function obtener_precio_paquete() {...}
 - Devuelve el precio por hora de un paquete seleccionado en formato JSON.

5.3 CRUD de LOGIN (CodeIgniter)

Controlador: `application/controllers/Login.php`

Este controlador maneja las operaciones de autenticación de usuarios.

- Mostrar Formulario de Login:
 - Método: GET
 - Ruta: /login

- Función: public function index() {...}
 - Muestra la vista del formulario de inicio de sesión.
- Autenticar Usuario:
 - Método: POST
 - Ruta: /login/autenticar
 - Función: public function autenticar() {...}
 - Verifica las credenciales del usuario (nombre de usuario y contraseña).
 - Si son correctas, redirige al dashboard (/dashboard).
 - Si no, muestra un mensaje de error en la página de login.

5.4 CRUD de Paquetes (CodeIgniter)

Controlador: ``application/controllers/Paquetes.php``

Este controlador maneja las operaciones CRUD de los paquetes.

- Listar Paquetes:
 - Método: GET
 - Ruta: /paquetes
 - Función: public function index() {...}
 - Muestra una lista de todos los paquetes.
- Crear Paquete:
 - Método: GET
 - Ruta: /paquetes/create
 - Función: public function create() {...}
 - Muestra el formulario para crear un nuevo paquete.
- Guardar Paquete:
 - Método: POST
 - Ruta: /paquetes/store
 - Función: public function store() {...}
 - Valida y guarda los datos de un nuevo paquete en la base de datos.
- Editar Paquete:
 - Método: GET
 - Ruta: /paquetes/edit/:id
 - Función: public function edit(\$id) {...}
 - Muestra el formulario para editar un paquete específico.
- Actualizar Paquete:
 - Método: POST
 - Ruta: /paquetes/update/:id
 - Función: public function update(\$id) {...}
 - Valida y actualiza los datos de un paquete en la base de datos.
- Eliminar Paquete:
 - Método: DELETE (o GET si no usas AJAX)
 - Ruta: /paquetes/delete/:id
 - Función: public function delete(\$id) {...}
 - Elimina un paquete específico de la base de datos.

5.5 Cronómetro de Uso de Salas (CodeIgniter)

Controlador: ``application/controllers/Salas.php``

Este controlador maneja las operaciones CRUD de las salas.

- Listar Salas:
 - Método: GET
 - Ruta: /salas
 - Función: public function index() {...}
 - Muestra una lista de todas las salas.
- Crear Sala:
 - Método: GET
 - Ruta: /salas/create
 - Función: public function create() {...}
 - Muestra el formulario para crear una nueva sala.
- Guardar Sala:
 - Método: POST
 - Ruta: /salas/store
 - Función: public function store() {...}
 - Guarda los datos de una nueva sala en la base de datos.
- Editar Sala:
 - Método: GET
 - Ruta: /salas/edit/:id
 - Función: public function edit(\$id) {...}
 - Muestra el formulario para editar una sala específica.
- Actualizar Sala:
 - Método: POST
 - Ruta: /salas/update/:id
 - Función: public function update(\$id) {...}
 - Actualiza los datos de una sala en la base de datos.
- Eliminar Sala:
 - Método: DELETE (o GET si no usas AJAX)
 - Ruta: /salas/delete/:id
 - Función: public function delete(\$id) {...}
 - Elimina una sala específica de la base de datos.

5.6 Cronómetro de Usuarios (CodeIgniter)

Controlador: ``application/controllers/Usuarios.php``

Este controlador maneja las operaciones CRUD de los usuarios.

- Listar Usuarios:
 - Método: GET
 - Ruta: `/usuarios`
 - Función: `public function index() {...}`
 - Muestra una lista de todos los usuarios.
- Crear Usuario:
 - Método: GET
 - Ruta: `/usuarios/create`
 - Función: `public function create() {...}`
 - Muestra el formulario para crear un nuevo usuario.
- Guardar Usuario:
 - Método: POST
 - Ruta: `/usuarios/store`
 - Función: `public function store() {...}`
 - Guarda los datos de un nuevo usuario en la base de datos (nombre y contraseña).
- Editar Usuario:
 - Método: GET
 - Ruta: `/usuarios/edit/:id`
 - Función: `public function edit($id) {...}`
 - Muestra el formulario para editar un usuario específico.
- Actualizar Usuario:
 - Método: POST

- Ruta: /usuarios/update/:id
- Función: public function update(\$id) {...}
- Actualiza los datos de un usuario (nombre y contraseña) en la base de datos.
- Eliminar Usuario:
 - Método: DELETE (o GET si no usas AJAX)
 - Ruta: /usuarios/delete/:id
 - Función: public function delete(\$id) {...}
 - Elimina un usuario específico de la base de datos.

5.7 Cronómetro de Usuarios (CodeIgniter)

Controlador: `application/controllers/Usuarios.php`

6. Lógica del cronómetro.

application/views/dashboard/sala.php

El código corresponde a una página web que muestra el detalle de una sala donde se pueden gestionar cronómetros o cuentas regresivas y calcular el costo del uso de la sala según el tiempo transcurrido.

Análisis de cada elemento:

1. Encabezado (<head>)

- Meta etiquetas:
 - charset="UTF-8": Define la codificación de caracteres como UTF-8.
 - viewport="width=device-width, initial-scale=1.0": Garantiza que la página sea responsiva en dispositivos móviles.
- Hoja de estilos:

Incluye la hoja de estilos de Bootstrap para utilizar componentes y estilos predefinidos, y un bloque de estilos CSS para personalizar el aspecto del cronómetro.

2. Cuerpo (<body>)

- Formulario: Se usa para seleccionar el cliente, paquete y modo (cronómetro o cuenta regresiva). Además, el campo tiempo se muestra solo cuando el modo seleccionado es "cuenta regresiva".
 - Cliente (<select id="cliente">): Despliega una lista de clientes.
 - Paquete (<select id="paquete">): Muestra los paquetes disponibles con su precio por hora.
 - Modo (<select id="modo">): Permite elegir entre "cronómetro" y "cuenta regresiva".
 - Tiempo (<input id="tiempo">): Se ingresa el tiempo en minutos si se selecciona "cuenta regresiva".

3. Contenedor del reloj (#contenedor)

- Visualización del cronómetro o cuenta regresiva con tres divs para horas, minutos y segundos.

4. Botones de control

- Iniciar (#inicio): Activa el cronómetro o la cuenta regresiva según el modo seleccionado.
- Detener (#parar): Pausa el cronómetro o la cuenta regresiva.
- Reanudar (#continuar): Continúa el cronómetro o la cuenta regresiva pausada.
- Reiniciar (#reinicio): Restablece el tiempo a "00:00:00".
- Liberar Sala (#liberar): Calcula el costo total del tiempo transcurrido según el paquete seleccionado y limpia el estado de la sala.

5. Lógica de scripts

- addEventListener("change",...): Cambia entre mostrar u ocultar el campo de tiempo según el modo seleccionado (cronómetro o cuenta regresiva).
- DOMContentLoaded: Al cargar la página, recupera el estado guardado de la sala desde el localStorage para reanudar el cronómetro o cuenta regresiva si estaba activo.
- Funciones principales:
 - guardarEstadoSala: Guarda el estado actual de la sala en localStorage (tiempo, modo, etc.).

- bloquearFormulario: Desactiva el formulario mientras el cronómetro está en marcha.
- desbloquearFormulario: Reactiva el formulario cuando el cronómetro se detiene o reinicia.
- liberarSala: Calcula el costo total del tiempo usado basándose en el paquete seleccionado y el modo, muestra un resumen con SweetAlert2 y borra el estado guardado de la sala.
- cronometro: Incrementa el tiempo en segundos y actualiza los valores mostrados (horas, minutos y segundos).
- cuentaRegresiva: Decrementa el tiempo restante y finaliza la cuenta regresiva cuando llega a 0.
- inicio: Inicia el cronómetro o la cuenta regresiva según el modo.
- parar: Pausa el cronómetro o la cuenta regresiva.
- continuar: Reanuda el cronómetro o la cuenta regresiva desde donde se dejó.
- reinicio: Resetea el cronómetro o cuenta regresiva a "00:00:00".

6. Librerías externas

- JQuery: Se utiliza para facilitar el manejo de eventos y la manipulación del DOM.
- Bootstrap JS: Para los componentes interactivos como el formulario.
- SweetAlert2: Muestra alertas y mensajes amigables, como el resumen de costos y el tiempo agotado.

7. Mantenimiento y Actualización

Backup de la Base de Datos: Realizar copias de seguridad periódicas de la base de datos.

8. Solución de Problemas Comunes

Error de Base de Datos: Verifica que la ruta a la base de datos este correcta.