

UNIVERSIDAD AUTÓNOMA GABRIEL RENE MORENO
FACULTAD DE INGENIERÍA EN CIENCIAS DE LA COMPUTACIÓN Y
TELECOMUNICACIONES



Primer Parcial

INTEGRANTES:

- Antezana Espinoza Ruth Veronica

REGISTRO:

- 218061722

CARRERA:

- Ingeniería Informática

MATERIA:

- Arquitectura de Software

DOCENTE:

- Ing. Josué Obed Veizaga Gonzales

2024

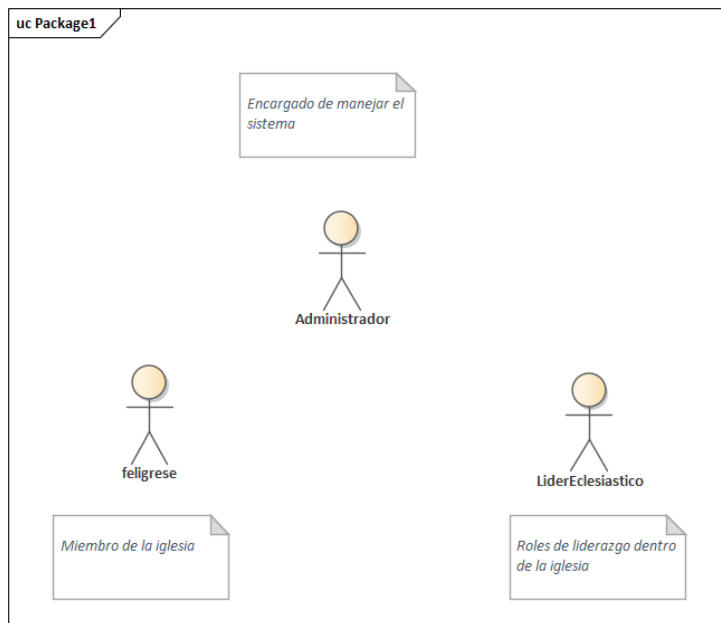
Santa Cruz de la Sierra – Bolivia

Contenido

1 FLUJO: REQUISITOS	3
1.1 IDENTIFICAR ACTORES	3
1.2 IDENTIFICAR CASOS DE USO	3
1.3 DETALLE DE CASO DE USO	4
1.4 DIAGRAMA GENERAL DE CASO DE USO	12
REQUISITOS NO FUNCIONALES	12
2. FLUJO DE ANALISIS DE LA ARQUITECTURA	13
2.1 IDENTIFICACION DE MODULOS	13
2.2 VISTA DE PAQUETES	13
2.3 ENCAPSULAMIENTO DE CASOS DE USO	15
3. FLUJO DISEÑO	16
3.1 DISEÑO DE LA ARQUITECTURA DE SOFTWARE	16
3.2 DISEÑO DE LA BASE DE DATOS	18
3.2.1 DISEÑO CONCEPTUAL	18
3.2.2 DISEÑO LOGICO	18
3.2.3 DISEÑO FISICO	19
3.2.3 DISEÑO DE DETALLE PROCEDIMENTAL	20
3.2.4 DIAGRAMAS DE SECUENCIA	23
4. CODIFICACION	26
CU1: Gestionar Cargo	26
CU2: Gestionar Usuario	39
CU3: Gestionar Tipo relación	53
CU5: Gestionar Evento	77
CU6: Gestionar Ministerio	88
CU7: Registro de Evento	99

1 FLUJO: REQUISITOS

1.1 IDENTIFICAR ACTORES



1.2 IDENTIFICAR CASOS DE USO

CU1: Gestionar Cargo

CU2: Gestionar Usuario

CU3: Gestionar Tipo de Relación

CU4: Gestionar Parentesco

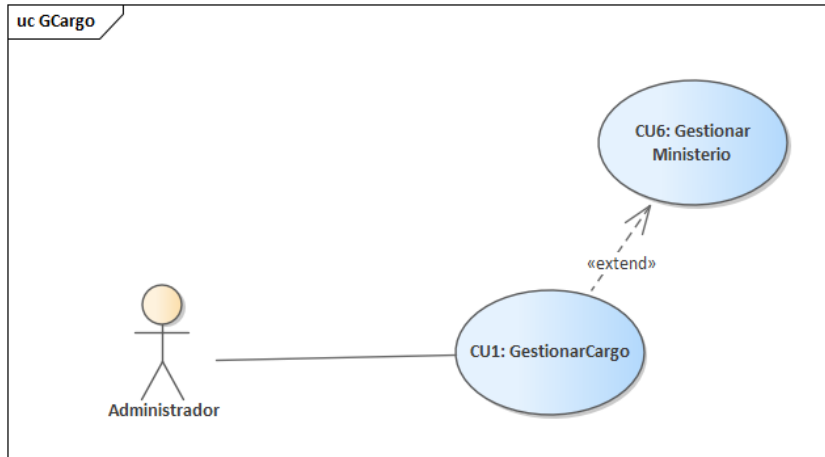
CU5: Gestionar Evento

CU6: Gestionar Ministerio

CU7: Gestionar Registro_Evento

1.3 DETALLE DE CASO DE USO

CU1. GESTIONAR CARGO



CASO DE USO	CU1 Gestionar Cargo.
Propósito	Gestionar Cargos
Descripción	El administrador podrá registrar, editar y eliminar los registros de los cargos que se manejan en la iglesia.
Actores	Administrador
Actor Iniciador	Administrador
Precondición	Ninguna
Proceso	Registrar Cargo El administrador debería poder registrar nuevos cargos en el sistema ingresando los siguientes datos: <ul style="list-style-type: none">• Nombre• Descripción Editar Cargo El administrador debería poder editar registros de los cargos que hay en el sistema. Eliminar Cargo El administrador podrá eliminar los cargos que ya estan fuera de uso en la iglesia.
Postcondición	Cargo registrado, editado o eliminado.
Excepciones	Campos vacíos.

Iglesia

Inicio
 Usuario
 Cargo
 Ministerio
 Evento
 Registro Evento
 Tipo Reduccion
 Parametro

Nuevo Cargo

Nombre

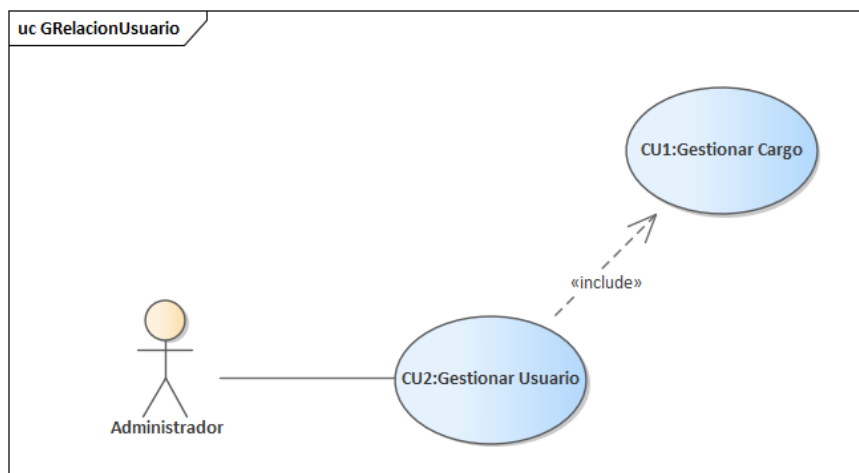
Descripcion

Ministerio

Cargos

Id	Nombre	Descripcion	Ministerio		
1	Presidente	Presidente	Danza	<input type="button" value="Editar"/>	<input type="button" value="Eliminar"/>
2	Obispo	desd	-	<input type="button" value="Editar"/>	<input type="button" value="Eliminar"/>

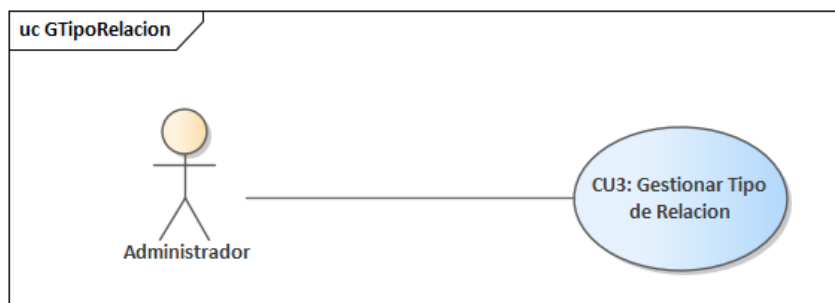
CU2. Gestionar Usuario



CASO DE USO	CU2 Gestionar Usuario.
Propósito	Gestionar Usuarios
Descripción	El administrador podrá registrar, editar y eliminar los registros de los usuarios que se manejan en la iglesia.
Actores	Administrador, Líder Eclesiástico, Feligreses
Actor Iniciador	Administrador
Precondición	Ninguna
Proceso	<p>Registrar Usuario El administrador debería poder registrar nuevos usuarios en el sistema ingresando los siguientes datos:</p> <ul style="list-style-type: none"> • Nombre • Apellido • Teléfono • Ci • Teléfono <p>Editar Usuario</p>

	<p>El administrador debería poder editar registros de los usuarios que hay en el sistema.</p> <p>Eliminar Usuario</p> <p>El administrador podrá eliminar a los miembros de la iglesia, tanto a los feligreses como a los líderes.</p>
Postcondición	Usuario registrado, editado o eliminado, asignado con un cargo y/o ministerio.
Excepciones	Campos vacíos.

CU3: Gestionar Tipo de relación



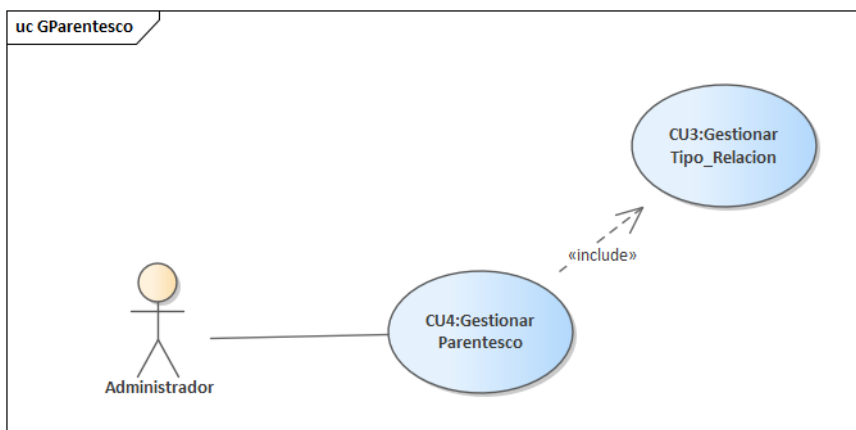
CASO DE USO	CU3 Gestionar Tipo Relación.
Propósito	Gestionar Tipo Relación
Descripción	El administrador podrá registrar, editar y eliminar los tipos de relaciones que hay en la iglesia o que se manejan en la iglesia.
Actores	Administrador
Actor Iniciador	Administrador
Precondición	Ninguna
Proceso	<p>Registrar Tipo Relación</p> <p>El administrador debería poder registrar nuevos tipos de relaciones que se manejan hoy en día o que la iglesia maneja, ingresando los siguientes datos:</p> <ul style="list-style-type: none"> Nombre

	<p>Editar Tipo Relación El administrador debería poder editar registros de los tipos de relaciones que hay en el sistema.</p> <p>Eliminar Tipo de Relación El administrador podrá eliminar a los tipos de relación de la iglesia.</p>
Pos condición	Tipo de Relación registrado, editado o eliminado.
Excepciones	Campos vacíos.

The screenshot shows a web application interface for a church. On the left is a sidebar menu with options: Inicio, Usuario, Cargo, Ministerio, Donatio, Registro Eventos, Tipo Relacion, and Parentesco. The main content area is titled 'Iglesia' and contains a form for 'Nuevo Tipo Relacion' with a 'Nombre' input field and an 'Agregar' button. Below the form is a table titled 'Tipo Relación - Usuarios'.

M	Nombre		
1	Hermano	Editar	Eliminar
2	Padre	Editar	Eliminar

CU4: Gestionar Parentesco



CASO DE USO	CU3 Gestionar Parentesco.
Propósito	Gestionar Parentesco
Descripción	El administrador podrá registrar, editar y eliminar los parentescos que hay en la iglesia o que se manejan en la iglesia.
Actores	Administrador
Actor Iniciador	Administrador
Precondición	Ninguna
Proceso	Registrar Parentesco

	<p>El administrador debería poder registrar los parentescos seleccionando los usuarios que estan registrados en el sistema.</p> <p>Editar Parentesco El administrador debería poder editar los parentescos que hay en el sistema.</p> <p>Eliminar Parentesco El administrador podrá eliminar a los parentescos de la iglesia.</p>
Pos condición	Tipo de Parentesco registrado, editado o eliminado.
Excepciones	Campos vacíos.

Iglesia App

Inicio
Usuario
Cargo
Ministerio
Evento
Registro Eventos
Tipo Relacion
Parentesco

Nuevo Parentesco

Usuario A: Paul Torrez

Usuario B: Paul Torrez

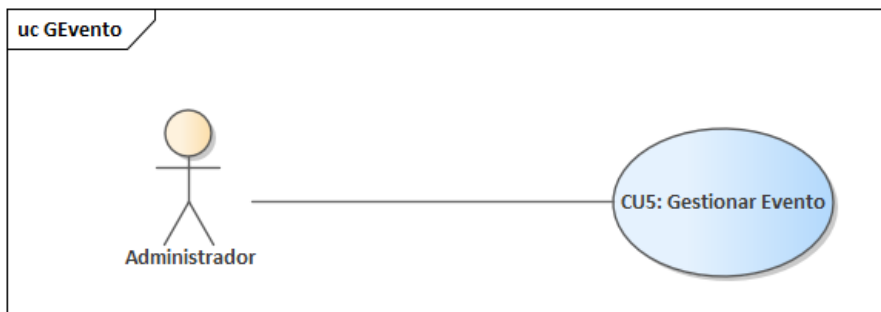
Tipo de Relación A: Hermano

Agregar Volver

Parentescos

Usuario_A	Usuario_B	Tipo Relacion		
Carla Torrez	Paul Torrez	Hermano	Editar	Eliminar
Paul Torrez	Maria Escalante	Padre	Editar	Eliminar

CU5: Gestionar Evento

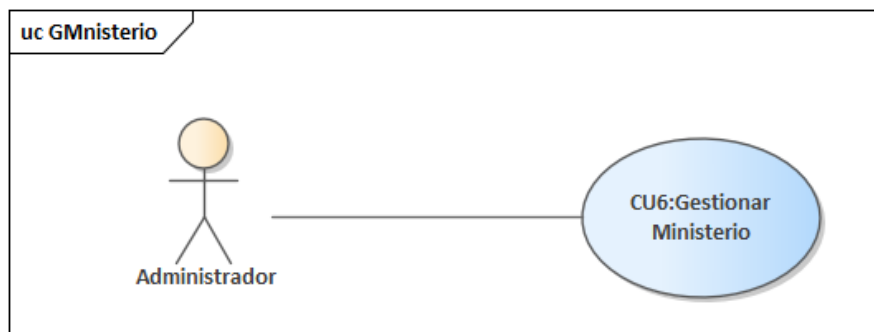


CASO DE USO	CU5 Gestionar Evento.
Propósito	Gestionar Evento
Descripción	El administrador podrá registrar, editar y eliminar los tipos de eventos que se manejan en la iglesia.
Actores	Administrador
Actor Iniciador	Administrador
Precondición	Ninguna
Proceso	<p>Registrar Evento El administrador debería poder registrar nuevos usuarios en el sistema ingresando los siguientes datos:</p> <ul style="list-style-type: none"> Nombre

	<ul style="list-style-type: none"> Descripción Editar Evento El administrador debería poder editar registros de los eventos que hay en el sistema. Eliminar Evento El administrador podrá eliminar a los eventos que se realizan en la iglesia
Postcondición	Evento registrado, editado o eliminado.
Excepciones	Campos vacíos.

Id	Nombre	Descripción
1	Bautizo	Bautizo
2	Matrimonio	Union de dos personas

CU6: Gestionar Ministerio



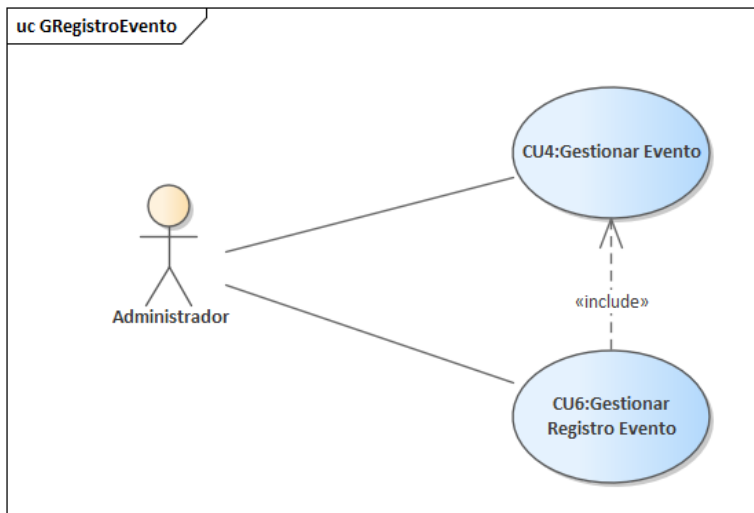
CASO DE USO	CU6 Gestionar Ministerio.
Propósito	Gestionar Ministerio
Descripción	El administrador podrá registrar, editar y eliminar los registros de los ministerios que hay en la iglesia.
Actores	Administrador
Actor Iniciador	Administrador
Precondición	Ninguna
Proceso	Registrar Ministerio El administrador debería poder registrar nuevos ministerios en el sistema ingresando los siguientes datos: <ul style="list-style-type: none"> Nombre

	<p>Editar Ministerio El administrador debería poder editar registros de los ministerios que hay en el sistema.</p> <p>Eliminar Ministerio El administrador podrá eliminar los ministerios , tanto a los feligreses como a los líderes.</p>
Postcondición	Ministerio registrado, editado o eliminado.
Excepciones	Campos vacíos.

The screenshot shows a web application titled 'Iglesia'. On the left is a sidebar menu with options: Inicio, Usuario, Cargo, Ministerio, Evento, Registro Eventos, Tipo Relacion, and Parametros. The main content area has a header 'Iglesia' and a section 'Nuevo Ministerio' with input fields for 'Nombre' and 'Descripción', and an 'Agregar' button. Below this is a table titled 'Ministerios' with columns 'M', 'Nombre', and 'Descripción'. The table contains two rows of data. Each row has 'Editar' and 'Eliminar' buttons to its right.

M	Nombre	Descripción
2	-	-
3	Danza	Ministerio Danza

CU7: Gestionar Registro Evento



CASO DE USO	CU7 Gestionar Registro Evento.
Propósito	Gestionar Registro Evento
Descripción	El administrador podrá registrar, editar y eliminar los registros de los eventos que se realizan en la iglesia.
Actores	Administrador
Actor Iniciador	Administrador
Precondición	Evento

Proceso	Registrar Registro de Evento El administrador debería poder registrar nuevos eventos que se realizaran en la iglesia ingresando los siguientes datos: <ul style="list-style-type: none">• Descripción• Fecha• lugar Editar Registro Evento El administrador debería poder editar registros de los ministerios que hay en el sistema. Eliminar Registro Evento El administrador podrá eliminar los ministerios , tanto a los feligreses como a los líderes.
Postcondición	Registro de evento registrado, editado o eliminado.
Excepciones	Campos vacíos.

Iglesia App

Inicio

Usuarios

Cargos

Ministerios

Eventos

Registro Eventos

Tipo Refecciones

Parroquianos

Nuevo Registro Evento

Lugar

Nota

Fecha

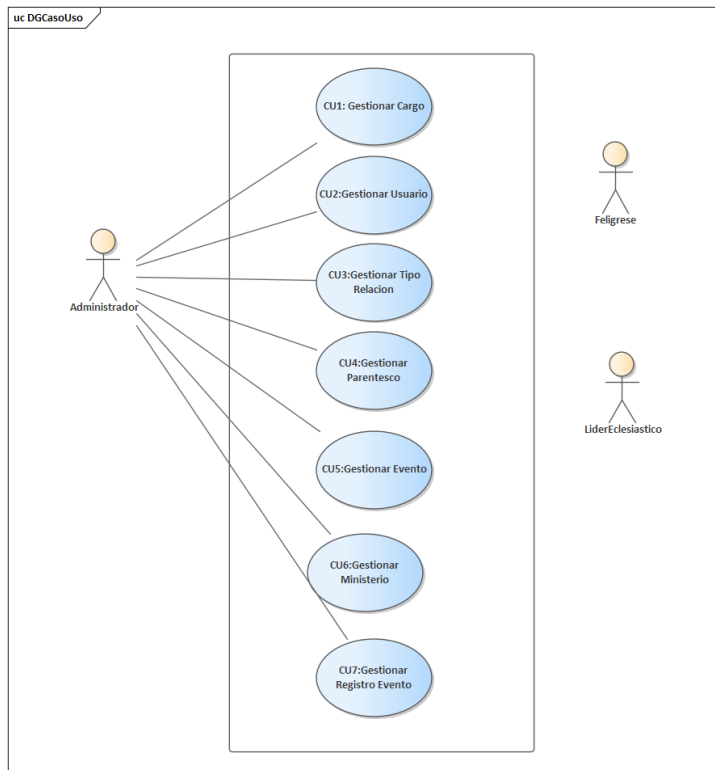
Evento

Registrar

Registro Eventos

Id	Lugar	Nota	Fecha	Evento				
1	Iglesia Manzana 80	Incluye	2024-04-30	Bautizo	Registrar Evento	Ver Usuarios	Eliminar	Eliminar

1.4 DIAGRAMA GENERAL DE CASO DE USO



REQUISITOS NO FUNCIONALES

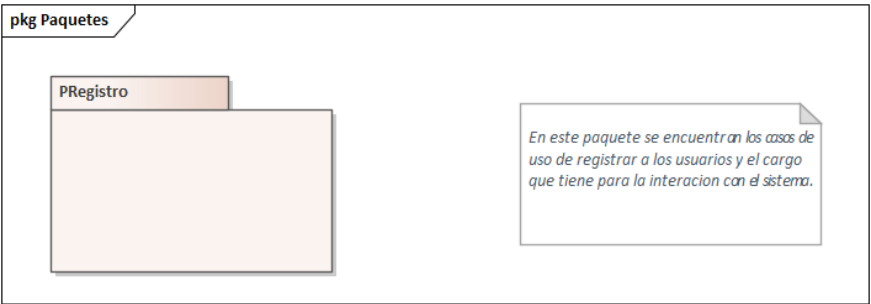
Este sistema web contara con las siguientes características

- **Usabilidad**
 - Navegación intuitiva: facilidad de uso del sistema, con una navegación clara e intuitiva para los usuarios.
- **Rendimiento**
 - Tiempo de carga: El sistema cumple con la carga rápida para proporcionar una experiencia de usuario fluida.
 - Escalabilidad: Capacidad del sistema para manejar un aumento en la carga de usuarios y transacciones sin degradación del rendimiento.
 - Disponibilidad: Garantiza que el sistema esté disponible y accesible para los usuarios durante todo el tiempo necesario.
- **Mantenibilidad**
 - Claridad del código: código bien estructurado y comentado para facilitar el mantenimiento futuro.
 - Facilidad de actualización: El sistema puede ser actualizado fácilmente para incorporar nuevas funcionalidades o corregir errores.

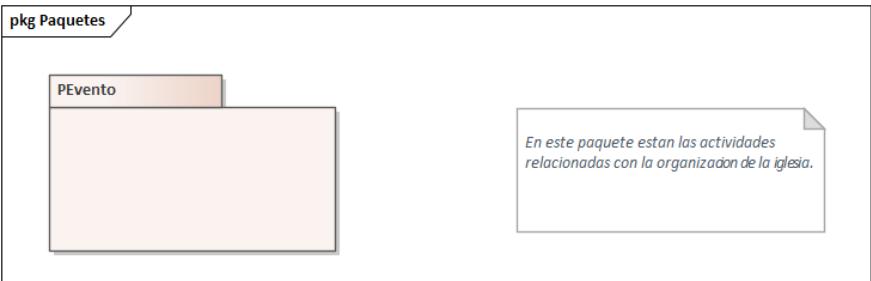
2. FLUJO DE ANALISIS DE LA ARQUITECTURA

2.1 IDENTIFICACION DE MODULOS

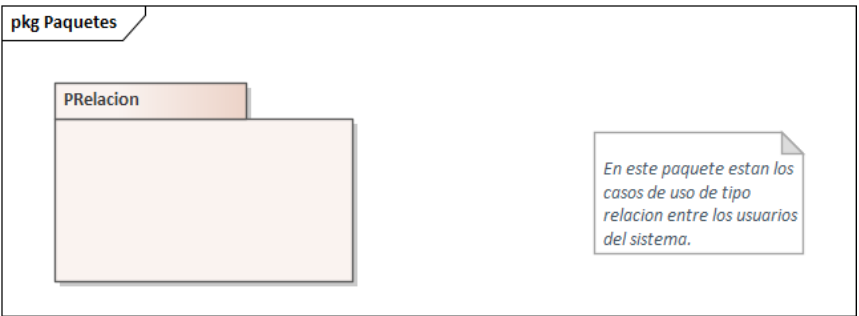
PAQUETE 1: Registro



PAQUETE 2: Evento

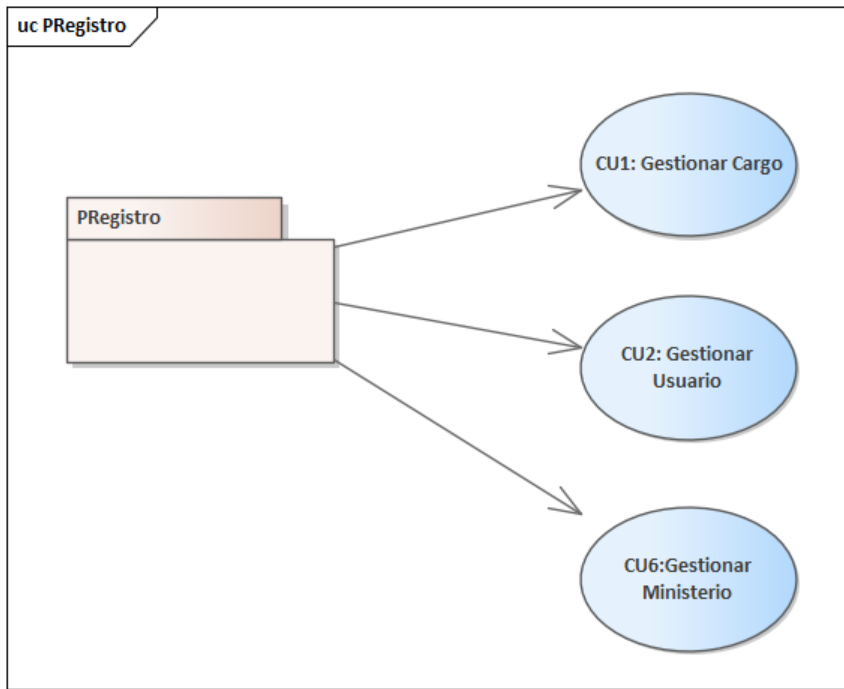


PAQUETE 3: Relación

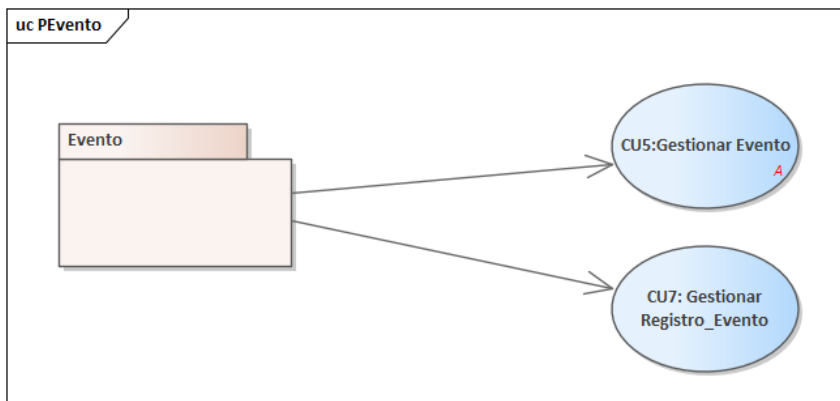


2.2 VISTA DE PAQUETES

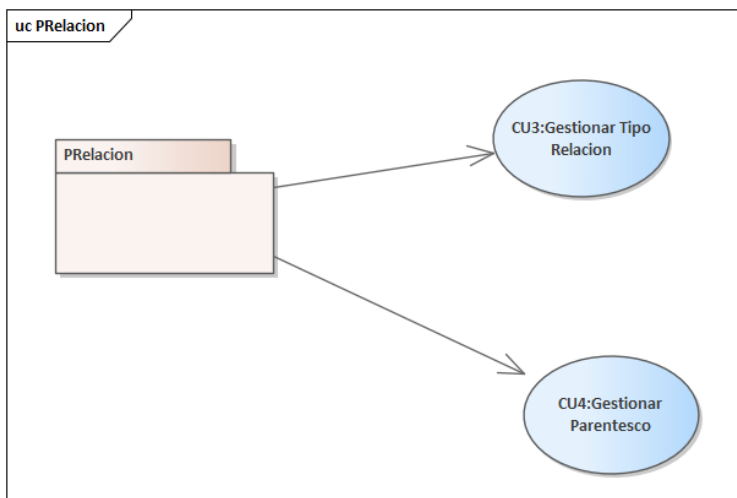
Paquete 1: Registro



Paquete 2: Evento

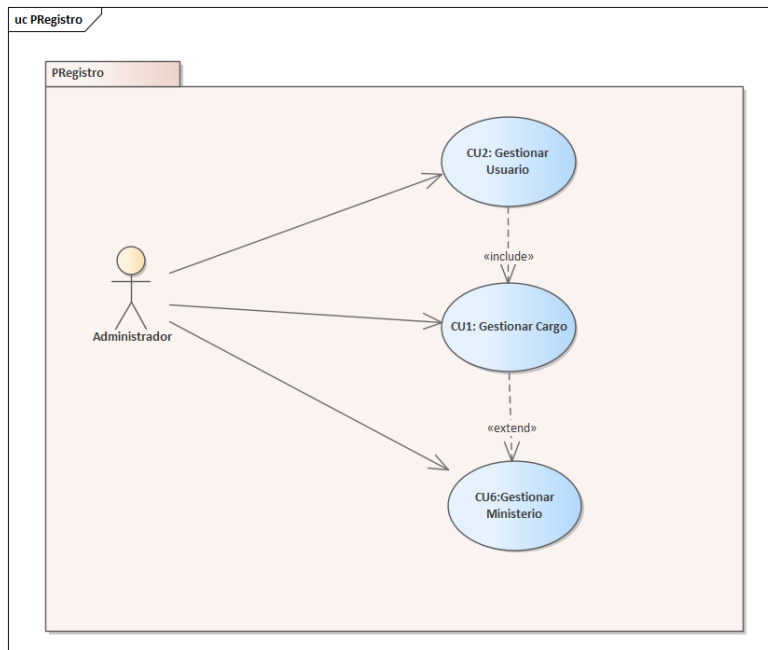


Paquete 3: Relación

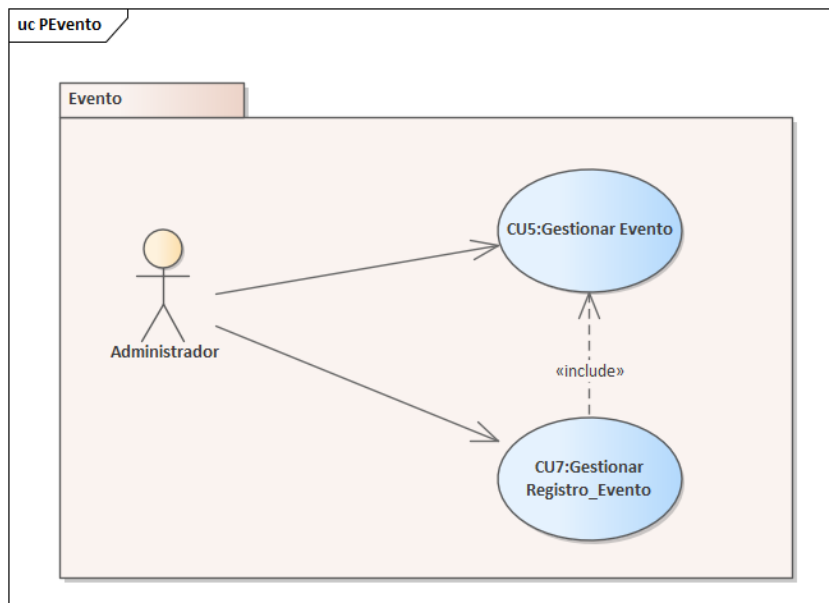


2.3 ENCAPSULAMIENTO DE CASOS DE USO

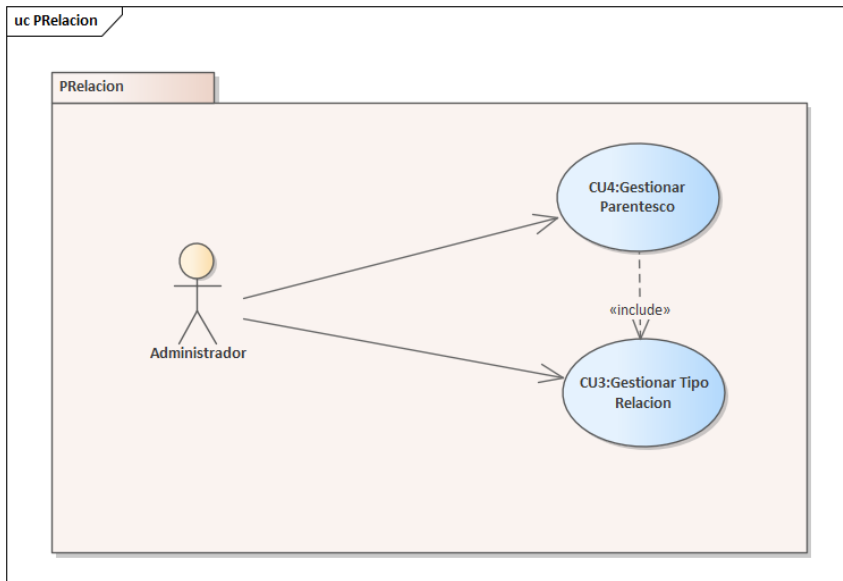
Paquete 1: Registro



Paquete 2: Evento



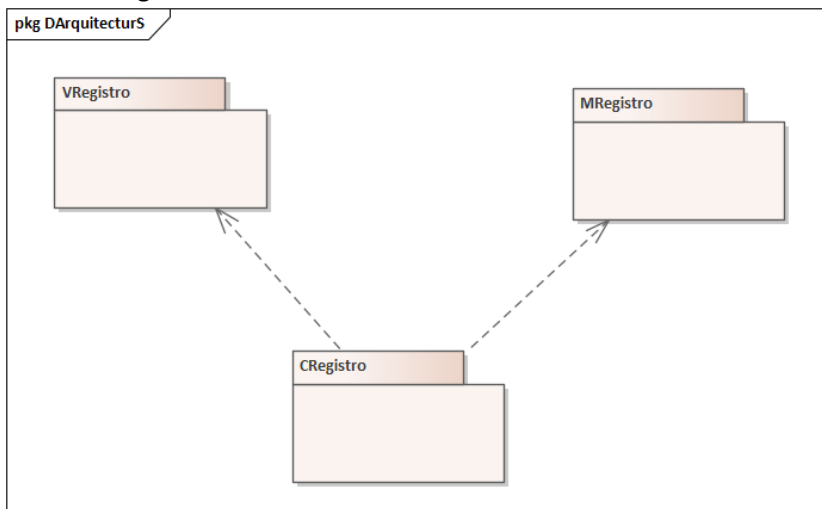
Paquete 3: Relacion



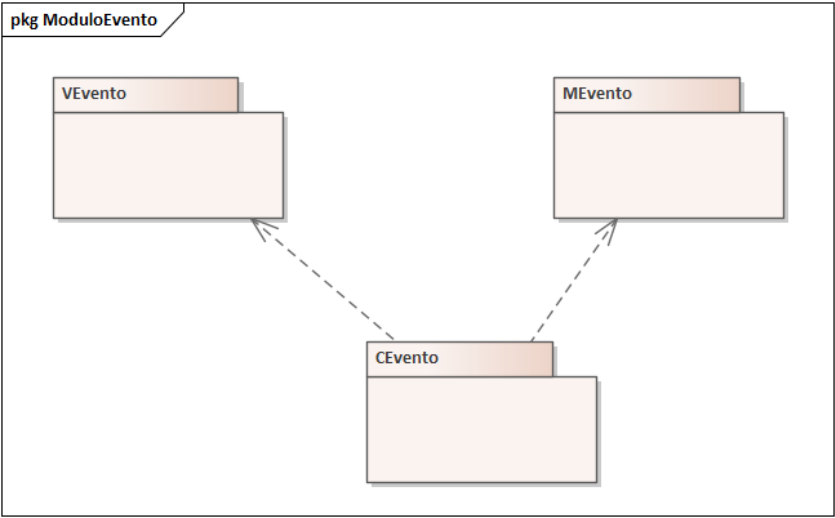
3. FLUJO DISEÑO

3.1 DISEÑO DE LA ARQUITECTURA DE SOFTWARE

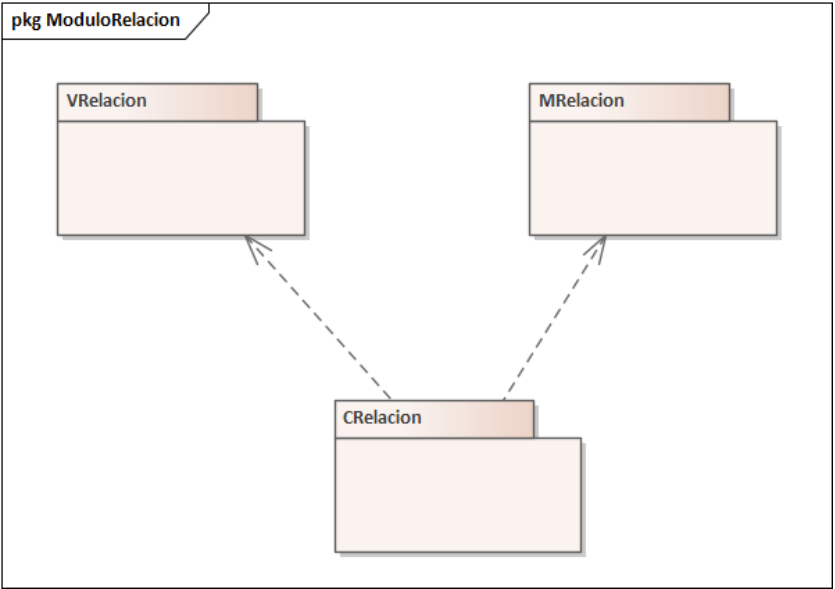
Modulo Registro



Modulo Evento

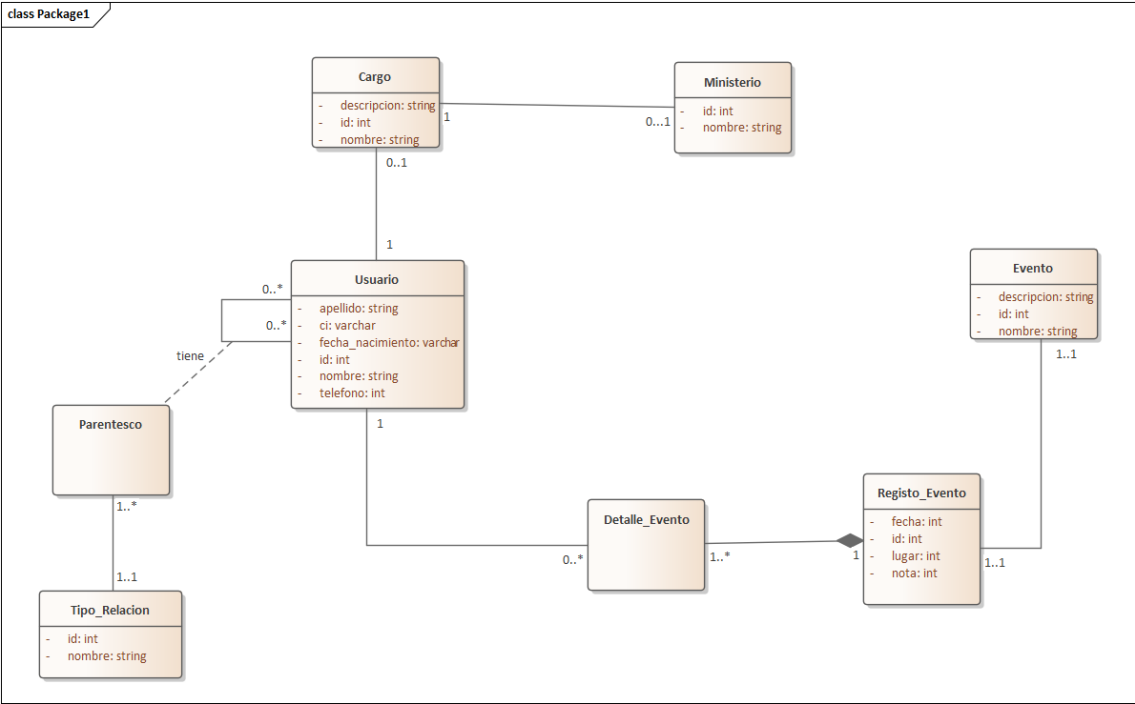


Modulo Relación



3.2 DISEÑO DE LA BASE DE DATOS

3.2.1 DISEÑO CONCEPTUAL



3.2.2 DISEÑO LOGICO

Ministerio		
id	nombre	descripción
PK		

Cargo			
id	nombre	descripción	ministerio_id
PK			FK

Usuario						
id	nombre	apellido	fecha_nacimiento	ci	telefono	cargo_id
PK						FK

Tipo_Relacion	
id	nombre
PK	

Parentesco	
usuario_id	tipo_relacion_id

Evento		
id	nombre	descripción
PK		

Registro_Evento				
id	lugar	nota	fecha	evento_id
PK				FK

Detalle_Evento	
usuario_id	registro_evento_id
PK/FK	PK/FK

3.2.3 DISEÑO FISICO

```
$createTableQueryMinisterios = " CREATE TABLE IF NOT EXISTS " . self::TABLE_MINISTERIO . " (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(255),
    descripcion VARCHAR(255)
)";
```

```
$createTableQueryCargos = " CREATE TABLE IF NOT EXISTS " . self::TABLE_CARGO . " (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(255),
    descripcion VARCHAR(255),
    ministerio_id INT,
    FOREIGN KEY (ministerio_id) REFERENCES " . self::TABLE_MINISTERIO . "(id)
)";
```

```
$createTableQueryTipoRelaciones = " CREATE TABLE IF NOT EXISTS " .
self::TABLE_TIPO_RELACION . " (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(255)
)";
```

```
$createTableQueryUsuarios = " CREATE TABLE IF NOT EXISTS " . self::TABLE_USUARIO . " (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nombre VARCHAR(255),
    apellido VARCHAR(255),
    fechanacimiento DATE,
    ci VARCHAR(255),
    telefono INT,
    cargo_id INT,
    FOREIGN KEY (cargo_id) REFERENCES " . self::TABLE_CARGO . "(id)
)";
```

```
$createTableQueryParentesco = " CREATE TABLE IF NOT EXISTS
" . self::TABLE_PARENTESCO . " (
    id INT AUTO_INCREMENT PRIMARY KEY,
```

```

usuario_a INT,
usuario_b INT,
tipo_relacion_a INT,
FOREIGN KEY (usuario_a) REFERENCES " . self::TABLE_USUARIO . "(id),
FOREIGN KEY (usuario_b) REFERENCES " . self::TABLE_USUARIO . "(id),
FOREIGN KEY (tipo_relacion_a) REFERENCES " . self::TABLE_TIPO_RELACION . "(id)
)";

```

```

$createTableQueryEventos = " CREATE TABLE IF NOT EXISTS " . self::TABLE_EVENTO . " (
id INT AUTO_INCREMENT PRIMARY KEY,
nombre VARCHAR(255),
descripcion VARCHAR(255)
)";

```

```

$createTableQueryRegistroEvento = " CREATE TABLE IF NOT EXISTS " .
self::TABLE_REGISTRO_EVENTO . " (
id INT AUTO_INCREMENT PRIMARY KEY,
lugar VARCHAR(255),
nota VARCHAR(255),
fecha DATE,
evento_id INT,
FOREIGN KEY (evento_id) REFERENCES " . self::TABLE_EVENTO . "(id)
)";

```

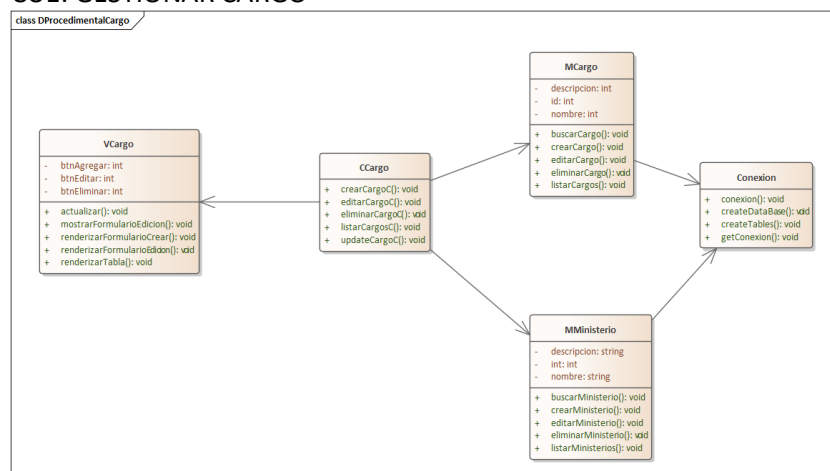
```

$createTableQueryDetalleEvento = " CREATE TABLE IF NOT EXISTS " .
self::TABLE_DETALLE_EVENTO . " (
id INT AUTO_INCREMENT PRIMARY KEY,
usuario INT,
registro_evento INT,
FOREIGN KEY (usuario) REFERENCES " . self::TABLE_USUARIO . "(id),
FOREIGN KEY (registro_evento) REFERENCES " . self::TABLE_REGISTRO_EVENTO . "(id)
)";

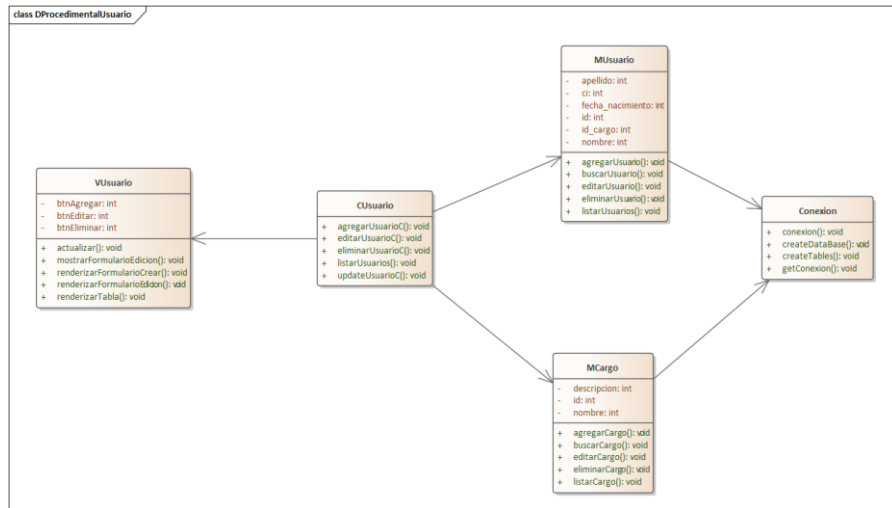
```

3.2.3 DISEÑO DE DETALLE PROCEDIMENTAL

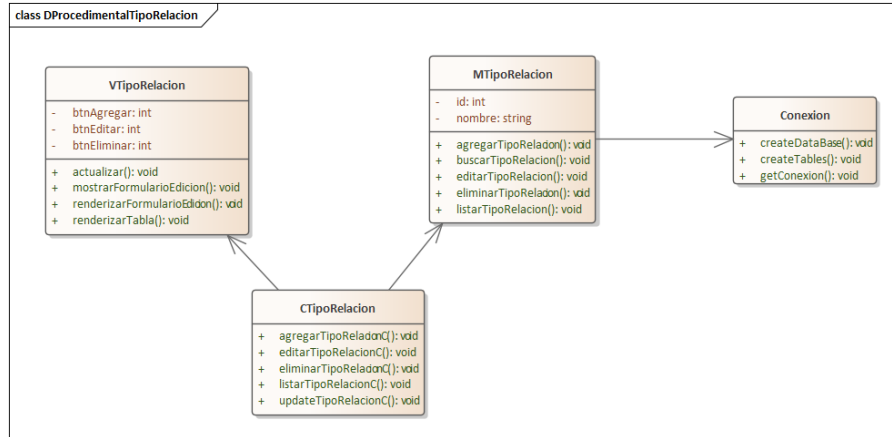
CU1: GESTIONAR CARGO



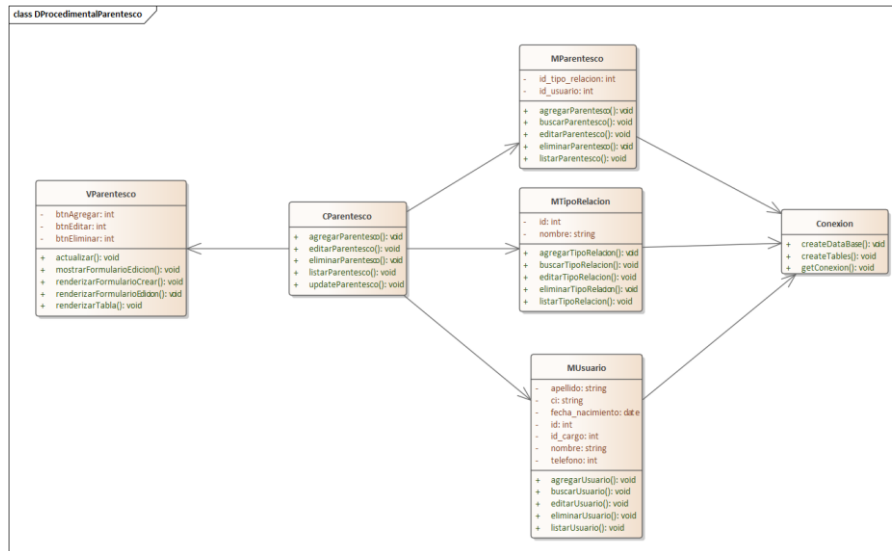
CU2: GESTIONAR USUARIO



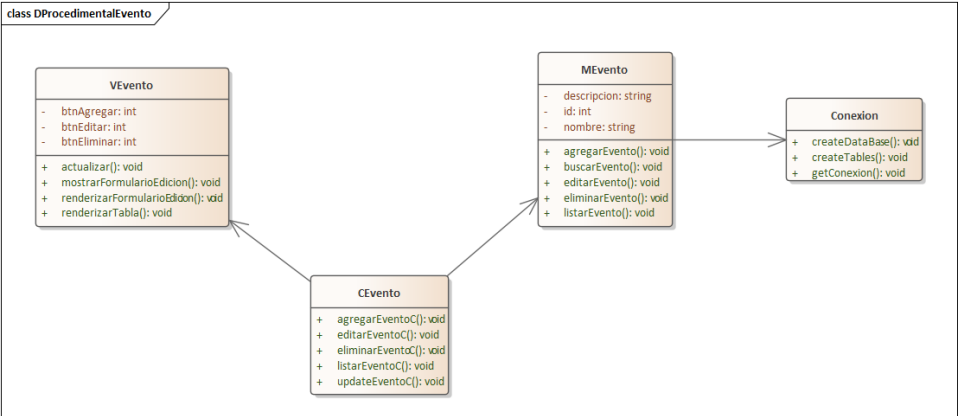
CU3: GESTIONAR TIPO RELACION



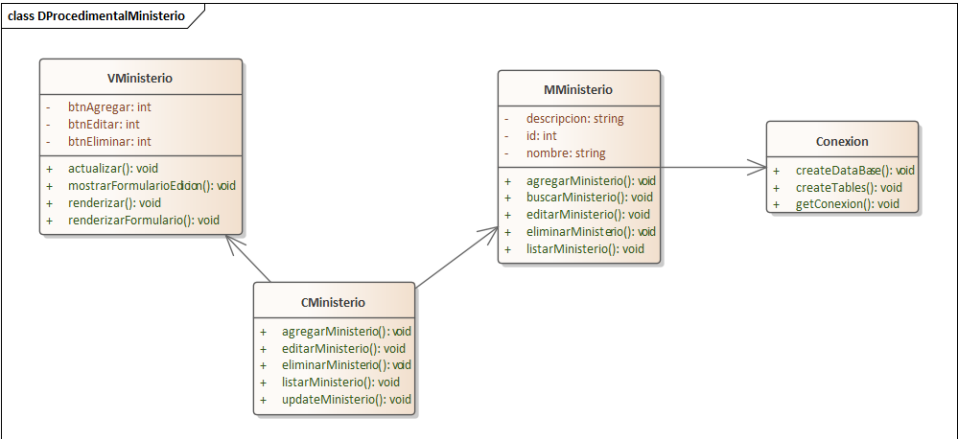
CU4: GESTIONAR PARENTESCO



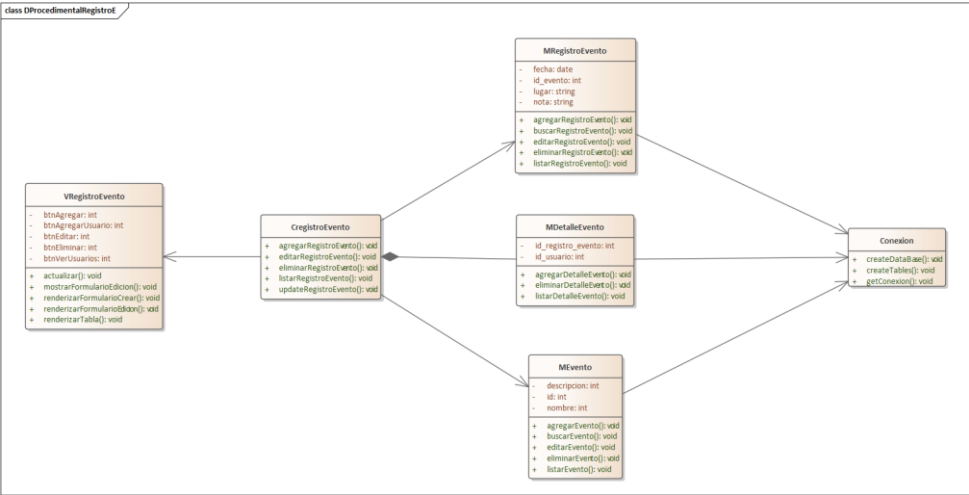
CU5: GESTIONAR EVENTO



CU6: GESTIONAR MINISTERIO

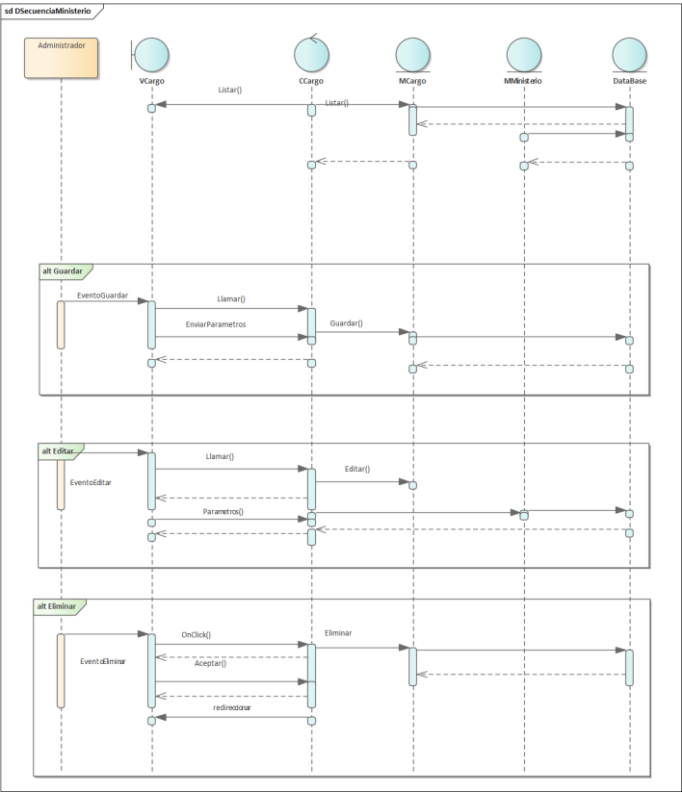


CU7: GESTIONAR REGISTRO EVENTO

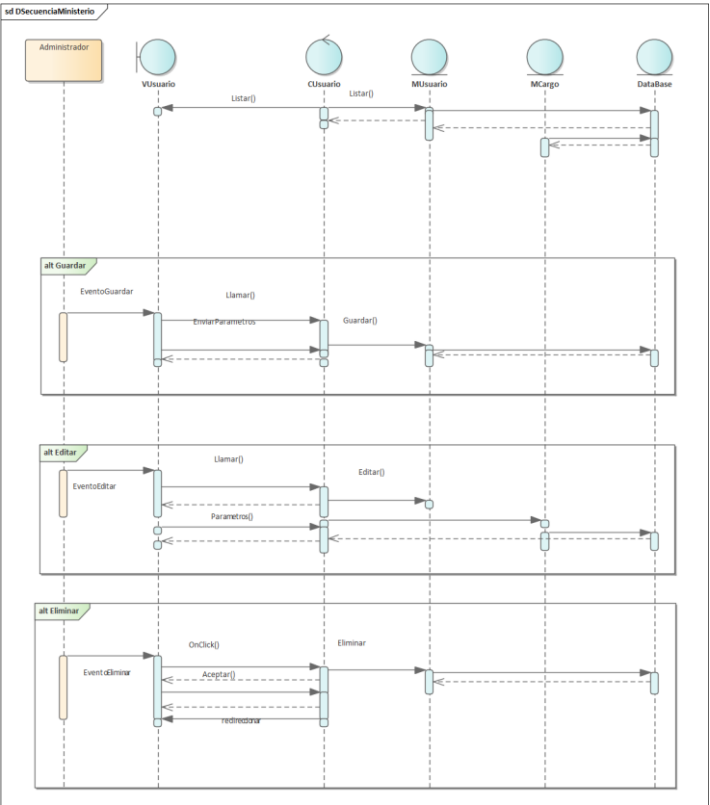


3.2.4 DIAGRAMAS DE SECUENCIA

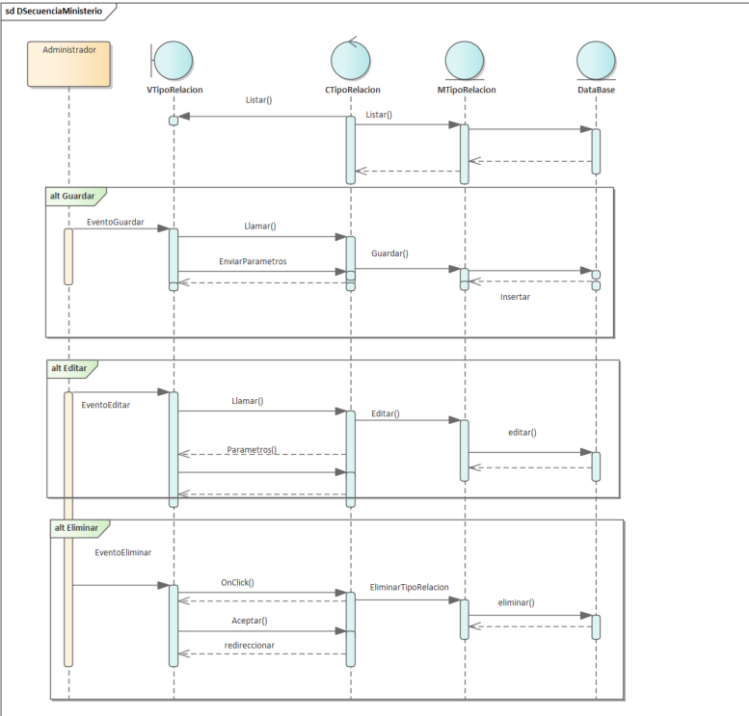
CU1: Gestionar Cargo



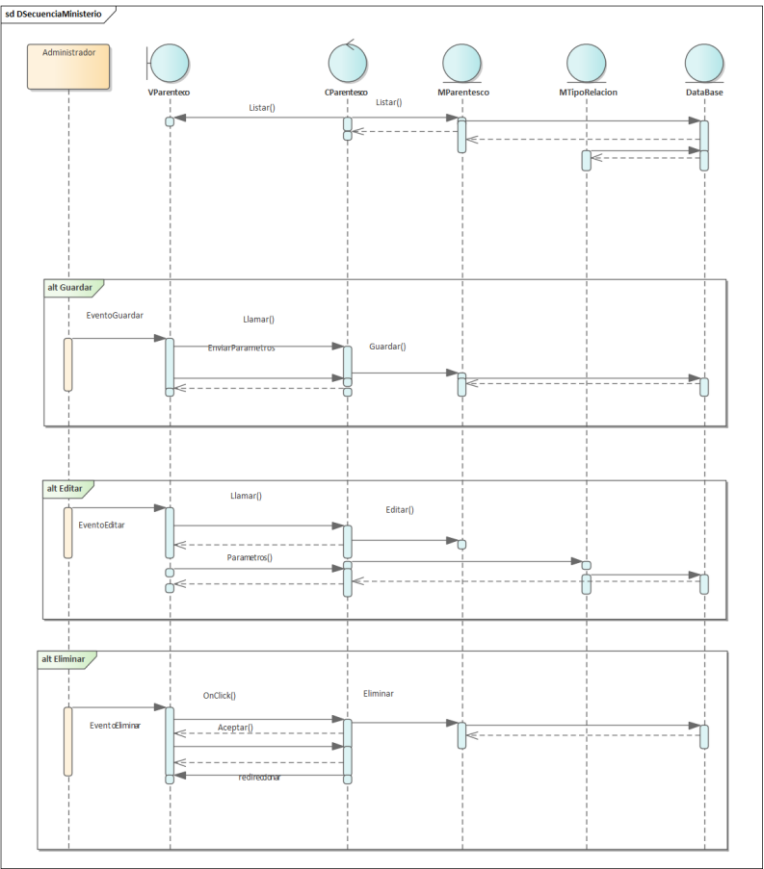
CU2: Gestionar Usuario



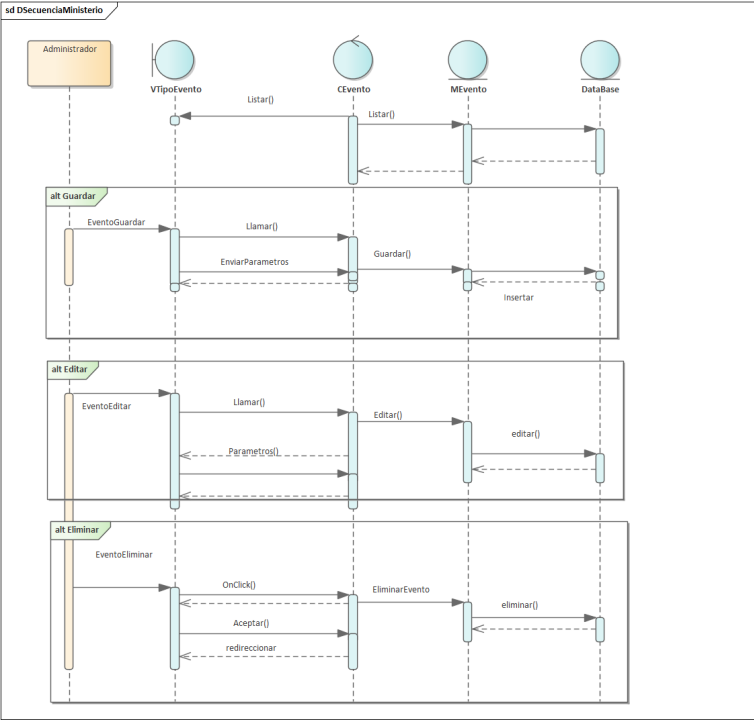
CU3: Gestionar Tipo Relación



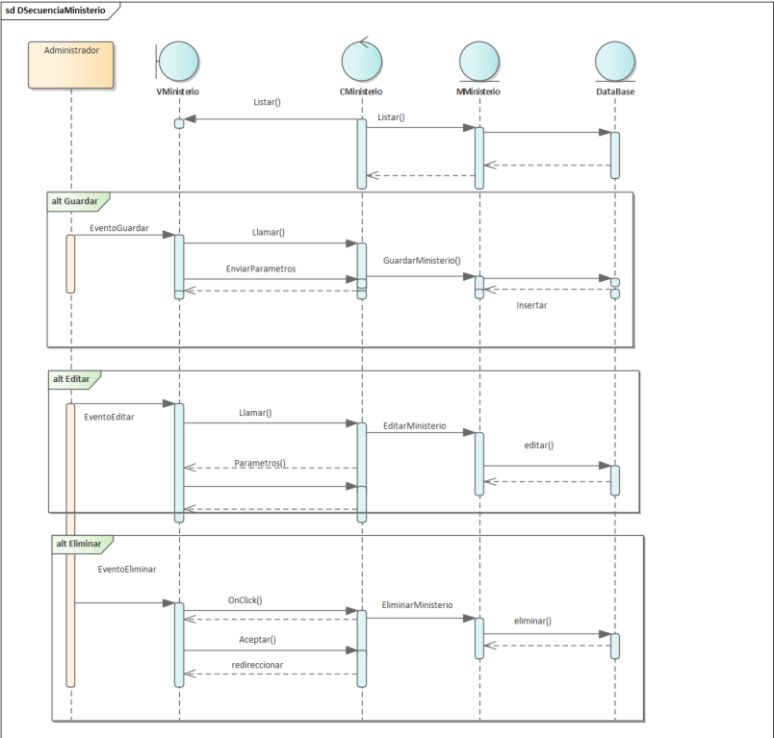
CU4: Gestionar Parentesco



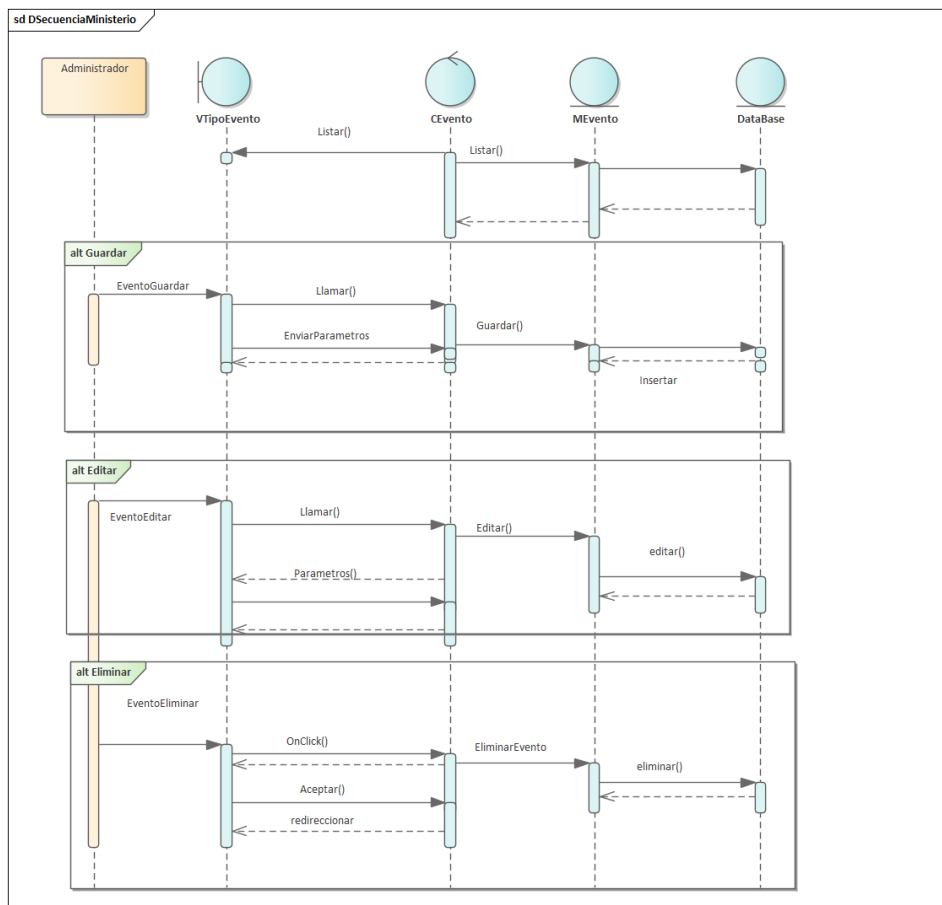
CU5: Gestionar Evento



CU6: Gestionar Ministerio



CU7: Gestionar Registro Evento



4. CODIFICACION

CU1: Gestionar Cargo

➤ Controlador

```
<?php
declare(strict_types=1);

require_once(' ../app/modelos/Cargo/MCargo.php');
require_once(' ../app/modelos/Ministerio/MMinisterio.php');
require_once(' ../app/vistas/Cargo/VCargo.php');

class CCargo
{
    private VCargo $vista;
    private MCargo $modelo;
    private MMinisterio $modeloMinisterio;

    public function __construct()
    {
```

```

➤     $this->vista = new VCargo();
➤     $this->modelo = new MCargo();
➤     $this->modeloMinisterio = new MMinisterio();
➤ }
➤
➤ public function listarCargosC(): void
➤ {
➤
➤     $cargos = $this->modelo->listarCargos();
➤     $ministerios = $this->modeloMinisterio-
➤ >listarMinisterios();
➤     $this->vista->actualizar($cargos,$ministerios);
➤ }
➤
➤ public function crearCargoC(string $nombre, string
➤ $descripcion, int $ministerio_id): void
➤ {
➤
➤     $this->modelo->crearCargo($nombre, $descripcion,
➤ $ministerio_id);
➤     $cargos = $this->modelo->listarCargos();
➤     $ministerios = $this->modeloMinisterio-
➤ >listarMinisterios();
➤     $this->vista->actualizar($cargos, $ministerios);
➤ }
➤
➤ public function eliminarCargoC(int $id): void
➤ {
➤     $this->modelo->eliminarCargo($id);
➤     $cargos = $this->modelo->listarCargos();
➤     $ministerios = $this->modeloMinisterio-
➤ >listarMinisterios();
➤     $this->vista->actualizar($cargos, $ministerios);
➤ }
➤
➤ public function updateCargoC(int $id): void
➤ {
➤     $cargos = $this->modelo->buscarCargo($id);
➤     $ministerios = $this->modeloMinisterio-
➤ >listarMinisterios();
➤     $this->vista-
➤ >mostrarFormularioEdicion($cargos,$ministerios);
➤ }
➤
➤ public function editarCargoC(int $id, string $nombre, string
➤ $descripcion, int $ministerio_id): void
➤ {
➤     $this->modelo->editarCargo($id, $nombre, $descripcion,
➤ $ministerio_id);

```

```

➤         $cargos = $this->modelo->listarCargos();
➤         $ministerios = $this->modeloMinisterio-
➤ >listarMinisterios();
➤         $this->vista->actualizar($cargos, $ministerios);
➤     }
➤ }
➤

```

➤ Modelo

```

➤ <?php
➤
➤ require_once(' ../app/modelos/iglesiaDB.php');
➤ require_once(' ../app/modelos/Cargo/Cargo.php');
➤
➤ class MCargo
➤ {
➤     private IglesiaDB $database;
➤
➤     public function __construct()
➤     {
➤         $this->database = new IglesiaDB();
➤     }
➤
➤     public function crearCargo(string $nombre, string
➤ $descripcion,$ministerio_id): void //crear
➤     {
➤         $bd = $this->database->getConnection();
➤         try {
➤             $query = "INSERT INTO " . $this->database::TABLE_CARGO
➤ . " (nombre, descripcion, ministerio_id) VALUES (?, ?, ?)";
➤             $stmt = $bd->prepare($query);
➤             $stmt->bind_param("ssi", $nombre,
➤ $descripcion,$ministerio_id);
➤             if ($stmt->execute()) {
➤                 error_log("Cargo insertado con éxito");
➤             }
➤         } catch (Exception $e) {
➤             error_log("Excepción al insertar el cargo en la base de
➤ datos: " . $e->getMessage());
➤         } finally {
➤             if (isset($stmt)) {
➤                 $stmt->close();
➤             }
➤             $bd->close();
➤         }
➤     }
➤
➤     public function listarCargos(): array //listar

```

```

➤ {
➤     $bd = $this->database->getConnection();
➤
➤     $cargos = [];
➤
➤     try {
➤         $result = $bd->query('SELECT * FROM ' . $this-
➤ >database::TABLE_CARGO);
➤
➤         if ($result) {
➤             while ($row = $result->fetch_assoc()) {
➤
➤                 $cargo = new Cargo($row['id'], $row['nombre'],
➤ $row['descripcion'],$row['ministerio_id']);
➤                 $cargos[] = $cargo;
➤             }
➤         }
➤     } catch (Exception $e) {
➤         error_log("Excepción en mostrarCargos: " . $e-
➤ >getMessage());
➤     } finally {
➤         $bd->close();
➤     }
➤
➤     return $cargos;
➤ }
➤
➤ public function buscarCargo(int $id): Cargo
➤ {
➤     $bd = $this->database->getConnection();
➤
➤     try {
➤         $query = "SELECT * FROM " . $this-
➤ >database::TABLE_CARGO . " WHERE id = ?";
➤         $stmt = $bd->prepare($query);
➤         $stmt->bind_param("i", $id);
➤         $stmt->execute();
➤         $result = $stmt->get_result();
➤
➤         if ($result && $result->num_rows > 0) {
➤             $row = $result->fetch_assoc();
➤             // Crear y devolver un objeto Cargo a partir de los
➤ datos de la fila
➤             $cargo = new Cargo($row['id'], $row['nombre'],
➤ $row['descripcion'],$row['ministerio_id']);
➤             return $cargo;
➤         }
➤     } catch (Exception $e) {

```

```

➤         error_log("Excepción en buscarCargo: " . $e-
➤ >getMessage());
➤     } finally {
➤         $bd->close();
➤     }
➤
➤     return null;
➤ }
➤
➤ public function editarCargo(int $id, string $nombre, string
➤ $descripcion, $ministerio_id): void
➤ {
➤     $bd = $this->database->getConnection();
➤
➤     try {
➤         $query = "UPDATE " . $this->database::TABLE_CARGO . "
➤ SET nombre = ?, descripcion = ? WHERE id = ?";
➤         $stmt = $bd->prepare($query);
➤         $stmt->bind_param("ssii", $nombre,
➤ $descripcion,$ministerio_id , $id);
➤
➤         if ($stmt->execute()) {
➤             error_log("Cargo editado con éxito");
➤         }
➤     } catch (Exception $e) {
➤         error_log("Excepción al editar el cargo: " . $e-
➤ >getMessage());
➤     } finally {
➤         if (isset($stmt)) {
➤             $stmt->close();
➤         }
➤         $bd->close();
➤     }
➤ }
➤
➤ public function eliminarCargo(int $id): void
➤ {
➤     $bd = $this->database->getConnection();
➤
➤     try {
➤         $query = "DELETE FROM " . $this->database::TABLE_CARGO
➤ . " WHERE id = ?";
➤         $stmt = $bd->prepare($query);
➤         $stmt->bind_param("i", $id);
➤
➤         if ($stmt->execute()) {
➤             error_log("Cargo eliminado con éxito");
➤         }
➤     } catch (Exception $e) {

```

```

➤         error_log("Excepción al eliminar el cargo: " . $e-
➤ >getMessage());
➤     } finally {
➤         if (isset($stmt)) {
➤             $stmt->close();
➤         }
➤         $bd->close();
➤     }
➤ }
➤ }
➤

```

```

<?php

class Cargo
{
    private int $id;
    private string $nombre;
    private string $descripcion;
    private $ministerio_id;

    public function __construct(int $id, string $nombre, string
$descripcion,$ministerio_id)
    {
        $this->id = $id;
        $this->nombre = $nombre;
        $this->descripcion = $descripcion;
        $this->ministerio_id= $ministerio_id;
    }

    public function getId(): int
    {
        return $this->id;
    }

    public function getNombre(): string
    {
        return $this->nombre;
    }

    public function getDescripcion(): string
    {
        return $this->descripcion;
    }

    public function setId(int $id): void
    {
        $this->id = $id;
    }
}

```

```

    }

    public function setNombre(string $nombre): void
    {
        $this->nombre = $nombre;
    }

    public function setDescription(string $descripcion): void
    {
        $this->descripcion = $descripcion;
    }

    public function getMinisterioId()
    {
        return $this->ministerio_id;
    }

    public function setMinisterioId($ministerio_id)
    {
        $this->ministerio_id = $ministerio_id;
    }

    public function __toString(): string
    {
        return "Cargo{" .
            "id=" . $this->id .
            ", nombre='" . $this->nombre . '\'' .
            ", descripcion='" . $this->descripcion . '\'' .
            ", ministerio_id=" . $this->ministerio_id .
            '\'';
    }
}

```

➤ Vista

```

➤ <?php
➤
➤ declare(strict_types=1);
➤
➤ class VCargo
➤ {
➤     private function renderizarTabla($cargos,array $ministerios):
string
➤     {
➤         $rowData = '';
➤
➤         if (empty($cargos)) {
➤             return "<tbody></tbody>";
➤         }
➤     }

```



```

➤         foreach ((array)$cargos as $cargo) {
➤             $rowData .= "<tr>";
➤             $rowData .= "<td>{$cargo->getId()}</td>";
➤             $rowData .= "<td>{$cargo->getNombre()}</td>";
➤             $rowData .= "<td>{$cargo->getDescripcion()}</td>";
➤
➤             // Busca el nombre del cargo basado en el ID del cargo
➤             $nombreMinisterio = "";
➤             foreach ($ministerios as $ministerio) {
➤                 if ($ministerio->getId() == $cargo-
➤ >getMinisterioId()) {
➤                     $rowData .= "<td>{$ministerio-
➤ >getNombre()}</td>";
➤                     break;
➤                 }
➤             }
➤
➤             $rowData .= "<td><a href='/editar_cargo?id={$cargo-
➤ >getId()}' class='edit-button'>Editar</a></td>";
➤             $rowData .= "<td>";
➤             $rowData .= "<form id='form_boton_eliminar'
➤ method='post' action='/eliminar_cargo'>";
➤             $rowData .= "<input type='hidden' name='id'
➤ value='{ $cargo->getId()}'>";
➤             $rowData .= "<button type='button'
➤ onclick='confirmarEliminar({ $cargo->getId()})' class='delete-
➤ button' style='font-size: 16px;'>Eliminar</button>";
➤
➤             $rowData .= "</form>";
➤             $rowData .= "</td>";
➤             $rowData .= "</tr>";
➤         }
➤
➤         return "<tbody>$rowData</tbody>";
➤     }
➤
➤     public function renderizarFormularioCrear($ministerios): string
➤     {
➤
➤         $nuevoCargoForm = '
➤         <div class="container">
➤             <h2>Nuevo Cargo </h2>
➤         </div>
➤         <div class="container">
➤
➤             <div id="form_cargo">

```

```

➤         <form action="/cargos" method="post">
➤             <div class="row">
➤                 <div class="col-12 col-md-4">
➤                     <label for="nombre">Nombre</label>
➤                     <input name="nombre" type="text"
➤ id="nombre" required>
➤                 </div>
➤                 <div class="col-12 col-md-4">
➤                     <label
➤ for="descripcion">Descripcion</label>
➤                     <input name="descripcion" type="text"
➤ id="descripcion" required>
➤                 </div>
➤
➤                 <div class="col-12 col-md-4">
➤                     <label for="ministerio">Ministerio</label>
➤                     <select name="ministerio" id="ministerio"
➤ >';
➤
➤                     foreach ($ministerios as $ministerio) {
➤                         $nuevoCargoForm .= "<option
➤ value='{ $ministerio->getId()}'>{ $ministerio-
➤ >getNombre()}</option>";
➤                     }
➤
➤                     $nuevoCargoForm .= '
➤                         </select>
➤                     </div>
➤                 </div>
➤                 <br>
➤                 <div class="row mt-3">
➤                     <div class="col-12 d-flex justify-content-end">
➤                         <button class="add-button" type="submit">Ag
➤regar</button>
➤                     </div>
➤                 </div>
➤
➤                 </div>
➤             </form>
➤         </div>
➤     </div>
➤     '
➤
➤     return $nuevoCargoForm;
➤ }
➤
➤ public function actualizar($cargos, $ministerios): void
➤ {

```

```

➤         $title = 'Cargos';
➤         $tbody = $this->renderizarTabla($cargos, $ministerios);
➤         $formulario = $this-
➤ >renderizarFormularioCrear($ministerios);
➤         include '../app/vistas/Cargo/index.php';
➤     }
➤
➤     private function renderizarFormularioEdicion($cargo,
➤ $ministerios): string
➤     {
➤         $formulario = '';
➤
➤         if (!empty($cargo)) {
➤             $formulario .= "<form method='post'
➤ action='/actualizar_cargo'>";
➤             $formulario .= "<input type='hidden' name='id'
➤ value='{ $cargo->getId()}'>";
➤             $formulario .= "<div class='container-fluid'>";
➤             $formulario .= "<div class='row'>";
➤             $formulario .= "<div class='col-12 col-md-4'>";
➤             $formulario .= "<label for='nombre'>Nombres:</label>";
➤             $formulario .= "<input type='text' id='nombre'
➤ name='nombre' value='{ $cargo->getNombre()}' required>";
➤             $formulario .= "</div>";
➤             $formulario .= "<div class='col-12 col-md-4'>";
➤             $formulario .= "<label
➤ for='apellido'>Apellidos:</label>";
➤             $formulario .= "<input type='text' id='apellido'
➤ name='apellido' value='{ $cargo->getDescripcion()}' required>";
➤             $formulario .= "</div>";
➤             $formulario .= "</div>";
➤             $formulario .= "<div class='col-12 col-md-4'>";
➤             $formulario .= "<label
➤ for='ministerio'>Ministerio:</label>";
➤             $formulario .= "<select name='ministerio'
➤ id='ministerio' required>";
➤
➤             foreach ($ministerios as $ministerio) {
➤                 $selected = $ministerio->getId() === $cargo-
➤ >getMinisterioId() ? 'selected' : '';
➤                 $formulario .= "<option value='{ $ministerio-
➤ >getId()}' {$selected}>{ $ministerio->getNombre()}</option>";
➤             }
➤
➤             $formulario .= "</select>";
➤             $formulario .= "</div>";
➤             $formulario .= "</div>";
➤             $formulario .= "<br>";
➤             $formulario .= "<div class='row mt-3'>";

```

```

➤         $formulario .= "<div class='col-12 d-flex justify-
content-end'>";
➤         $formulario .= "<button class='add-button'
type='submit'>Actualizar</button>";
➤         $formulario .= "<a href='/menu' class='back-
button'>Volver</a>";
➤         $formulario .= "</div>";
➤         $formulario .= "</div>";
➤         $formulario .= "</form>";
➤     } else {
➤         $formulario .= "El cargo no se encontró o no existe.";
➤     }
➤
➤     return $formulario;
➤ }
➤
➤ public function mostrarFormularioEdicion($cargo, $ministerios):
void
➤ {
➤     $title = 'Cargo - Editar';
➤     $formulario = $this->renderizarFormularioEdicion($cargo,
$ministerios);
➤     include '../app/vistas/Cargo/edit.php';
➤ }
➤ }
➤
➤ <!DOCTYPE html>
➤ <html>
➤
➤ <head>
➤     <meta charset="UTF-8">
➤     <meta name="viewport" content="width=device-width, initial-
scale=1">
➤     <title>Iglesia </title>
➤     <link rel="stylesheet"
href="https://fonts.googleapis.com/css?family=Lora&display=swap">
➤     <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootst
rap.min.css">
➤     <link rel="stylesheet" href="css/estilo.css">
➤ </head>
➤ <nav class="sidebar">
➤     <a href="/menu">Inicio</a>
➤     <a href="/usuarios">Usuario</a>
➤     <a href="/cargos">Cargo</a>
➤     <a href="/ministerios">Ministerio</a>
➤     <a href="/eventos">Evento</a>
➤     <a href="/registro_evento">Registro Eventos</a>
➤     <a href="/tipo_relacion">Tipo Relacion</a>

```

```

➤      <a href="/parentescos">Parentesco</a>
➤    </nav>
➤  <body>
➤    <header>
➤      <div class="container">
➤        <h1>Iglesia </h1>
➤      </div>
➤    </header>
➤    <nav>
➤      <div class="container">
➤
➤      </div>
➤    </nav>
➤    <main>
➤      <?= $formulario ?>
➤      <h1><?= $title ?></h1>
➤      <table>
➤        <thead>
➤          <tr>
➤            <th>Id</th>
➤            <th>Nombre</th>
➤            <th>Descripcion</th>
➤            <th>Ministerio</th>
➤          </tr>
➤        </thead>
➤        <?= $tbody ?>
�~      </table>
➤    </div>
➤  </main>
➤  <footer>
➤    <div class="container">
➤      <p>&copy; <?php echo date("Y"); ?> VeronicaAntezana</p>
➤    </div>
➤  </footer>
➤  <script>
➤    function confirmarEliminar(id) {
➤      if (confirm('¿Estás seguro de que deseas eliminar este
cargo?')) {
➤        // Si el usuario confirma, enviar el formulario de
eliminación
➤        document.querySelector('form[action="/eliminar_carg
o"] input[name="id"]').value = id;
➤        document.querySelector('form[action="/eliminar_carg
o"]').submit();
➤      }
➤    }
➤  </script>
➤ </body>
➤

```

```

➤ </html>
➤ <!DOCTYPE html>
➤ <html>
➤
➤ <head>
➤     <meta charset="UTF-8">
➤     <meta name="viewport" content="width=device-width, initial-
➤ scale=1">
➤     <title>Iglesia App</title>
➤     <link rel="stylesheet"
➤ href="https://fonts.googleapis.com/css?family=Lora&display=swap">
➤     <link rel="stylesheet"
➤ href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootst
➤ rap.min.css">
➤     <link rel="stylesheet" href="css/estilo.css">
➤ </head>
➤
➤ <body>
➤     <header>
➤         <div class="container">
➤             <h1>Iglesia App</h1>
➤         </div>
➤     </header>
➤
➤     <main>
➤         <div class="container">
➤             <h2><?= $title ?></h2>
➤
➤             <!-- Muestra el formulario de edición aquí -->
➤             <?= $formulario ?>
➤
➤             <!-- Más contenido HTML si es necesario -->
➤         </div>
➤     </main>
➤     <footer>
➤         <div class="container">
➤             <p>&copy; <?php echo date("Y"); ?> Veronica
➤ Antezana</p>
➤         </div>
➤     </footer>
➤     <script>
➤         function confirmarEliminar(id) {
➤             if (confirm('¿Estás seguro de que deseas eliminar este
➤ Tipo de Relacion?')) {
➤                 // Si el usuario confirma, enviar el formulario de
➤ eliminación
➤                 document.querySelector('form[action="/eliminar_carg
➤ o"] input[name="id"]').value = id;

```

```

➤         document.querySelector('form[action="/eliminar_carg
➤ o"]').submit();
➤     }
➤ }
➤ </script>
➤ </body>
➤
➤ </html>

```

CU2: Gestionar Usuario

➤ Controlador

```

➤ <?php
➤
➤ declare(strict_types=1);
➤
➤ require_once(' ../app/modelos/Usuario/MUsuario.php');
➤ require_once(' ../app/modelos/Cargo/MCargo.php');
➤ require_once(' ../app/vistas/Usuario/VUsuario.php');
➤
➤ class CUsuario
➤ {
➤     private VUsuario $vista;
➤     private MUsuario $modeloUsuario;
➤     private MCargo $modeloCargo;
➤
➤     public function __construct()
➤     {
➤         $this->vista = new VUsuario();
➤         $this->modeloUsuario = new MUsuario();
➤         $this->modeloCargo = new MCargo();
➤     }
➤
➤     public function listarUsuariosC(): void
➤     {
➤         $usuarios = $this->modeloUsuario->listarUsuarios();
➤         $cargos = $this->modeloCargo->listarCargos();
➤         /* var_dump($cargos); */
➤         $this->vista->actualizar($usuarios, $cargos);
➤     }
➤
➤     public function agregarUsuarioC(string $nombre, string
➤ $apellido, string $fechanacimiento, string $ci, int
➤ $telefono,int $cargo_id): void
➤     {
➤         $this->modeloUsuario->agregarUsuario($nombre, $apellido,
➤ $fechanacimiento, $ci, $telefono, $cargo_id);
➤         $usuarios = $this->modeloUsuario->listarUsuarios();

```

```

➤         $cargos = $this->modeloCargo->listarCargos();
➤         $this->vista->actualizar($usuarios, $cargos);
➤     }
➤
➤     public function eliminarUsuarioC(int $id): void
➤     {
➤         $this->modeloUsuario->eliminarUsuario($id);
➤         $usuarios = $this->modeloUsuario->listarUsuarios();
➤         $cargos = $this->modeloCargo->listarCargos();
➤         $this->vista->actualizar($usuarios, $cargos);
➤     }
➤
➤     public function editarUsuarioC(int $id, string $nombre,
➤ string $apellido, string $fechanacimiento, string $ci, int
➤ $telefono, int $cargo_id): void
➤     {
➤         $this->modeloUsuario->editarUsuario($id, $nombre,
➤ $apellido, $fechanacimiento, $ci, $telefono,$cargo_id);
➤         $usuarios = $this->modeloUsuario->listarUsuarios();
➤         $cargos = $this->modeloCargo->listarCargos();
➤         $this->vista->actualizar($usuarios, $cargos);
➤     }
➤
➤     public function updateUsuarioC(int $id): void
➤     {
➤         $usuario = $this->modeloUsuario->buscarUsuario($id);
➤         $cargos = $this->modeloCargo->listarCargos();
➤         $this->vista->mostrarFormularioEdicion($usuario,
➤ $cargos);
➤     }
➤ }

```

➤ Modelo

```

<?php

require_once(' ../app/modelos/IglesiaDB.php');
require_once(' ../app/modelos/Usuario/Usuario.php');

class MUsuario
{
    private IglesiaDB $database;

    public function __construct()
    {
        $this->database = new IglesiaDB();
    }
}

```



```

    public function agregarUsuario($nombre, $apellido, $fechanacimiento,
$ci, $telefono,$cargo_id): void
    {
        $bd = $this->database->getConnection();
        try {
            $query = "INSERT INTO " . $this->database::TABLE_USUARIO . "
(nombre, apellido, fechanacimiento, ci, telefono, cargo_id) VALUES (?, ?,
?, ?, ?, ?)";
            $stmt = $bd->prepare($query);
            $stmt->bind_param("ssssi", $nombre, $apellido,
$fechanacimiento, $ci, $telefono, $cargo_id);
            if ($stmt->execute()) {
                error_log("Usuario insertado con éxito");
            }
        } catch (Exception $e) {
            error_log("Excepción al insertar el usuario en la base de
datos: " . $e->getMessage());
        } finally {
            if (isset($stmt)) {
                $stmt->close();
            }
            $bd->close();
        }
    }

    public function listarUsuarios(): array
    {
        $bd = $this->database->getConnection();

        $usuarios = [];

        try {
            $result = $bd->query('SELECT * FROM ' . $this-
>database::TABLE_USUARIO);

            if ($result) {
                while ($row = $result->fetch_assoc()) {
                    $usuario = new Usuario($row['id'], $row['nombre'],
$row['apellido'], $row['fechanacimiento'], $row['ci'], $row['telefono'],
$row['cargo_id']);
                    $usuarios[] = $usuario;
                }
            }
        } catch (Exception $e) {
            error_log("Excepción en mostrarUsuarios: " . $e-
>getMessage());
        } finally {
            $bd->close();
        }
    }

```

```

    }

    return $usuarios;
}

public function buscarUsuario($id)
{
    $bd = $this->database->getConnection();

    try {
        $query = "SELECT * FROM " . $this->database::TABLE_USUARIO .
" WHERE id = ?";
        $stmt = $bd->prepare($query);
        $stmt->bind_param("i", $id);
        $stmt->execute();
        $result = $stmt->get_result();

        if ($result && $result->num_rows > 0) {
            $row = $result->fetch_assoc();
            $usuario = new Usuario($row['id'], $row['nombre'],
$row['apellido'], $row['fechanacimiento'], $row['ci'], $row['telefono'],
$row['cargo_id']);
            return $usuario;
        }
    } catch (Exception $e) {
        error_log("Excepción en buscarUsuario: " . $e->getMessage());
    } finally {
        $bd->close();
    }

    return null;
}

public function editarUsuario($id, $nombre, $apellido,
$fechanacimiento, $ci, $telefono, $cargo_id): void
{
    $bd = $this->database->getConnection();

    try {
        $query = "UPDATE " . $this->database::TABLE_USUARIO . " SET
nombre = ?, apellido = ?, fechanacimiento = ?, ci = ?, telefono = ?,
cargo_id = ? WHERE id = ?";
        $stmt = $bd->prepare($query);
        $stmt->bind_param("ssssiii", $nombre, $apellido,
$fechanacimiento, $ci, $telefono, $cargo_id, $id);

        if ($stmt->execute()) {
            error_log("Usuario editado con éxito");
        }
    }

```

```

        } catch (Exception $e) {
            error_log("Excepción al editar el usuario: " . $e-
>getMessage());
        } finally {
            if (isset($stmt)) {
                $stmt->close();
            }
            $bd->close();
        }
    }

    public function eliminarUsuario($id): void
    {
        $bd = $this->database->getConnection();

        try {
            $query = "DELETE FROM " . $this->database::TABLE_USUARIO . "
WHERE id = ?";
            $stmt = $bd->prepare($query);
            $stmt->bind_param("i", $id);

            if ($stmt->execute()) {
                error_log("Usuario eliminado con éxito");
            }
        } catch (Exception $e) {
            error_log("Excepción al eliminar el usuario: " . $e-
>getMessage());
        } finally {
            if (isset($stmt)) {
                $stmt->close();
            }
            $bd->close();
        }
    }
}

```

```

<?php

class Usuario
{
    private $id;
    private $nombre;
    private $apellido;
    private $fechanacimiento;
    private $ci;
    private $telefono;
    private $cargo_id;
}

```

```
    public function __construct($id, $nombre, $apellido,
$fechanacimiento, $ci,$telefono, $cargo_id)
    {
        $this->id = $id;
        $this->nombre = $nombre;
        $this->apellido = $apellido;
        $this->fechanacimiento = $fechanacimiento;
        $this->ci = $ci;
        $this->telefono = $telefono;
        $this->cargo_id = $cargo_id;
    }

    public function getId()
    {
        return $this->id;
    }

    public function getNombre()
    {
        return $this->nombre;
    }

    public function setNombre($nombre)
    {
        $this->nombre = $nombre;
    }

    public function getApellido()
    {
        return $this->apellido;
    }

    public function setApellido($apellido)
    {
        $this->apellido = $apellido;
    }

    public function getFechaNacimiento()
    {
        return $this->fechanacimiento;
    }

    public function setFechaNacimiento($fechanacimiento)
    {
        $this->fechanacimiento = $fechanacimiento;
    }

    public function getCI()
```

```

{
    return $this->ci;
}

public function setCI($ci)
{
    $this->ci = $ci;
}

public function getTelefono()
{
    return $this->telefono;
}

public function setTelefono($telefono)
{
    $this->telefono = $telefono;
}

public function getCargoId()
{
    return $this->cargo_id;
}

public function setCargoId($cargo_id)
{
    $this->cargo_id = $cargo_id;
}

public function __toString()
{
    return "Usuario{" .
        "id=" . $this->id .
        ", nombre='" . $this->nombre . '\'' .
        ", apellido='" . $this->apellido . '\'' .
        ", fechanacimiento='" . $this->fechanacimiento . '\'' .
        ", ci='" . $this->ci . '\'' .
        ", telefono='" . $this->telefono . '\'' .
        ", cargo_id=" . $this->cargo_id .
        '}';
}
}

```

➤ Vista

➤ <?php

➤

➤ declare(strict_types=1);

➤

```

➤ class VUsuario
➤ {
➤     private function renderizarTabla($usuarios,array $cargos):
string
➤     {
➤         $rowData = '';
➤
➤         if (empty($usuarios)) {
➤             return "<tbody></tbody>";
➤         }
➤
➤         foreach ((array)$usuarios as $usuario) {
➤             $rowData .= "<tr>";
➤             $rowData .= "<td>{$usuario->getId()}</td>";
➤             $rowData .= "<td>{$usuario->getNombre()}</td>";
➤             $rowData .= "<td>{$usuario->getApellido()}</td>";
➤             $rowData .= "<td>{$usuario-
➤>getFechaNacimiento()}</td>";
➤             $rowData .= "<td>{$usuario->getCI()}</td>";
➤             $rowData .= "<td>{$usuario->getTelefono()}</td>";
➤
➤             // Busca el nombre del cargo basado en el ID del cargo
➤             $nombreCargo = "";
➤             foreach ($cargos as $cargo) {
➤                 if ($cargo->getId() == $usuario->getCargoId()) {
➤                     $rowData .= "<td>{$cargo->getNombre()}</td>";
➤                     break;
➤                 }
➤             }
➤
➤             $rowData .= "<td><a href='/editar_usuario?id={$usuario-
➤>getId()}' class='edit-button'>Editar</a></td>";
➤             $rowData .= "<td>";
➤             $rowData .= "<form id='form_boton_eliminar'
method='post' action='/eliminar_usuario'>";
➤             $rowData .= "<input type='hidden' name='id'
value='{$usuario->getId()}'>";
➤             $rowData .= "<button type='button'
onclick='confirmarEliminar({$usuario->getId()})' class='delete-
button' style='font-size: 16px;'>Eliminar</button>";
➤
➤             $rowData .= "</form>";
➤             $rowData .= "</td>";
➤             $rowData .= "</tr>";
➤         }
➤
➤         return "<tbody>{$rowData}</tbody>";

```

```

    }

    public function renderizarFormularioCrear($cargos): string
    {

        $nuevoUsuarioForm = '
        <div class="container">
            <h2>Nuevo Usuario </h2>
        </div>
        <div class="container">

            <div id="form_usuario">
                <form action="/usuarios" method="post">
                    <div class="row">
                        <div class="col-12 col-md-4">
                            <label for="nombre">Nombre</label>
                            <input name="nombre" type="text"
id="nombre" required>
                        </div>
                        <div class="col-12 col-md-4">
                            <label for="apellido">Apellido</label>
                            <input name="apellido" type="text"
id="apellido" required>
                        </div>
                        <div class="col-12 col-md-4">
                            <label for="fechanacimiento">Fecha
Nacimiento</label>
                            <input name="fechanacimiento" type="date"
id="fechanacimiento" required>
                        </div>
                        <div class="col-12 col-md-4">
                            <label for="ci">CI</label>
                            <input name="ci" type="text" id="ci"
required>
                        </div>

                        <div class="col-12 col-md-4">
                            <label for="telefono">Telefono</label>
                            <input name="telefono" type="text"
id="telefono" required>
                        </div>
                        <div class="col-12 col-md-4">
                            <label for="cargo">Cargo</label>
                            <select name="cargo"
id="cargo" required>';

                            foreach ($cargos as $cargo) {
                                $nuevoUsuarioForm .= "<option
value='{${$cargo->getId()}}'>{${$cargo->getNombre()}}</option>";

```

```

    }

    $nuevoUsuarioForm .= '
        </select>
    </div>
</div>
<br>
<div class="row mt-3">
    <div class="col-12 d-flex justify-content-end">
        <button class="add-button"
type="submit">Agregar</button>
    </div>
</div>

    </div>
</form>
</div>
</div>
';

    return $nuevoUsuarioForm;
}

public function actualizar($usuarios, $cargos): void
{
    $title = 'Usuarios';
    $tbody = $this->renderizarTabla($usuarios, $cargos);
    $formulario = $this->renderizarFormularioCrear($cargos);
    include '../app/vistas/Usuario/index.php';
}

private function renderizarFormularioEdicion($usuario,
$cargos): string
{
    $formulario = '';

    if (!empty($usuario)) {
        $formulario .= "<form method='post'
action='/actualizar_usuario'>";
        $formulario .= "<input type='hidden' name='id'
value='{ $usuario->getId()}'>";
        $formulario .= "<div class='container-fluid'>";
        $formulario .= "<div class='row'>";
        $formulario .= "<div class='col-12 col-md-4'>";
        $formulario .= "<label for='nombre'>Nombres:</label>";
        $formulario .= "<input type='text' id='nombre'
name='nombre' value='{ $usuario->getNombre()}' required>";
        $formulario .= "</div>";
    }
}

```



```

➤ $formulario .= "<div class='col-12 col-md-4'>";
➤ $formulario .= "<label
for='apellido'>Apellidos:</label>";
➤ $formulario .= "<input type='text' id='apellido'
name='apellido' value='{ $usuario->getApellido()}' required>";
➤ $formulario .= "</div>";
➤ $formulario .= "</div>";
➤ $formulario .= "<div class='row'>";
➤ $formulario .= "<div class='col-12 col-md-4'>";
➤ $formulario .= "<label for='fechanacimiento'>Fecha
Nacimiento:</label>";
➤ $formulario .= "<input type='date' id='fechanacimiento'
name='fechanacimiento' value='{ $usuario->getFechaNacimiento()}'
required>";
➤ $formulario .= "</div>";
➤ $formulario .= "<div class='col-12 col-md-4'>";
➤ $formulario .= "<label for='ci'>CI:</label>";
➤ $formulario .= "<input type='text' id='ci' name='ci'
value='{ $usuario->getCI()}' required>";
➤ $formulario .= "</div>";
➤ $formulario .= "<div class='col-12 col-md-4'>";
➤ $formulario .= "<label
for='telefono'>Telefono:</label>";
➤ $formulario .= "<input type='number' id='telefono'
name='telefono' value='{ $usuario->getTelefono()}' required>";
➤ $formulario .= "</div>";
➤ $formulario .= "<div class='col-12 col-md-4'>";
➤ $formulario .= "<label for='cargo'>Cargo:</label>";
➤ $formulario .= "<select name='cargo' id='cargo'
required>";
➤
➤     foreach ($cargos as $cargo) {
➤         $selected = $cargo->getId() === $usuario-
➤ >getCargoId() ? 'selected' : '';
➤         $formulario .= "<option value='{ $cargo->getId()}'
{ $selected}>{ $cargo->getNombre()}</option>";
➤     }
➤
➤     $formulario .= "</select>";
➤     $formulario .= "</div>";
➤     $formulario .= "</div>";
➤     $formulario .= "<br>";
➤     $formulario .= "<div class='row mt-3'>";
➤     $formulario .= "<div class='col-12 d-flex justify-
content-end'>";
➤     $formulario .= "<button class='add-button'
type='submit'>Actualizar</button>";
➤     $formulario .= "<a href='/usuarios' class='back-
button'>Volver</a>";

```

```

➤         $formulario .= "</div>";
➤         $formulario .= "</div>";
➤         $formulario .= "</form>";
➤     } else {
➤         $formulario .= "El usuario no se encontró o no
existe.";
➤     }
➤
➤     return $formulario;
➤ }
➤
➤ public function mostrarFormularioEdicion($usuario, $cargos):
void
➤ {
➤     $title = 'Usuario - Editar';
➤     $formulario = $this->renderizarFormularioEdicion($usuario,
$cargos);
➤     include '../app/vistas/Usuario/edit.php';
➤ }
➤ }
➤
➤ <!DOCTYPE html>
➤ <html>
➤
➤ <head>
➤     <meta charset="UTF-8">
➤     <meta name="viewport" content="width=device-width, initial-
scale=1">
➤     <title>Iglesia App</title>
➤     <link rel="stylesheet"
➤ href="https://fonts.googleapis.com/css?family=Lora&display=swap">
➤     <link rel="stylesheet"
➤ href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootst
rap.min.css">
➤     <link rel="stylesheet" href="css/estilo.css">
➤ </head>
➤ <nav class="sidebar">
➤     <a href="/menu">Inicio</a>
➤     <a href="/usuarios">Usuario</a>
➤     <a href="/cargos">Cargo</a>
➤     <a href="/ministerios">Ministerio</a>
➤     <a href="/eventos">Evento</a>
➤     <a href="/eventos">Registro Eventos</a>
➤     <a href="/tipo_relacion">Tipo Relacion</a>
➤     <a href="/parentescos">Parentesco</a>
➤ </nav>
➤ <body>
➤     <header>
➤         <div class="container">

```

```

➤         <h1>Iglesia </h1>
➤     </div>
➤ </header>
➤ <nav>
➤     <div class="container">
➤
➤     </div>
➤ </nav>
➤ <main>
➤     <?= $formulario ?>
➤     <h1><?= $title ?></h1>
➤     <table>
➤         <thead>
➤             <tr>
➤                 <th>Id</th>
➤                 <th>Nombre</th>
➤                 <th>Apellido</th>
➤                 <th>Fecha Nacimiento</th>
➤                 <th>CI</th>
➤                 <th>Telefono</th>
➤                 <th>Cargo</th>
➤             </tr>
➤         </thead>
➤         <?= $tbody ?>
➤     </table>
➤ </div>
➤ </main>
➤ <footer>
➤     <div class="container">
➤         <p>&copy; <?php echo date("Y"); ?> VeronicaAntezana</p>
➤     </div>
➤ </footer>
➤ <script>
➤     function confirmarEliminar(id) {
➤         if (confirm('¿Estás seguro de que deseas eliminar este
➤ usuario?')) {
➤             // Si el usuario confirma, enviar el formulario de
➤ eliminación
➤             document.querySelector('form[action="/eliminar_usua
➤ rio"] input[name="id"]').value = id;
➤             document.querySelector('form[action="/eliminar_usua
➤ rio"]').submit();
➤         }
➤     }
➤ </script>
➤ </body>
➤ </html>

```

```

<!DOCTYPE html>
<html>

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Iglesia App</title>
    <link rel="stylesheet"
href="https://fonts.googleapis.com/css?family=Lora&display=swap">
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.mi
n.css">
    <link rel="stylesheet" href="css/estilo.css">
</head>

<body>
    <header>
        <div class="container">
            <h1>Iglesia App</h1>
        </div>
    </header>

    <main>
        <div class="container">
            <h2><?= $title ?></h2>

            <!-- Muestra el formulario de edición aquí -->
            <?= $formulario ?>

            <!-- Más contenido HTML si es necesario -->
        </div>
    </main>
    <footer>
        <div class="container">
            <p>&copy; <?php echo date("Y"); ?> Veronica Antezana</p>
        </div>
    </footer>
    <script>
        function confirmarEliminar(id) {
            if (confirm('¿Estás seguro de que deseas eliminar este Tipo
de Relacion?')) {
                // Si el usuario confirma, enviar el formulario de
eliminación
                document.querySelector('form[action="/eliminar_usuario"]
input[name="id"]').value = id;
                document.querySelector('form[action="/eliminar_usuario"]'
).submit();
            }
        }
    </script>

```

```
    }  
    </script>  
</body>  
  
</html>
```

CU3: Gestionar Tipo relación

➤ Controlador

```
➤ <?php  
➤  
➤ declare(strict_types=1);  
➤  
➤ require_once(' ../app/modelos/TipoRelacion/MTipoRelacion.php');  
➤ require_once(' ../app/vistas/TipoRelacion/VTipoRelacion.php');  
➤  
➤ class CTipoRelacion  
➤ {  
➤     private VTipoRelacion $vista;  
➤     private MTipoRelacion $modelo;  
➤  
➤     public function __construct()  
➤     {  
➤         $this->vista = new VTipoRelacion();  
➤         $this->modelo = new MTipoRelacion();  
➤     }  
➤  
➤     public function listarTipoRelacionC(): void  
➤     {  
➤  
➤         $tipo_relaciones = $this->modelo->listarTipos();  
➤  
➤         $this->vista->actualizar($tipo_relaciones);  
➤     }  
➤  
➤     public function crearTipoRelacionC(string $nombre): void  
➤     {  
➤  
➤         $this->modelo->crearTipos($nombre);  
➤  
➤         $tipo_relaciones = $this->modelo->listarTipos();  
➤  
➤         $this->vista->actualizar($tipo_relaciones);  
➤     }  
➤  
➤     public function eliminarTipoRelacionC(int $id): void  
➤     {  
➤  
➤         $this->modelo->eliminarTipo($id);  
➤         $tipo_relaciones = $this->modelo->listarTipos();
```

```

➤         $this->vista->actualizar($tipo_relaciones);
➤     }
➤
➤     public function updateTipoRelacionC(int $id): void
➤     {
➤         $tipo_relacion = $this->modelo->buscarTipo($id);
➤         $this->vista->mostrarFormularioEdicion($tipo_relacion);
➤     }
➤
➤     public function editarTipoRelacionC(int $id, string $nombre):
void
➤     {
➤
➤         $this->modelo->editarTipo($id, $nombre);
➤
➤         $tipo_relaciones = $this->modelo->listarTipos();
➤
➤         $this->vista->actualizar($tipo_relaciones);
➤     }
➤ }
➤

```

➤ Modelo

```

➤ <?php
➤
➤ require_once(' ../app/modelos/iglesiaDB.php');
➤ require_once(' ../app/modelos/TipoRelacion/TipoRelacion.php');
➤
➤ class MTipoRelacion
➤ {
➤     private IglesiaDB $database;
➤
➤     public function __construct()
➤     {
➤         $this->database = new IglesiaDB();
➤     }
➤
➤     public function crearTipos($nombre): void
➤     {
➤
➤         $bd = $this->database->getConnection();
➤         try {
➤
➤             $query = "INSERT INTO " . $this->
➤ database::TABLE_TIPO_RELACION . " (nombre) VALUES (?)";
➤             $stmt = $bd->prepare($query);
➤             $stmt->bind_param("s", $nombre);
➤             if ($stmt->execute()) {

```

```

➤         error_log("Tipo de relación insertada con éxito");
➤     }
➤ } catch (Exception $e) {
➤
➤     error_log("Excepción al insertar el tipo de relación a
la base de datos: " . $e->getMessage());
➤ } finally {
➤
➤     if (isset($stmt)) {
➤         $stmt->close();
➤     }
➤     $bd->close();
➤ }
➤ }
➤
➤ public function listarTipos(): array
➤ {
➤     $bd = $this->database->getConnection();
➤
➤     $tiposRelacion = []; // Un arreglo para almacenar objetos
TipoRelacion
➤
➤     try {
➤         $result = $bd->query('SELECT * FROM ' . $this-
➤>database::TABLE_TIPO_RELACION);
➤
➤         if ($result) {
➤             while ($row = $result->fetch_assoc()) {
➤                 // Crear un objeto TipoRelacion a partir de los
datos de la fila
➤                 $tipoRelacion = new TipoRelacion($row['id'],
➤ $row['nombre']);
➤                 $tiposRelacion[] = $tipoRelacion; // Agregar el
objeto al arreglo
➤             }
➤         }
➤     } catch (Exception $e) {
➤         error_log(" Excepción en getTipoRelacion: " . $e-
➤>getMessage());
➤     } finally {
➤         $bd->close();
➤     }
➤
➤     return $tiposRelacion;
➤ }
➤
➤ public function buscarTipo($id)
➤ {
➤     $bd = $this->database->getConnection();

```

```

➤
➤
➤      try {
➤          $query = "SELECT * FROM " . $this->
➤ database::TABLE_TIPO_RELACION . " WHERE id = ?";
➤          $stmt = $bd->prepare($query);
➤          $stmt->bind_param("i", $id);
➤          $stmt->execute();
➤          $result = $stmt->get_result();
➤
➤          if ($result && $result->num_rows > 0) {
➤              $row = $result->fetch_assoc();
➤              // Crear y devolver un objeto TipoRelacion a partir
➤ de los datos de la fila
➤              $tipoRelacion = new TipoRelacion($row['id'],
➤ $row['nombre']);
➤              return $tipoRelacion;
➤          }
➤      } catch (Exception $e) {
➤          error_log("Excepción en buscarTipo: " . $e->
➤ getMessage());
➤      } finally {
➤          $bd->close();
➤      }
➤
➤      // Si no se encontró una coincidencia, puedes devolver un
➤ valor nulo
➤      return null;
➤  }
➤
➤  public function editarTipo($id, $nombre): void
➤  {
➤      $bd = $this->database->getConnection();
➤
➤      try {
➤          $query = "UPDATE " . $this->
➤ database::TABLE_TIPO_RELACION . " SET nombre = ? WHERE id = ?";
➤          $stmt = $bd->prepare($query);
➤          $stmt->bind_param("si", $nombre, $id);
➤
➤          if ($stmt->execute()) {
➤              error_log("Tipo de relación editada con éxito");
➤          }
➤      } catch (Exception $e) {
➤          error_log("Excepción al editar el tipo de relación: " .
➤ $e->getMessage());
➤      } finally {
➤
➤          if (isset($stmt)) {
➤              $stmt->close();
➤          }
➤      }
➤  }

```



```

➤         }
➤         $bd->close();
➤     }
➤ }
➤
➤ public function eliminarTipo($id): void
➤ {
➤     $bd = $this->database->getConnection();
➤
➤     try {
➤         $query = "DELETE FROM " . $this->
➤ >database::TABLE_TIPO_RELACION . " WHERE id = ?";
➤         $stmt = $bd->prepare($query);
➤         $stmt->bind_param("i", $id);
➤
➤         if ($stmt->execute()) {
➤             error_log("Tipo de relación eliminada con éxito");
➤         }
➤     } catch (Exception $e) {
➤         error_log("Excepción al eliminar el tipo de relación: "
➤ . $e->getMessage());
➤     } finally {
➤
➤         if (isset($stmt)) {
➤             $stmt->close();
➤         }
➤         $bd->close();
➤     }
➤ }
➤ }
➤ }

```

```

<?php

class TipoRelacion {

    private $id;
    private $nombre;

    public function __construct(int $id, string $nombre) {
        $this->id = $id;
        $this->nombre = $nombre;
    }

    public function getId() {
        return $this->id;
    }
}

```

```

public function setId($id) {
    $this->id = $id;
}

public function getNombre() {
    return $this->nombre;
}

public function setNombre($nombre) {
    $this->nombre = $nombre;
}

public function __toString() {
    return "TipoRelacion{" .
        "id=" . $this->id .
        ", nombre=" . $this->nombre . '\'' .
        '}';
}
}

```

➤ Vista

```

➤ <?php
➤
➤ declare(strict_types=1);
➤
➤ class VTipoRelacion
➤ {
➤     private function renderizarTabla($rows): string
➤     {
➤         $rowData = '';
➤
➤         if (empty($rows)) {
➤             return "<tbody></tbody>";
➤         }
➤
➤         foreach ((array)$rows as $row) {
➤             $rowData .= "<tr>";
➤             $rowData .= "<td>{$row->getId()}</td>";
➤             $rowData .= "<td>{$row->getNombre()}</td>";
➤             $rowData .= "<td><a href='/editar_tipo_relacion?id={$row-
➤ >getId()}' class='edit-button'>Editar</a></td>";
➤             $rowData .= "<td>";
➤             $rowData .= "<form id='form_boton_eliminar' method='post'
➤ action='/eliminar_tipo_relacion'>";
➤             $rowData .= "<input type='hidden' name='id' value='{ $row-
➤ >getId()}'>";

```

```

➤         $rowData .= "<button type='button'
onclick='confirmarEliminar({$row->getId()})' class='delete-button'
style='font-size: 16px;'>Eliminar</button>";
➤         $rowData .= "</form>";
➤         $rowData .= "</td>";
➤         $rowData .= "</tr>";
➤     }
➤
➤     return "<tbody>$rowData</tbody>";
➤ }
➤
➤ public function actualizar($tipo_relaciones): void
➤ {
➤     $title = 'Tipo Relación - Usuarios';
➤     $tbody = $this->renderizarTabla($tipo_relaciones);
➤     include '../app/vistas/TipoRelacion/index.php';
➤ }
➤
➤ private function renderizarFormularioEdicion($tipo_relacion):
string
➤ {
➤     $formulario = '';
➤
➤     if (!empty($tipo_relacion) && $tipo_relacion instanceof
TipoRelacion) {
➤         $formulario .= "<form method='post'
action='/actualizar_tipo_relacion'>";
➤         $formulario .= "<input type='hidden' name='id'
value='{ $tipo_relacion->getId()}'>";
➤         $formulario .= "<div class='container-fluid'>";
➤         $formulario .= "<div class='row'>";
➤         $formulario .= "<div class='col-12 col-md-4'>";
➤         $formulario .= "<label for='nombre'>Nombres:</label>";
➤         $formulario .= "<input type='text' id='nombre' name='nombre'
value='{ $tipo_relacion->getNombre()}' required>";
➤         $formulario .= "</div>";
➤         $formulario .= "</div>";
➤         $formulario .= "<div class='row mt-3'>";
➤         $formulario .= "<div class='col-12 d-flex justify-content-
end'>";
➤         $formulario .= "<button class='add-button'
type='submit'>Actualizar</button>";
➤         $formulario .= "<a href='/tipo_relacion' class='back-
button'>Volver</a>";
➤         $formulario .= "</div>";
➤         $formulario .= "</div>";
➤         $formulario .= "</form>";
➤     } else {

```

```

➤     $formulario .= "El tipo de relación no se encontró o no
existe.";
➤ }
➤
➤     return $formulario;
➤
➤ }
➤
➤     public function mostrarFormularioEdicion($tipo_relacion): void
➤     {
➤         $title = 'Tipo Relación - Editar';
➤         $formulario = $this-
➤ >renderizarFormularioEdicion($tipo_relacion);
➤         include '../app/vistas/TipoRelacion/edit.php';
➤     }
➤ }

```

```

<!DOCTYPE html>
<html>

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Iglesia App</title>
    <link rel="stylesheet"
href="https://fonts.googleapis.com/css?family=Lora&display=swap">
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.mi
n.css">
    <link rel="stylesheet" href="css/estilo.css">
</head>

<body>
    <header>
        <div class="container">
            <h1>Iglesia</h1>
        </div>
    </header>
    <nav class="sidebar">
        <a href="/menu">Inicio</a>
        <a href="/usuarios">Usuario</a>

        <a href="/cargos">Cargo</a>
        <a href="/ministerios">Ministerio</a>
        <a href="/eventos">Evento</a>
        <a href="/eventos">Registro Eventos</a>
        <a href="/tipo_relacion">Tipo Relacion</a>
    </nav>

```

```

        <a href="/parentescos">Parentesco</a>
    </nav>
    <main>
    <div class="container">
        <h2>Nuevo Tipo Relacion</h2>
    </div>
    <div class="container">
        <div id='form_tipo_relacion'>
            <form action="/tipo_relacion" method="post">
                <div class="row">
                    <div class="col-12 col-md-12">
                        <label for="nombre">Nombre</label>
                        <input name="nombre" type="text"
id="nombre" required>
                    </div>
                </div>
                <div class="row mt-3">
                    <div class="col-12 d-flex justify-content-
end">
                        <button class="add-button"
type="submit">Agregar</button>
                    </div>
                </div>
            </form>
        </div>
    </div>

    <h1><?= $title ?></h1>
    <table class="table">
        <thead>
            <tr>
                <th>Id</th>
                <th>Nombre</th> <!-- Cambiado a Nombre -->
            </tr>
        </thead>
        <?= $tbody ?>
    </table>
    </div>
</main>

<footer>
    <div class="container text-center">
        <p>&copy; <?php echo date("Y"); ?> VeronicaAntezana</p>
    </div>
</footer>
<script>
    function confirmarEliminar(id) {

```

```

        if (confirm('¿Estás seguro de que deseas eliminar este
ministerio?')) {
            // Si el usuario confirma, enviar el formulario de
eliminación
            document.querySelector('form[action="/eliminar_tipo_relac
ion"] input[name="id"]').value = id;
            document.querySelector('form[action="/eliminar_tipo_relac
ion"]').submit();
        }
    }
</script>
</body>

</html>

```

```

<!DOCTYPE html>
<html>

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Iglesia App</title>
    <link rel="stylesheet"
href="https://fonts.googleapis.com/css?family=Lora&display=swap">
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.mi
n.css">
    <link rel="stylesheet" href="css/estilo.css">
</head>

<body>
    <header>
        <div class="container">
            <h1>Iglesia App</h1>
        </div>
    </header>

    <main>
        <div class="container">
            <h2><?= $title ?></h2>

            <!-- Muestra el formulario de edición aquí -->
            <?= $formulario ?>

            <!-- Más contenido HTML si es necesario -->
        </div>
    </main>

```

```

<footer>
    <div class="container">
        <p>&copy; <?php echo date("Y"); ?> Veronica Antezana</p>
    </div>
</footer>
<script>
    function confirmarEliminar(id) {
        if (confirm('¿Estás seguro de que deseas eliminar este Tipo
de Relacion?')) {
            // Si el usuario confirma, enviar el formulario de
eliminación
            document.querySelector('form[action="/eliminar_tipo_relac
ion"] input[name="id"]').value = id;
            document.querySelector('form[action="/eliminar_tipo_relac
ion"]').submit();
        }
    }
</script>
</body>
</html>

```

CU4: Gestionar Parentesco

```

<?php

declare(strict_types=1);

require_once(' ../app/modelos/Parentesco/MParentesco.php');
require_once(' ../app/modelos/Usuario/MUsuario.php');
require_once(' ../app/modelos/TipoRelacion/MTipoRelacion.php');
require_once(' ../app/vistas/Parentesco/VParentesco.php');

class CParentesco
{
    private VParentesco $vistaParentesco;
    private MTipoRelacion $modeloTipoRelacion;
    private MUsuario $modeloUsuario;
    private MParentesco $modeloParentesco;

    public function __construct()
    {
        $this->vistaParentesco = new VParentesco();
        $this->modeloTipoRelacion = new MTipoRelacion();
        $this->modeloUsuario = new MUsuario();
        $this->modeloParentesco = new MParentesco();
    }

    public function listarParentescosC(): void

```

```

{
    $relaciones = $this->modeloParentesco->listarParentescos();
    $usuarios = $this->modeloUsuario->listarUsuarios();
    $tiposRelaciones = $this->modeloTipoRelacion->listarTipos();
    $this->vistaParentesco->actualizar($relaciones, $usuarios,
$tiposRelaciones);
}

    public function agregarParentescoC(int $usuarioA, int $usuarioB, int
$tipoRelacionA): void
    {
        $this->modeloParentesco->agregarParentesco($usuarioA, $usuarioB,
$tipoRelacionA);

        $relaciones = $this->modeloParentesco->listarParentescos();
        $usuarios = $this->modeloUsuario->listarUsuarios();
        $tiposRelaciones = $this->modeloTipoRelacion->listarTipos();
        $this->vistaParentesco->actualizar($relaciones, $usuarios,
$tiposRelaciones);
    }

    public function eliminarRelacionC(int $id): void
    {
        $this->modeloParentesco->eliminarParentesco($id);

        $relaciones = $this->modeloParentesco->listarParentescos();
        $usuarios = $this->modeloUsuario->listarUsuarios();
        $tiposRelaciones = $this->modeloTipoRelacion->listarTipos();
        $this->vistaParentesco->actualizar($relaciones, $usuarios,
$tiposRelaciones);
    }

    public function editarRelacionC(int $id, int $usuarioA, int
$usuarioB, int $tipoRelacionA): void
    {
        $this->modeloParentesco->editarParentesco($id, $usuarioA,
$usuarioB, $tipoRelacionA);

        $relaciones = $this->modeloParentesco->listarParentescos();
        $usuarios = $this->modeloUsuario->listarUsuarios();
        $tiposRelaciones = $this->modeloTipoRelacion->listarTipos();
        $this->vistaParentesco->actualizar($relaciones, $usuarios,
$tiposRelaciones);
    }

    public function updateRelacionC(int $id): void
    {
        $relacion = $this->modeloParentesco->buscarParentesco($id);
        $usuarios = $this->modeloUsuario->listarUsuarios();

```



```

        $tiposRelaciones = $this->modeloTipoRelacion->listarTipos();
        $this->vistaParentesco->mostrarFormularioEdicion($relacion,
        $usuarios, $tiposRelaciones);
    }
}

```

➤ Modelo

```

<?php
require_once(' ../app/modelos/iglesiaDB.php');
require_once(' ../app/modelos/Parentesco/Parentesco.php');

class MParentesco
{
    private iglesiaDB $database;

    public function __construct()
    {
        $this->database = new iglesiaDB();
    }

    // Función para agregar una nueva relación
    public function agregarParentesco($usuarioA, $usuarioB,
    $tipoRelacionA ): void
    {
        $bd = $this->database->getConnection();
        try {
            $query = "INSERT INTO " . $this->
            >database::TABLE_PARENTESCO . " (usuario_a, usuario_b,
            tipo_relacion_a) VALUES (?, ?, ?)";
            $stmt = $bd->prepare($query);
            $stmt->bind_param("iii", $usuarioA, $usuarioB,
            $tipoRelacionA);
            if ($stmt->execute()) {
                error_log("Relación insertada con éxito");
            }
        } catch (Exception $e) {
            error_log("Excepción al insertar la relación a la base
            de datos: " . $e->getMessage());
        } finally {
            if (isset($stmt)) {
                $stmt->close();
            }
            $bd->close();
        }
    }
}

```

```

➤ // Función para obtener todas las relaciones
➤ public function listarParentescos(): array
➤ {
➤     $bd = $this->database->getConnection();
➤     $relaciones = [];
➤
➤     try {
➤         $result = $bd->query('SELECT * FROM ' . $this->
➤ database::TABLE_PARENTESCO);
➤
➤         if ($result) {
➤             while ($row = $result->fetch_assoc()) {
➤                 $relacion = new Parentesco($row['id'],
➤ $row['usuario_a'], $row['usuario_b'], $row['tipo_relacion_a']);
➤                 $relaciones[] = $relacion;
➤             }
➤         }
➤     } catch (Exception $e) {
➤         error_log("Excepción en obtenerRelaciones: " . $e-
➤ >getMessage());
➤     } finally {
➤         $bd->close();
➤     }
➤
➤     return $relaciones;
➤ }
➤
➤ // Función para buscar una relación por su ID
➤ public function buscarParentesco($id)
➤ {
➤     $bd = $this->database->getConnection();
➤     try {
➤         $query = "SELECT * FROM " . $this->
➤ database::TABLE_PARENTESCO . " WHERE id = ?";
➤         $stmt = $bd->prepare($query);
➤         $stmt->bind_param("i", $id);
➤         $stmt->execute();
➤         $result = $stmt->get_result();
➤
➤         if ($result && $result->num_rows > 0) {
➤             $row = $result->fetch_assoc();
➤             $relacion = new Parentesco($row['id']
➤ , $row['usuario_a'], $row['usuario_b'], $row['tipo_relacion_a']);
➤             return $relacion;
➤         }
➤     } catch (Exception $e) {
➤         error_log("Excepción en buscarRelacion: " . $e-
➤ >getMessage());
➤     } finally {

```

```

➤         $bd->close();
➤     }
➤
➤     return null;
➤ }
➤
➤ // Función para editar una relación
➤ public function editarParentesco($id, $usuarioA, $usuarioB,
➤ $tipoRelacionA): void
➤ {
➤     $bd = $this->database->getConnection();
➤     try {
➤         $query = "UPDATE " . $this->database::TABLE_PARENTESCO
➤ . " SET usuario_a = ?, usuario_b = ?, tipo_relacion_a = ? WHERE id
➤ = ?";
➤
➤         $stmt = $bd->prepare($query);
➤         $stmt->bind_param("iiii", $usuarioA, $usuarioB,
➤ $tipoRelacionA, $id);
➤
➤         if ($stmt->execute()) {
➤             error_log("Relación editada con éxito");
➤         }
➤     } catch (Exception $e) {
➤         error_log("Excepción al editar la relación: " . $e-
➤ >getMessage());
➤     } finally {
➤         if (isset($stmt)) {
➤             $stmt->close();
➤         }
➤         $bd->close();
➤     }
➤ }
➤
➤ // Función para eliminar una relación por su ID
➤ public function eliminarParentesco($id): void
➤ {
➤     $bd = $this->database->getConnection();
➤     try {
➤         $query = "DELETE FROM " . $this-
➤ >database::TABLE_PARENTESCO . " WHERE id = ?";
➤         $stmt = $bd->prepare($query);
➤         $stmt->bind_param("i", $id);
➤
➤         if ($stmt->execute()) {
➤             error_log("Relación eliminada con éxito");
➤         }
➤     } catch (Exception $e) {
➤         error_log("Excepción al eliminar la relación: " . $e-
➤ >getMessage());

```

```

➤         } finally {
➤             if (isset($stmt)) {
➤                 $stmt->close();
➤             }
➤             $bd->close();
➤         }
➤     }
➤ }
➤

```

```

<?php

class Parentesco {
    private $id;
    private $usuarioA;
    private $usuarioB;
    private $tipoRelacionA;

    public function __construct($id, $usuarioA, $usuarioB,
$tipoRelacionA) {
        $this->id = $id;
        $this->usuarioA = $usuarioA;
        $this->usuarioB = $usuarioB;
        $this->tipoRelacionA = $tipoRelacionA;

    }

    public function getId() {
        return $this->id;
    }

    public function getUsuarioA() {
        return $this->usuarioA;
    }

    public function setUsuarioA($usuarioA) {
        $this->usuarioA = $usuarioA;
    }

    public function getUsuarioB() {
        return $this->usuarioB;
    }

    public function setUsuarioB($usuarioB) {
        $this->usuarioB = $usuarioB;
    }
}

```

```

public function getTipoRelacionA() {
    return $this->tipoRelacionA;
}

public function setTipoRelacionA($tipoRelacionA) {
    $this->tipoRelacionA = $tipoRelacionA;
}

public function __toString() {
    return "Relación [ID: {$this->id}, Usuario A: {$this->usuarioA},
Usuario B: {$this->usuarioB}, Tipo de Relación A: {$this->tipoRelacionA}]";
}
}

```

➤ Vista

```

➤ <?php
➤
➤ declare(strict_types=1);
➤
➤ class VParentesco
➤ {
➤     private function renderizarTabla($relaciones, $usuarios,
➤ $tiposRelaciones): string
➤     {
➤         $rowData = '';
➤
➤         if (empty($relaciones)) {
➤             return "<tbody></tbody>";
➤         }
➤
➤         foreach ((array)$relaciones as $relacion) {
➤             $rowData .= "<tr>";
➤             // $rowData .= "<td>{$relacion->getId()}</td>";
➤
➤             // Buscar Usuario A por ID
➤             $usuarioA = null;
➤             foreach ($usuarios as $usuario) {
➤                 if ($usuario->getId() === $relacion->getUsuarioA())
➤ {
➤                     $usuarioA = $usuario;
➤                     break;
➤                 }
➤             }
➤
➤             $rowData .= "<td>";

```

```

➤         $rowData .= ($usuarioA ? $usuarioA->getNombre() . " " .
$usuarioA->getApellido() : "Usuario no encontrado");
➤         $rowData .= "</td>";
➤
➤         // Buscar Usuario B por ID
➤         $usuarioB = null;
➤         foreach ($usuarios as $usuario) {
➤             if ($usuario->getId() === $relacion->getUsuarioB())
➤         {
➤                 $usuarioB = $usuario;
➤                 break;
➤             }
➤         }
➤
➤         $rowData .= "<td>";
➤         $rowData .= ($usuarioB ? $usuarioB->getNombre() . " " .
$usuarioB->getApellido() : "Usuario no encontrado");
➤         $rowData .= "</td>";
➤
➤         $nombreRelacion = "";
➤         foreach ($tiposRelaciones as $tipoRelacion) {
➤             if ($tipoRelacion->getId() == $relacion-
➤ >getTipoRelacionA()) {
➤                 $rowData .= "<td>{$tipoRelacion-
➤ >getNombre()}</td>";
➤                 break;
➤             }
➤         }
➤         // Buscar Tipo de Relación A por ID
➤         /*$tipoRelacionA = null;
➤         foreach ($tiposRelaciones as $tipoRelacion) {
➤             if ($tipoRelacion->getId() === $relacion-
➤ >getTipoRelacionA()) {
➤                 $tipoRelacionA = $tipoRelacion;
➤                 break;
➤             }
➤         }
➤
➤         $rowData .= "<td>";
➤         $rowData .= ($tipoRelacionA ? $tipoRelacionA-
➤ >getNombre() : "Tipo de relación no encontrado");
➤         $rowData .= "</td>";
➤
➤         */
➤         // Buscar Tipo de Relación B por ID
➤
➤         $rowData .= "<td><a
href='/editar_parentesco?id={$relacion->getId()}' class='edit-
button'>Editar</a></td>";

```

```

➤         $rowData .= "<td>";
➤         $rowData .= "<form id='form_boton_eliminar'
method='post' action='/eliminar_parentesco'>";
➤         $rowData .= "<input type='hidden' name='id'
value='{ $relacion->getId()}'>";
➤         $rowData .= "<button type='button'
onclick='confirmarEliminar({ $relacion->getId()})' class='delete-
button' style='font-size: 16px;'>Eliminar</button>";
➤         $rowData .= "</form>";
➤         $rowData .= "</td>";
➤
➤         $rowData .= "</tr>";
➤     }
➤
➤     return "<tbody>$rowData</tbody>";
➤ }
➤
➤ private function renderizarFormularioCreacion($usuarios,
$tiposRelaciones): string
➤ {
➤     $formulario = '';
➤
➤     $formulario .= "<div class='container'>";
➤     $formulario .= "<h2>Nuevo Parentesco</h2>";
➤     $formulario .= "</div>";
➤     $formulario .= "<div class='container'>";
➤     $formulario .= "<form method='post'
action='/parentescos'>";
➤
➤     $formulario .= "<div class='row'>";
➤     $formulario .= "<div class='col-12 col-md-6'>";
➤     $formulario .= "<label for='usuarioA'>Usuario A</label>";
➤     $formulario .= "<select name='usuarioA' id='usuarioA'
required>";
➤     foreach ($usuarios as $usuario) {
➤         $formulario .= "<option value='{ $usuario-
>getId()}'>{ $usuario->getNombre() } { $usuario-
>getApellido()}</option>";
➤     }
➤     $formulario .= "</select>";
➤     $formulario .= "</div>";
➤
➤     $formulario .= "<div class='col-12 col-md-6'>";
➤     $formulario .= "<label for='usuarioB'>Usuario B</label>";
➤     $formulario .= "<select name='usuarioB' id='usuarioB'
required>";
➤     foreach ($usuarios as $usuario) {

```

```

➤         $formulario .= "<option value='{$_usuario->getId()}'>{$_usuario->getNombre()} {$_usuario->getApellido()}</option>";
➤     }
➤     $formulario .= "</select>";
➤     $formulario .= "</div>";
➤
➤     $formulario .= "<div class='col-12 col-md-6'>";
➤     $formulario .= "<label for='tipoRelacionA'>Tipo de Relación
A</label>";
➤     $formulario .= "<select name='tipoRelacionA'
id='tipoRelacionA' required>";
➤         foreach ($tiposRelaciones as $tipoRelacion) {
➤             $formulario .= "<option value='{$_tipoRelacion->getId()}'>{$_tipoRelacion->getNombre()}</option>";
➤         }
➤     $formulario .= "</select>";
➤     $formulario .= "</div>";
➤
➤     $formulario .= "</div>";
➤     $formulario .= "<div class='row mt-3'>";
➤     $formulario .= "<div class='col-12 d-flex justify-content-
end'>";
➤         $formulario .= "<button class='add-button'
type='submit'>Agregar</button>";
➤         $formulario .= "<a href='/menu' class='back-
button'>Volver</a>";
➤     $formulario .= "</div>";
➤     $formulario .= "</div>";
➤     $formulario .= "</form>";
➤     $formulario .= "</div>";
➤
➤     return $formulario;
➤ }
➤
➤     public function actualizar($relaciones, $usuarios,
$tiposRelaciones): void
➤     {
➤         $title = 'Parentescos';
➤         $tbody = $this->renderizarTabla($relaciones, $usuarios,
$tiposRelaciones);
➤         $formulario = $this->
renderizarFormularioCreacion($usuarios, $tiposRelaciones);
➤         include '../app/vistas/Parentesco/index.php';
➤     }
➤
➤     private function renderizarFormularioEdicion($relacion,
$usuarios, $tiposRelaciones): string
➤     {

```



```

➤ $formulario = '';
➤
➤ if (!empty($relacion) && $relacion instanceof Parentesco) {
➤     $formulario .= "<form method='post'
➤ action='/actualizar_parentesco'>";
➤     $formulario .= "<input type='hidden' name='id'
➤ value='{ $relacion->getId()}'>";
➤     $formulario .= "<div class='container-fluid'>";
➤     $formulario .= "<div class='row'>";
➤     $formulario .= "<div class='col-12 col-md-6'>"; //
➤ Columna 1 para usuarios
➤
➤     // Para el campo 'Usuario A'
➤     $formulario .= "<label for='usuarioA'>Usuario
➤ A:</label>";
➤     $formulario .= "<select id='usuarioA' name='usuarioA'
➤ required>";
➤     foreach ($usuarios as $usuario) {
➤         $selected = ($usuario->getId() == $relacion-
➤ >getUsuarioA()) ? 'selected' : '';
➤         $formulario .= "<option value='{ $usuario->getId()}'
➤ $selected>{ $usuario->getNombre()}</option>";
➤     }
➤     $formulario .= "</select>";
➤
➤     // Continuar con más campos de usuario...
➤
➤     $formulario .= "</div>";
➤     $formulario .= "<div class='col-12 col-md-6'>"; //
➤ Columna 2 para usuarios
➤
➤     // Para el campo 'Usuario B'
➤     $formulario .= "<label for='usuarioB'>Usuario
➤ B:</label>";
➤     $formulario .= "<select id='usuarioB' name='usuarioB'
➤ required>";
➤     foreach ($usuarios as $usuario) {
➤         $selected = ($usuario->getId() == $relacion-
➤ >getUsuarioB()) ? 'selected' : '';
➤         $formulario .= "<option value='{ $usuario->getId()}'
➤ $selected>{ $usuario->getNombre()}</option>";
➤     }
➤     $formulario .= "</select>";
➤
➤     $formulario .= "</div>";
➤     $formulario .= "</div>";
➤
➤     // Ahora, dividir los campos de relaciones en dos
➤ columnas de 6

```

```

➤         $formulario .= "<div class='row'>";
➤         $formulario .= "<div class='col-12 col-md-6'>"; //
Columna 1 para tipos de relaciones
➤
➤         // Para el campo 'Tipo de Relación A'
➤         $formulario .= "<label for='tipoRelacionA'>Tipo de
Relación A:</label>";
➤         $formulario .= "<select id='tipoRelacionA'
name='tipoRelacionA' required>";
➤         foreach ($tiposRelaciones as $tipoRelacion) {
➤             $selected = ($tipoRelacion->getId() == $relacion-
➤>getTipoRelacionA()) ? 'selected' : '';
➤
➤             $formulario .= "<option value='{ $tipoRelacion-
➤>getId()}' $selected>{ $tipoRelacion->getNombre()}</option>";
➤             }
➤             $formulario .= "</select>";
➤
➤         // $formulario .= "<div class='col-12 col-md-6'>"; //
Columna 2 para tipos de relaciones
➤
➤         // Para el campo 'Tipo de Relación B'
➤         // Esto en adelante esta igual
➤         // $formulario .= "</select>";
➤
➤         $formulario .= "</div>";
➤         $formulario .= "</div>";
➤         $formulario .= "<div class='row mt-3'>";
➤         $formulario .= "<div class='col-12 d-flex justify-
content-end'>";
➤         $formulario .= "<button class='add-button'
type='submit'>Actualizar</button>";
➤         $formulario .= "<a href='/parentescos' class='back-
button'>Volver</a>";
➤         $formulario .= "</div>";
➤         $formulario .= "</div>";
➤         $formulario .= "</form>";
➤     } else {
➤         $formulario .= "La relación no se encontró o no
existe.";
➤     }
➤
➤     return $formulario;
➤ }
➤
➤ public function mostrarFormularioEdicion($relacion, $usuarios,
$tiposRelaciones): void

```

```

➤ {
➤     $title = 'Parentesco - Editar';
➤     $formulario = $this->renderizarFormularioEdicion($relacion,
➤ $usuarios, $tiposRelaciones);
➤     include '../app/vistas/Parentesco/edit.php';
➤ }
➤ }
➤
➤ <!DOCTYPE html>
➤ <html>
➤
➤ <head>
➤     <meta charset="UTF-8">
➤     <meta name="viewport" content="width=device-width, initial-
➤ scale=1">
➤     <title>Iglesia App</title>
➤     <link rel="stylesheet"
➤ href="https://fonts.googleapis.com/css?family=Lora&display=swap">
➤     <link rel="stylesheet"
➤ href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootst
➤ rap.min.css">
➤     <link rel="stylesheet" href="css/estilo.css">
➤ </head>
➤ <nav class="sidebar">
➤     <a href="/menu">Inicio</a>
➤     <a href="/usuarios">Usuario</a>
➤
➤     <a href="/cargos">Cargo</a>
➤     <a href="/ministerios">Ministerio</a>
➤     <a href="/eventos">Evento</a>
➤     <a href="/registro_evento">Registro Eventos</a>
➤     <a href="/tipo_relacion">Tipo Relacion</a>
➤     <a href="/parentescos">Parentesco</a>
➤ </nav>
➤ <body>
➤     <header>
➤         <div class="container">
➤             <h1>Iglesia App</h1>
➤         </div>
➤     </header>
➤     <nav>
➤         <div class="container">
➤
➤         </div>
➤     </nav>
➤     <main>
➤         <?= $formulario ?>
➤         <h1><?= $title ?></h1>
➤         <table>

```

```

➤         <thead>
➤             <tr>
➤
➤                 <th>Usuario_A</th>
➤                 <th>Usuario_B</th>
➤                 <th>Tipo Relacion</th>
➤             </tr>
➤         </thead>
➤         <?= $tbody ?>
➤     </table>
➤ </div>
➤ </main>
➤ <footer>
➤     <div class="container">
➤         <p>&copy; <?php echo date("Y"); ?> VeronicaAntezana</p>
➤     </div>
➤ </footer>
➤ <script>
➤     function confirmarEliminar(id) {
➤         if (confirm('¿Estás seguro de que deseas eliminar este
➤ usuario?')) {
➤             // Si el usuario confirma, enviar el formulario de
➤ eliminación
➤             document.querySelector('form[action="/eliminar_pare
➤ ntesco"] input[name="id"]').value = id;
➤             document.querySelector('form[action="/eliminar_pare
➤ ntesco"]').submit();
➤         }
➤     }
➤ </script>
➤ </body>
➤
➤ </html>
➤ <!DOCTYPE html>
➤ <html>
➤
➤ <head>
➤     <meta charset="UTF-8">
➤     <meta name="viewport" content="width=device-width, initial-
➤ scale=1">
➤     <title>Iglesia App</title>
➤     <link rel="stylesheet"
➤ href="https://fonts.googleapis.com/css?family=Lora&display=swap">
➤     <link rel="stylesheet"
➤ href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootst
➤ rap.min.css">
➤     <link rel="stylesheet" href="css/estilo.css">
➤ </head>

```

```

> <body>
>   <header>
>     <div class="container">
>       <h1>Iglesia App</h1>
>     </div>
>   </header>
>
>   <main>
>     <div class="container">
>       <h2><?= $title ?></h2>
>
>       <!-- Muestra el formulario de edición aquí -->
>       <?= $formulario ?>
>
>       <!-- Más contenido HTML si es necesario -->
>     </div>
>   </main>
>   <footer>
>     <div class="container">
>       <p>&copy; <?php echo date("Y"); ?> Veronica
Antezana</p>
>     </div>
>   </footer>
>   <script>
>     function confirmarEliminar(id) {
>       if (confirm('¿Estás seguro de que deseas eliminar este
Tipo de Relacion?')) {
>         // Si el usuario confirma, enviar el formulario de
eliminación
>         document.querySelector('form[action="/eliminar_pare
ntesco"] input[name="id"]').value = id;
>         document.querySelector('form[action="/eliminar_pare
ntesco"]').submit();
>       }
>     }
>   </script>
> </body>
>
> </html>

```

CU5: Gestionar Evento

Controlador

```

<?php

declare(strict_types=1);

require_once(' ../app/modelos/Evento/MEvento.php');

```

```
require_once('../app/vistas/Evento/VEvento.php');

class CEvento
{
    private VEvento $vista;
    private MEvento $modelo;

    public function __construct()
    {
        $this->vista = new VEvento();
        $this->modelo = new MEvento();
    }

    public function listarEventosC(): void
    {
        $tipo_relaciones = $this->modelo->listarEventos();

        $this->vista->actualizar($tipo_relaciones);
    }

    public function crearEventoC(string $nombre, string $descripcion): void
    {
        $this->modelo->crearEvento($nombre, $descripcion);

        $tipo_relaciones = $this->modelo->listarEventos();

        $this->vista->actualizar($tipo_relaciones);
    }

    public function eliminarEventoC(int $id): void
    {
        $this->modelo->eliminarEvento($id);

        $tipo_relaciones = $this->modelo->listarEventos();

        $this->vista->actualizar($tipo_relaciones);
    }

    public function updateEventoC(int $id): void
    {
        $tipo_relacion = $this->modelo->buscarEvento($id);

        $this->vista->mostrarFormularioEdicion($tipo_relacion);
    }
}
```

```

        public function editarEventoC(int $id, string $nombre, string
$descripcion): void
        {

            $this->modelo->editarEvento($id, $nombre, $descripcion);

            $tipo_relaciones = $this->modelo->listarEventos();

            $this->vista->actualizar($tipo_relaciones);

        }
    }
}

```

Modelo

```

<?php

require_once(' ../app/modelos/iglesiaDB.php');
require_once(' ../app/modelos/Evento/Evento.php');

class MEvento
{
    private IglesiaDB $database;

    public function __construct()
    {
        $this->database = new IglesiaDB();
    }

    public function crearEvento(string $nombre, string $descripcion):
void //crear
    {
        $bd = $this->database->getConnection();
        try {
            $query = "INSERT INTO " . $this->database::TABLE_EVENTO . "
(nombre, descripcion) VALUES (?, ?)";
            $stmt = $bd->prepare($query);
            $stmt->bind_param("ss", $nombre, $descripcion);
            if ($stmt->execute()) {
                error_log("Cargo insertado con éxito");
            }
        } catch (Exception $e) {
            error_log("Excepción al insertar el cargo en la base de
datos: " . $e->getMessage());
        } finally {
            if (isset($stmt)) {
                $stmt->close();
            }
        }
    }
}

```

```

        $bd->close();
    }
}

public function listarEventos(): array //listar
{
    $bd = $this->database->getConnection();

    $cargos = [];

    try {
        $result = $bd->query('SELECT * FROM ' . $this->database::TABLE_EVENTO);

        if ($result) {
            while ($row = $result->fetch_assoc()) {

                $cargo = new Evento($row['id'], $row['nombre'],
                $row['descripcion']);
                $cargos[] = $cargo;
            }
        }
    } catch (Exception $e) {
        error_log("Excepción en mostrarCargos: " . $e->getMessage());
    } finally {
        $bd->close();
    }

    return $cargos;
}

public function buscarEvento(int $id): Evento
{
    $bd = $this->database->getConnection();

    try {
        $query = "SELECT * FROM " . $this->database::TABLE_EVENTO . "
        WHERE id = ?";
        $stmt = $bd->prepare($query);
        $stmt->bind_param("i", $id);
        $stmt->execute();
        $result = $stmt->get_result();

        if ($result && $result->num_rows > 0) {
            $row = $result->fetch_assoc();
            // Crear y devolver un objeto Cargo a partir de los datos
            de la fila
            $cargo = new Evento($row['id'], $row['nombre'],
            $row['descripcion']);

```



```

        return $cargo;
    }
} catch (Exception $e) {
    error_log("Excepción en buscarCargo: " . $e->getMessage());
} finally {
    $bd->close();
}

return null;
}

public function editarEvento(int $id, string $nombre, string
$descripcion): void
{
    $bd = $this->database->getConnection();

    try {
        $query = "UPDATE " . $this->database::TABLE_EVENTO . " SET
nombre = ?, descripcion = ? WHERE id = ?";
        $stmt = $bd->prepare($query);
        $stmt->bind_param("ssi", $nombre, $descripcion, $id);

        if ($stmt->execute()) {
            error_log("Cargo editado con éxito");
        }
    } catch (Exception $e) {
        error_log("Excepción al editar el cargo: " . $e-
>getMessage());
    } finally {
        if (isset($stmt)) {
            $stmt->close();
        }
        $bd->close();
    }
}

public function eliminarEvento(int $id): void
{
    $bd = $this->database->getConnection();

    try {
        $query = "DELETE FROM " . $this->database::TABLE_EVENTO . "
WHERE id = ?";
        $stmt = $bd->prepare($query);
        $stmt->bind_param("i", $id);

        if ($stmt->execute()) {
            error_log("Cargo eliminado con éxito");
        }
    } catch (Exception $e) {
        error_log("Excepción al eliminar el cargo: " . $e->getMessage());
    } finally {
        if (isset($stmt)) {
            $stmt->close();
        }
        $bd->close();
    }
}

```

```

    }
    } catch (Exception $e) {
        error_log("Excepción al eliminar el cargo: " . $e-
>getMessage());
    } finally {
        if (isset($stmt)) {
            $stmt->close();
        }
        $bd->close();
    }
}
}
}

```

```

<?php

class Evento
{
    private int $id;
    private string $nombre;
    private string $descripcion;

    public function __construct(int $id, string $nombre, string
$descripcion)
    {
        $this->id = $id;
        $this->nombre = $nombre;
        $this->descripcion = $descripcion;
    }

    public function getId(): int
    {
        return $this->id;
    }

    public function getNombre(): string
    {
        return $this->nombre;
    }

    public function getDescripcion(): string
    {
        return $this->descripcion;
    }

    public function setId(int $id): void
    {
        $this->id = $id;
    }
}

```

```

    }

    public function setNombre(string $nombre): void
    {
        $this->nombre = $nombre;
    }

    public function setDescription(string $descripcion): void
    {
        $this->descripcion = $descripcion;
    }

    public function __toString(): string
    {
        return "Evento{" .
            "id=" . $this->id .
            ", nombre=" . $this->nombre . '\'' .
            ", descripcion=" . $this->descripcion . '\'' .
            '}';
    }
}

```

Vista

```

<?php

declare(strict_types=1);

class VEvento
{
    private function renderizarTabla(array $eventos): string
    {
        $rowData = '';

        if (empty($eventos)) {
            return "<tbody></tbody>";
        }

        foreach ((array)$eventos as $evento) {
            $rowData .= "<tr>";
            $rowData .= "<td>{$evento->getId()}</td>";
            $rowData .= "<td>{$evento->getNombre()}</td>";
            $rowData .= "<td>{$evento->getDescripcion()}</td>";
            $rowData .= "<td><a href='/editar_evento?id={$evento->getId()}' class='edit-button'>Editar</a></td>";
            $rowData .= "<td>";
        }
    }
}

```

```

        $rowData .= "<form id='form_boton_eliminar' method='post'
action='/eliminar_evento'>";
        $rowData .= "<input type='hidden' name='id' value='{ $evento-
>getId()}'>";
        $rowData .= "<button type='button'
onclick='confirmarEliminar({ $evento->getId()})' class='delete-button'
style='font-size: 16px;'>Eliminar</button>";
        $rowData .= "</form>";
        $rowData .= "</td>";
        $rowData .= "</tr>";
    }

    return "<tbody>$rowData</tbody>";
}

public function actualizar(array $eventos): void
{
    $title = 'Tipos de Eventos';
    $tbody = $this->renderizarTabla($eventos);
    include '../app/vistas/Evento/index.php';
}

private function renderizarFormularioEdicion(Evento $evento): string
{
    $formulario = '';

    if (!empty($evento) && $evento instanceof Evento) {
        $formulario .= "<form method='post'
action='/actualizar_evento'>";
        $formulario .= "<input type='hidden' name='id'
value='{ $evento->getId()}'>";
        $formulario .= "<div class='container-fluid'>";
        $formulario .= "<div class='row'>";
        $formulario .= "<div class='col-12 col-md-4'>";
        $formulario .= "<label for='nombre'>Nombres:</label>";
        $formulario .= "<input type='text' id='nombre' name='nombre'
value='{ $evento->getNombre()}' required>";
        $formulario .= "</div>";
        $formulario .= "<div class='col-12 col-md-4'>";
        $formulario .= "<label
for='descripcion'>Descripción:</label>";
        $formulario .= "<input type='text' id='descripcion'
name='descripcion' value='{ $evento->getDescripcion()}' required>";
        $formulario .= "</div>";
        $formulario .= "</div>";
        $formulario .= "<div class='row mt-3'>";
        $formulario .= "<div class='col-12 d-flex justify-content-
end'>";
    }

```

```

        $formulario .= "<button class='add-button'
type='submit'>Actualizar</button>";
        $formulario .= "<a href='/eventos' class='back-
button'>Volver</a>";
        $formulario .= "</div>";
        $formulario .= "</div>";
        $formulario .= "</form>";
    } else {
        $formulario .= "El evento no se encontró o no existe.";
    }

    return $formulario;
}

public function mostrarFormularioEdicion(Evento $evento): void
{
    $title = 'Evento - Editar';
    $formulario = $this->renderizarFormularioEdicion($evento);
    include '../app/vistas/Evento/edit.php';
}
}

<!DOCTYPE html>
<html>

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Iglesia App</title>
    <link rel="stylesheet"
href="https://fonts.googleapis.com/css?family=Lora&display=swap">
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.mi
n.css">
    <link rel="stylesheet" href="css/estilo.css">
</head>

<body>
    <header>
        <div class="container">
            <h1>Iglesia </h1>
        </div>
    </header>
    <nav class="sidebar">
        <a href="/menu">Inicio</a>
        <a href="/usuarios">Usuario</a>
        <a href="/cargos">Cargo</a>
        <a href="/ministerios">Ministerio</a>

```



```

        </div>
    </main>

    <footer>
        <div class="container text-center">
            <p>&copy; <?php echo date("Y"); ?> VeronicaAntezana</p>
        </div>
    </footer>
    <script>
        function confirmarEliminar(id) {
            if (confirm('¿Estás seguro de que deseas eliminar este
Evento?')) {
                // Si el usuario confirma, enviar el formulario de
eliminación
                document.querySelector('form[action="/eliminar_evento"]
input[name="id"]').value = id;
                document.querySelector('form[action="/eliminar_evento"]')
.submit();
            }
        }
    </script>
</body>

</html>
<!DOCTYPE html>
<html>

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Iglesia App</title>
    <link rel="stylesheet"
href="https://fonts.googleapis.com/css?family=Lora&display=swap">
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.mi
n.css">
    <link rel="stylesheet" href="css/estilo.css">
</head>

<body>
    <header>
        <div class="container">
            <h1>Iglesia App</h1>
        </div>
    </header>

    <main>
        <div class="container">
            <h2><?= $title ?></h2>

```

```

        <!-- Muestra el formulario de edición aquí -->
        <?= $formulario ?>

        <!-- Más contenido HTML si es necesario -->
    </div>
</main>
<footer>
    <div class="container">
        <p>&copy; <?php echo date("Y"); ?> Veronica Antezana</p>
    </div>
</footer>
<script>
    function confirmarEliminar(id) {
        if (confirm('¿Estás seguro de que deseas eliminar este
cargo?')) {
            // Si el usuario confirma, enviar el formulario de
eliminación
            document.querySelector('form[action="/eliminar_evento"]
input[name="id"]').value = id;
            document.querySelector('form[action="/eliminar_evento"]')
.submit();
        }
    }
</script>
</body>

</html>

```

CU6: Gestionar Ministerio

Controlador

```

<?php

declare(strict_types=1);

require_once(' ../app/modelos/Ministerio/MMinisterio.php');
require_once(' ../app/vistas/Ministerio/VMinisterio.php');

class CMinisterio
{
    private VMinisterio $vista;
    private MMinisterio $modelo;

    public function __construct()

```



```
{
    $this->vista = new VMinisterio();
    $this->modelo = new MMinisterio();
}

public function listarMinisteriosC(): void
{
    $ministerio = $this->modelo->listarMinisterios();

    $this->vista->actualizar($ministerio);
}

public function crearMinisterioC(string $nombre, string
$descripcion): void
{
    $this->modelo->crearMinisterio($nombre, $descripcion);

    $ministerio = $this->modelo->listarMinisterios();

    $this->vista->actualizar($ministerio);
}

public function eliminarMinisterioC(int $id): void
{
    $this->modelo->eliminarMinisterio($id);

    $ministerio = $this->modelo->listarMinisterios();

    $this->vista->actualizar($ministerio);
}

public function updateMinisterioC(int $id): void
{
    $ministerio = $this->modelo->buscarMinisterio($id);

    $this->vista->mostrarFormularioEdicion($ministerio);
}

public function editarMinisterioC(int $id, string $nombre, string
$descripcion): void
{
    $this->modelo->editarMinisterio($id, $nombre, $descripcion);

    $ministerio = $this->modelo->listarMinisterios();

    $this->vista->actualizar($ministerio);
}
```

```
}  
}
```

Modelo

```
<?php  
  
require_once('../app/modelos/iglesiaDB.php');  
require_once('../app/modelos/Ministerio/Ministerio.php');  
  
class MMinisterio  
{  
    private IglesiaDB $database;  
  
    public function __construct()  
    {  
        $this->database = new IglesiaDB();  
    }  
  
    public function crearMinisterio(string $nombre, string $descripcion):  
void //crear  
    {  
        $bd = $this->database->getConnection();  
        try {  
            $query = "INSERT INTO " . $this->database::TABLE_MINISTERIO .  
" (nombre, descripcion) VALUES (?, ?)";  
            $stmt = $bd->prepare($query);  
            $stmt->bind_param("ss", $nombre, $descripcion);  
            if ($stmt->execute()) {  
                error_log("Ministerio insertado con éxito");  
            }  
        } catch (Exception $e) {  
            error_log("Excepción al insertar el ministerio en la base de  
datos: " . $e->getMessage());  
        } finally {  
            if (isset($stmt)) {  
                $stmt->close();  
            }  
            $bd->close();  
        }  
    }  
  
    public function listarMinisterios(): array //listar  
    {  
        $bd = $this->database->getConnection();  
  
        $ministerios = [];  
  
        try {
```

```

        $result = $bd->query('SELECT * FROM ' . $this->database::TABLE_MINISTERIO);

        if ($result) {
            while ($row = $result->fetch_assoc()) {

                $ministerio = new Ministerio($row['id'],
                $row['nombre'], $row['descripcion']);
                $ministerios[] = $ministerio;
            }
        }
    } catch (Exception $e) {
        error_log("Excepción en mostrarMinisterios: " . $e->getMessage());
    } finally {
        $bd->close();
    }

    return $ministerios;
}

public function buscarMinisterio(int $id): Ministerio
{
    $bd = $this->database->getConnection();

    try {
        $query = "SELECT * FROM " . $this->database::TABLE_MINISTERIO
        . " WHERE id = ?";
        $stmt = $bd->prepare($query);
        $stmt->bind_param("i", $id);
        $stmt->execute();
        $result = $stmt->get_result();

        if ($result && $result->num_rows > 0) {
            $row = $result->fetch_assoc();
            // Crear y devolver un objeto Cargo a partir de los datos
            de la fila
            $ministerio = new Ministerio($row['id'], $row['nombre'],
            $row['descripcion']);
            return $ministerio;
        }
    } catch (Exception $e) {
        error_log("Excepción en buscarMinisterio: " . $e->getMessage());
    } finally {
        $bd->close();
    }

    return null;
}

```

```

    }

    public function editarMinisterio(int $id, string $nombre, string
$descripcion): void
    {
        $bd = $this->database->getConnection();

        try {
            $query = "UPDATE " . $this->database::TABLE_MINISTERIO . "
SET nombre = ?, descripcion = ? WHERE id = ?";
            $stmt = $bd->prepare($query);
            $stmt->bind_param("ssi", $nombre, $descripcion, $id);

            if ($stmt->execute()) {
                error_log("Ministerio editado con éxito");
            }
        } catch (Exception $e) {
            error_log("Excepción al editar el ministerio: " . $e-
>getMessage());
        } finally {
            if (isset($stmt)) {
                $stmt->close();
            }
            $bd->close();
        }
    }

    public function eliminarMinisterio(int $id): void
    {
        $bd = $this->database->getConnection();

        try {
            $query = "DELETE FROM " . $this->database::TABLE_MINISTERIO .
" WHERE id = ?";
            $stmt = $bd->prepare($query);
            $stmt->bind_param("i", $id);

            if ($stmt->execute()) {
                error_log("Ministerio eliminado con éxito");
            }
        } catch (Exception $e) {
            error_log("Excepción al eliminar el ministerio: " . $e-
>getMessage());
        } finally {
            if (isset($stmt)) {
                $stmt->close();
            }
            $bd->close();
        }
    }

```

```
}  
}  
}
```

```
<?php  
  
class Ministerio  
{  
    private int $id;  
    private string $nombre;  
    private string $descripcion;  
  
    public function __construct(int $id, string $nombre, string  
$descripcion)  
    {  
        $this->id = $id;  
        $this->nombre = $nombre;  
        $this->descripcion = $descripcion;  
    }  
  
    public function getId(): int  
    {  
        return $this->id;  
    }  
  
    public function getNombre(): string  
    {  
        return $this->nombre;  
    }  
  
    public function getDescripcion(): string  
    {  
        return $this->descripcion;  
    }  
  
    public function setId(int $id): void  
    {  
        $this->id = $id;  
    }  
  
    public function setNombre(string $nombre): void  
    {  
        $this->nombre = $nombre;  
    }  
  
    public function setDescripcion(string $descripcion): void  
    {
```

```

        $this->descripcion = $descripcion;
    }

    public function __toString(): string
    {
        return "Ministerio{" .
            "id=" . $this->id .
            ", nombre='" . $this->nombre . '\'' .
            ", descripcion='" . $this->descripcion . '\'' .
            '}';
    }
}

```

Vistas

```

<?php

declare(strict_types=1);

class VMinisterio
{
    private function renderizarTabla(array $ministerios): string
    {
        $rowData = '';

        if (empty($ministerios)) {
            return "<tbody></tbody>";
        }

        foreach ((array)$ministerios as $ministerio) {
            $rowData .= "<tr>";
            $rowData .= "<td>{$ministerio->getId()}</td>";
            $rowData .= "<td>{$ministerio->getNombre()}</td>";
            $rowData .= "<td>{$ministerio->getDescripcion()}</td>";
            $rowData .= "<td><a href='/editar_ministerio?id={$ministerio->getId()}' class='edit-button'>Editar</a></td>";
            $rowData .= "<td>";
            $rowData .= "<form id='form_boton_eliminar' method='post'";
            $rowData .= "<input type='hidden' name='id'";
            $rowData .= "value='{$ministerio->getId()}'>";
            $rowData .= "<button type='button'";
            $rowData .= "onclick='confirmarEliminar({$ministerio->getId()})' class='delete-button'";
            $rowData .= "style='font-size: 16px;'>Eliminar</button>";
            $rowData .= "</form>";
            $rowData .= "</td>";
            $rowData .= "</tr>";
        }
    }
}

```

```

    }

    return "<tbody>$rowData</tbody>";
}

public function actualizar(array $ministerios): void
{
    $title = 'Ministerios';
    $tbody = $this->renderizarTabla($ministerios);
    include '../app/vistas/Ministerio/index.php';
}

private function renderizarFormularioEdicion(Ministerio $ministerio):
string
{
    $formulario = '';

    if (!empty($ministerio) && $ministerio instanceof Ministerio) {
        $formulario .= "<form method='post'
action='/actualizar_ministerio'>";
        $formulario .= "<input type='hidden' name='id'
value='{ $ministerio->getId()}'>";
        $formulario .= "<div class='container-fluid'>";
        $formulario .= "<div class='row'>";
        $formulario .= "<div class='col-12 col-md-4'>";
        $formulario .= "<label for='nombre'>Nombres:</label>";
        $formulario .= "<input type='text' id='nombre' name='nombre'
value='{ $ministerio->getNombre()}' required>";
        $formulario .= "</div>";
        $formulario .= "<div class='col-12 col-md-4'>";
        $formulario .= "<label
for='descripcion'>Descripción:</label>";
        $formulario .= "<input type='text' id='descripcion'
name='descripcion' value='{ $ministerio->getDescripcion()}' required>";
        $formulario .= "</div>";
        $formulario .= "</div>";
        $formulario .= "<div class='row mt-3'>";
        $formulario .= "<div class='col-12 d-flex justify-content-
end'>";
        $formulario .= "<button class='add-button'
type='submit'>Actualizar</button>";
        $formulario .= "<a href='/ministerios' class='back-
button'>Volver</a>";
        $formulario .= "</div>";
        $formulario .= "</div>";
        $formulario .= "</form>";
    } else {
        $formulario .= "El Ministerio no se encontró o no existe.";
    }
}

```

```

        return $formulario;
    }

    public function mostrarFormularioEdicion(Ministerio $ministerio):
void
    {
        $title = 'Ministerio - Editar';
        $formulario = $this->renderizarFormularioEdicion($ministerio);
        include '../app/vistas/Ministerio/edit.php';
    }
}

<!DOCTYPE html>
<html>

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Iglesia App</title>
    <link rel="stylesheet"
href="https://fonts.googleapis.com/css?family=Lora&display=swap">
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.mi
n.css">
    <link rel="stylesheet" href="css/estilo.css">
</head>

<body>
    <header>
        <div class="container">
            <h1>Iglesia</h1>
        </div>
    </header>
    <nav class="sidebar">
        <a href="/menu">Inicio</a>
        <a href="/usuarios">Usuario</a>

        <a href="/cargos">Cargo</a>
        <a href="/ministerios">Ministerio</a>
        <a href="/eventos">Evento</a>
        <a href="/registro_evento">Registro Eventos</a>
        <a href="/tipo_relacion">Tipo Relacion</a>
        <a href="/parentescos">Parentesco</a>
    </nav>
    <main>
        <div class="container">
            <h2>Nuevo Ministerio</h2>

```



```

        </div>
        <div class="container">
            <div id='form_ministerio'>
                <form action="/ministerios" method="post">
                    <div class="row">
                        <div class="col-12 col-md-6">
                            <label for="nombre">Nombre</label>
                            <input name="nombre" type="text"
id="nombre" required>
                        </div>
                        <div class="col-12 col-md-6">
                            <label
for="descripcion">Descripción</label>
                            <input name="descripcion" type="text"
id="descripcion" required>
                        </div>
                    </div>
                    <div class="row mt-3">
                        <div class="col-12 d-flex justify-content-
end">
                            <button class="add-button"
type="submit">Agregar</button>
                        </div>
                    </div>
                </form>
            </div>

        </div>

        <h1><?= $title ?></h1>
        <table class="table">
            <thead>
                <tr>
                    <th>Id</th>
                    <th>Nombre</th> <!-- Cambiado a Nombre -->
                    <th>Descripción</th> <!-- Agregado campo de
descripcion -->
                </tr>
            </thead>
            <?= $tbody ?>
        </table>
    </div>
</main>

<footer>
    <div class="container text-center">
        <p>&copy; <?php echo date("Y"); ?> VeronicaAntezana</p>
    </div>
</footer>

```

```

    <script>
        function confirmarEliminar(id) {
            if (confirm('¿Estás seguro de que deseas eliminar este
ministerio?')) {
                // Si el usuario confirma, enviar el formulario de
eliminación
                document.querySelector('form[action="/eliminar_ministerio
"] input[name="id"]').value = id;
                document.querySelector('form[action="/eliminar_ministerio
"]').submit();
            }
        }
    </script>
</body>

</html>

```

```

<!DOCTYPE html>
<html>

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Iglesia App</title>
    <link rel="stylesheet"
href="https://fonts.googleapis.com/css?family=Lora&display=swap">
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.mi
n.css">
    <link rel="stylesheet" href="css/estilo.css">
</head>

<body>
    <header>
        <div class="container">
            <h1>Iglesia App</h1>
        </div>
    </header>

    <main>
        <div class="container">
            <h2><?= $title ?></h2>

            <!-- Muestra el formulario de edición aquí -->
            <?= $formulario ?>

            <!-- Más contenido HTML si es necesario -->

```

```

        </div>
    </main>
    <footer>
        <div class="container">
            <p>&copy; <?php echo date("Y"); ?> Veronica Antezana</p>
        </div>
    </footer>
    <script>
        function confirmarEliminar(id) {
            if (confirm('¿Estás seguro de que deseas eliminar este
Ministerio?')) {
                // Si el usuario confirma, enviar el formulario de
eliminación
                document.querySelector('form[action="/eliminar_ministerio
"] input[name="id"]').value = id;
                document.querySelector('form[action="/eliminar_ministerio
"]').submit();
            }
        }
    </script>
</body>
</html>

```

CU7: Registro de Evento
Controlador

```

<?php

declare(strict_types=1);

require_once(' ../app/modelos/RegistroEvento/MRegistroEvento.php');
require_once(' ../app/modelos/Evento/MEvento.php');
require_once(' ../app/modelos/DetalleEvento/MDetalleEvento.php');
require_once(' ../app/vistas/RegistroEvento/VRegistroEvento.php');

class CRegistroEvento
{
    private VRegistroEvento $vista;
    private MRegistroEvento $modelo;
    private MEvento $modeloEvento;
    private MDetalleEvento $modeloDetalleEvento;

    public function __construct()

```

```

{
    $this->vista = new VRegistroEvento();
    $this->modelo = new MRegistroEvento();
    $this->modeloEvento = new MEvento();
    $this->modeloDetalleEvento = new MDetalleEvento();
}

public function listarRegistroEventosC(): void
{
    $cargos = $this->modelo->listarRegistroEventos();
    $eventos = $this->modeloEvento->listarEventos();
    $this->vista->actualizar($cargos,$eventos);
}

    public function crearRegistroEventoC(string $lugar, string $nota,
string $fecha, int $evento_id): void
    {
        $this->modelo->crearRegistroEvento($lugar, $nota, $fecha ,
$evento_id);
        $cargos = $this->modelo->listarRegistroEventos();
        $ministerios = $this->modeloEvento->listarEventos();
        $this->vista->actualizar($cargos, $ministerios);
    }

public function eliminarRegistroEventoC(int $id): void
{
    $this->modelo->eliminarRegistroEvento($id);
    $cargos = $this->modelo->listarRegistroEventos();
    $ministerios = $this->modeloEvento->listarEventos();
    $this->vista->actualizar($cargos, $ministerios);
}

public function updateRegistroEventoC(int $id): void
{
    $cargos = $this->modelo->buscarRegistroEvento($id);
    $ministerios = $this->modeloEvento->listarEventos();
    $this->vista->mostrarFormularioEdicion($cargos,$ministerios);
}

    public function editarRegistroEventoC(int $id, string $lugar, string
$nota, string $fecha, int $evento_id): void
    {
        $this->modelo->editarRegistroEvento($id, $lugar, $nota, $fecha,
$evento_id);
        $cargos = $this->modelo->listarRegistroEventos();
        $ministerios = $this->modeloEvento->listarEventos();
        $this->vista->actualizar($cargos, $ministerios);
    }

```

```
}  
}
```

Modelo

```
<?php  
  
require_once('../app/modelos/iglesiaDB.php');  
require_once('../app/modelos/RegistroEvento/RegistroEvento.php');  
  
class MRegistroEvento  
{  
    private IglesiaDB $database;  
  
    public function __construct()  
    {  
        $this->database = new IglesiaDB();  
    }  
  
    public function crearRegistroEvento( $lugar,$nota,$fecha,$evento_id):  
void //crear  
    {  
        $bd = $this->database->getConnection();  
        try {  
            $query = "INSERT INTO " . $this->  
database::TABLE_REGISTRO_EVENTO . " (lugar, nota, fecha, evento_id)  
VALUES (?, ?, ?,?)";  
            $stmt = $bd->prepare($query);  
            $stmt->bind_param("sssi", $lugar, $nota, $fecha, $evento_id);  
            if ($stmt->execute()) {  
                error_log("Registro de Evento insertado con éxito");  
            }  
        } catch (Exception $e) {  
            error_log("Excepción al insertar el Registro de Evento en la  
base de datos: " . $e->getMessage());  
        } finally {  
            if (isset($stmt)) {  
                $stmt->close();  
            }  
            $bd->close();  
        }  
    }  
  
    public function listarRegistroEventos(): array //listar  
    {  
        $bd = $this->database->getConnection();
```

```

        $registro_eventos = [];

        try {
            $result = $bd->query('SELECT * FROM ' . $this->database::TABLE_REGISTRO_EVENTO);

            if ($result) {
                while ($row = $result->fetch_assoc()) {

                    $registro_evento = new RegistroEvento($row['id'],
$row['lugar'], $row['nota'], $row['fecha'], $row['evento_id']);
                    $registro_eventos[] = $registro_evento;
                }
            }
        } catch (Exception $e) {
            error_log("Excepción en mostrar Registros de Eventos: " . $e->getMessage());
        } finally {
            $bd->close();
        }

        return $registro_eventos;
    }

    public function buscarRegistroEvento(int $id): RegistroEvento
    {
        $bd = $this->database->getConnection();

        try {
            $query = "SELECT * FROM " . $this->database::TABLE_REGISTRO_EVENTO . " WHERE id = ?";
            $stmt = $bd->prepare($query);
            $stmt->bind_param("i", $id);
            $stmt->execute();
            $result = $stmt->get_result();

            if ($result && $result->num_rows > 0) {
                $row = $result->fetch_assoc();
                // Crear y devolver un objeto Cargo a partir de los datos
de la fila
                $carga = new RegistroEvento($row['id'], $row['lugar'],
$row['nota'], $row['fecha'], $row['evento_id']);
                return $carga;
            }
        } catch (Exception $e) {
            error_log("Excepción en buscarRegistro: " . $e->getMessage());
        } finally {
            $bd->close();
        }
    }

```

```

    }

    return null;
}

public function editarRegistroEvento(int $id, string $lugar, string
$nota, string $fecha, $evento_id): void
{
    $bd = $this->database->getConnection();

    try {
        $query = "UPDATE " . $this->database::TABLE_REGISTRO_EVENTO .
" SET lugar = ?, nota = ?, fecha = ?, evento_id = ? WHERE id = ?";
        $stmt = $bd->prepare($query);
        $stmt->bind_param("sssi", $lugar, $nota, $fecha, $evento_id
,$id);

        if ($stmt->execute()) {
            error_log("Registro editado con éxito");
        }
    } catch (Exception $e) {
        error_log("Excepción al editar el registro: " . $e-
>getMessage());
    } finally {
        if (isset($stmt)) {
            $stmt->close();
        }
        $bd->close();
    }
}

public function eliminarRegistroEvento(int $id): void
{
    $bd = $this->database->getConnection();

    try {
        $query = "DELETE FROM " . $this-
>database::TABLE_REGISTRO_EVENTO . " WHERE id = ?";
        $stmt = $bd->prepare($query);
        $stmt->bind_param("i", $id);

        if ($stmt->execute()) {
            error_log("Registro de Evento eliminado con éxito");
        }
    } catch (Exception $e) {
        error_log("Excepción al eliminar el registro: " . $e-
>getMessage());
    } finally {

```

```

        if (isset($stmt)) {
            $stmt->close();
        }
        $bd->close();
    }
}
}

```

```

<?php

class RegistroEvento
{
    private int $id;
    private string $lugar;
    private string $nota;
    private string $fecha;
    private $evento_id;

    public function __construct(int $id, string $lugar, string $nota,
string $fecha, $evento_id )
    {
        $this->id = $id;
        $this->lugar = $lugar;
        $this->nota = $nota;
        $this->fecha = $fecha;
        $this->evento_id= $evento_id;
    }

    public function getId(): int
    {
        return $this->id;
    }

    public function getLugar(): string
    {
        return $this->lugar;
    }

    public function getNota(): string
    {
        return $this->nota;
    }
    public function getFecha(): string
    {
        return $this->fecha;
    }
}

```



```

public function setId(int $id): void
{
    $this->id = $id;
}

public function setLugar(string $lugar): void
{
    $this->lugar = $lugar;
}

public function setNota(string $nota): void
{
    $this->nota = $nota;
}

public function setFecha(string $fecha): void
{
    $this->fecha = $fecha;
}

public function getEventoId()
{
    return $this->evento_id;
}

public function setEventoId($evento_id)
{
    $this->evento_id = $evento_id;
}

public function __toString(): string
{
    return "RegistroEvento{" .
        "id=" . $this->id .
        ", lugar='" . $this->lugar . '\'' .
        ", nota='" . $this->nota . '\'' .
        ", fecha='" . $this->fecha . '\'' .
        ", evento_id=" . $this->evento_id .
        '\'';
}
}

```

Vista

```

<?php

declare(strict_types=1);

class VRegistroEvento

```

```

{
    private function renderizarTabla($registroeventos,array $eventos):
string
    {
        $rowData = '';

        if (empty($registroeventos)) {
            return "<tbody></tbody>";
        }

        foreach ((array)$registroeventos as $registroevento) {
            $rowData .= "<tr>";
            $rowData .= "<td>{$registroevento->getId()}</td>";
            $rowData .= "<td>{$registroevento->getLugar()}</td>";
            $rowData .= "<td>{$registroevento->getNota()}</td>";
            $rowData .= "<td>{$registroevento->getFecha()}</td>";

            // Busca el nombre del cargo basado en el ID del cargo
            $nombreEvento = "";
            foreach ($eventos as $evento) {
                if ($evento->getId() == $registroevento->getEventoId()) {
                    $rowData .= "<td>{$evento->getNombre()}</td>";
                    break;
                }
            }

            $rowData .= "<td><a
href='/detalle_evento?id={$registroevento->getId()}'
class='agregarUsuario-button'>Agregar Usuario</a></td>";
            $rowData .= "<td><a
href='/detalle_evento?id={$registroevento->getId()}' class='verUsuario-
button'>Ver Usuarios</a></td>";
            $rowData .= "<td><a
href='/editar_registro_evento?id={$registroevento->getId()}' class='edit-
button'>Editar</a></td>";
            $rowData .= "<td>";
            $rowData .= "<form id='form_boton_eliminar' method='post'
action='/eliminar_registro_evento'>";
            $rowData .= "<input type='hidden' name='id'
value='{ $registroevento->getId()}'>";
            $rowData .= "<button type='button'
onclick='confirmarEliminar({ $registroevento->getId()})' class='delete-
button' style='font-size: 16px;'>Eliminar</button>";

            $rowData .= "</form>";
            $rowData .= "</td>";
        }
    }
}

```

```

        $rowData .= "</tr>";
    }

    return "<tbody>$rowData</tbody>";
}

public function renderizarFormularioCrear($eventos): string
{
    $nuevoCargoForm = '
    <div class="container">
        <h2>Nuevo Registro Evento </h2>
    </div>
    <div class="container">

        <div id="form_cargo">
            <form action="/registro_evento" method="post">
                <div class="row">
                    <div class="col-12 col-md-4">
                        <label for="lugar">Lugar</label>
                        <input name="lugar" type="text" id="lugar"
required>

                    </div>
                    <div class="col-12 col-md-4">
                        <label for="nota">Nota</label>
                        <input name="nota" type="text" id="nota"
required>

                    </div>
                    <div class="col-12 col-md-4">
                        <label for="fecha">Fecha</label>
                        <input name="fecha" type="date" id="fecha" required>
                    </div>

                    <div class="col-12 col-md-4">
                        <label for="evento">Evento</label>
                        <select name="evento" id="evento" >;

                            foreach ($eventos as $evento) {
                                $nuevoCargoForm .= "<option value='{ $evento->getId()}'>{ $evento->getNombre()}</option>";
                            }

                        $nuevoCargoForm .= '
                        </select>
                    </div>
                </div>
                <br>
                <div class="row mt-3">
                    <div class="col-12 d-flex justify-content-end">

```

```

        <button class="add-button"
type="submit">Agregar</button>
    </div>
</div>

</div>
</form>
</div>
</div>
';

    return $nuevoCargoForm;
}

public function actualizar($registroeventos, $eventos): void
{
    $title = 'Registro Eventos';
    $tbody = $this->renderizarTabla($registroeventos, $eventos);
    $formulario = $this->renderizarFormularioCrear($eventos);
    include '../app/vistas/RegistroEvento/index.php';
}

private function renderizarFormularioEdicion($registroevento,
$eventos): string
{
    $formulario = '';

    if (!empty($registroevento)) {
        $formulario .= "<form method='post'
action='/actualizar_registro_evento'>";
        $formulario .= "<input type='hidden' name='id'
value='{ $registroevento->getId()}'>";
        $formulario .= "<div class='container-fluid'>";
        $formulario .= "<div class='row'>";
        $formulario .= "<div class='col-12 col-md-4'>";
        $formulario .= "<label for='lugar'>Lugar:</label>";
        $formulario .= "<input type='text' id='lugar' name='lugar'
value='{ $registroevento->getLugar()}' required>";
        $formulario .= "</div>";
        $formulario .= "<div class='col-12 col-md-4'>";
        $formulario .= "<label for='nota'>Nota:</label>";
        $formulario .= "<input type='text' id='nota' name='nota'
value='{ $registroevento->getNota()}' required>";
        $formulario .= "</div>";
        $formulario .= "<div class='col-12 col-md-4'>";
        $formulario .= "<label for='fecha'>Fecha:</label>";
    }
}

```

```

        $formulario .= "<input type='date' id='fecha' name='fecha'
value='{ $registroevento->getFecha()}' required>";
        $formulario .= "</div>";
        $formulario .= "</div>";
        $formulario .= "<div class='col-12 col-md-4'>";
        $formulario .= "<label for='evento'>Evento:</label>";
        $formulario .= "<select name='evento' id='evento' required>";

        foreach ($eventos as $evento) {
            $selected = $evento->getId() === $registroevento-
>getEventoId() ? 'selected' : '';
            $formulario .= "<option value='{ $evento->getId()}'
{$selected}>{$evento->getNombre()}</option>";
        }

        $formulario .= "</select>";
        $formulario .= "</div>";
        $formulario .= "</div>";
        $formulario .= "<br>";
        $formulario .= "<div class='row mt-3'>";
        $formulario .= "<div class='col-12 d-flex justify-content-
end'>";

        $formulario .= "<button class='add-button'
type='submit'>Actualizar</button>";
        $formulario .= "<a href='/menu' class='back-
button'>Volver</a>";
        $formulario .= "</div>";
        $formulario .= "</div>";
        $formulario .= "</form>";
    } else {
        $formulario .= "El cargo no se encontró o no existe.";
    }

    return $formulario;
}

public function mostrarFormularioEdicion($registroevento, $eventos):
void
{
    $title = 'Registro - Editar';
    $formulario = $this->renderizarFormularioEdicion($registroevento,
$eventos);
    include '../app/vistas/RegistroEvento/edit.php';
}
}

```

```
<!DOCTYPE html>
```

```

<html>

<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <title>Iglesia App</title>
  <link rel="stylesheet"
href="https://fonts.googleapis.com/css?family=Lora&display=swap">
  <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.min.css">
  <link rel="stylesheet" href="css/estilo.css">
</head>
<nav class="sidebar">
  <a href="/menu">Inicio</a>
  <a href="/usuarios">Usuario</a>
  <a href="/cargos">Cargo</a>
  <a href="/ministerios">Ministerio</a>
  <a href="/eventos">Evento</a>
  <a href="/registro_evento">Registro Eventos</a>
  <a href="/tipo_relacion">Tipo Relacion</a>
  <a href="/parentescos">Parentesco</a>
</nav>
<body>
  <header>
    <div class="container">
      <h1>Iglesia App</h1>
    </div>
  </header>
  <nav>
    <div class="container">

    </div>
  </nav>
  <main>
    <?= $formulario ?>
    <h1><?= $title ?></h1>
    <table>
      <thead>
        <tr>
          <th>Id</th>
          <th>Lugar</th>
          <th>Nota</th>
          <th>Fecha</th>
          <th>Evento</th>
        </tr>
      </thead>
      <?= $tbody ?>
    </table>

```

```

        </div>
    </main>
    <footer>
        <div class="container">
            <p>&copy; <?php echo date("Y"); ?> VeronicaAntezana</p>
        </div>
    </footer>
    <script>
        function confirmarEliminar(id) {
            if (confirm('¿Estás seguro de que deseas eliminar este
cargo?')) {
                // Si el usuario confirma, enviar el formulario de
eliminación
                document.querySelector('form[action="/eliminar_registro_e
vento"] input[name="id"]').value = id;
                document.querySelector('form[action="/eliminar_registro_e
vento"]').submit();
            }
        }
    </script>
</body>
</html>

```

```

<!DOCTYPE html>
<html>

<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <title>Iglesia App</title>
    <link rel="stylesheet"
href="https://fonts.googleapis.com/css?family=Lora&display=swap">
    <link rel="stylesheet"
href="https://stackpath.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.mi
n.css">
    <link rel="stylesheet" href="css/estilo.css">
</head>

<body>
    <header>
        <div class="container">
            <h1>Iglesia App</h1>
        </div>
    </header>

    <main>
        <div class="container">

```

```

        <h2><?= $title ?></h2>

        <!-- Muestra el formulario de edición aquí -->
        <?= $formulario ?>

        <!-- Más contenido HTML si es necesario -->
    </div>
</main>
<footer>
    <div class="container">
        <p>&copy; <?php echo date("Y"); ?> Veronica Antezana</p>
    </div>
</footer>
<script>
    function confirmarEliminar(id) {
        if (confirm('¿Estás seguro de que deseas eliminar este Tipo
de Relacion?')) {
            // Si el usuario confirma, enviar el formulario de
eliminación
            document.querySelector('form[action="/eliminar_registro_e
vento"] input[name="id"]').value = id;
            document.querySelector('form[action="/eliminar_registro_e
vento"]').submit();
        }
    }
</script>
</body>
</html>

```