

# Proyecto Final Programado

---

Nombre del Proyecto:

Página web para escuela privada (Avance no.4 - Final)

Nombre del Estudiante/Desarrollador:

Anthony Ariel Chaves Mora

Profesor/Curso:

Estefanía Boza Villalobos / Programación Avanzada

Fecha:

10/12/2025

# Sumario

Objetivo General .....	3
Objetivos Específicos.....	3
Justificación del Proyecto .....	4
Alcances Esperados.....	4
Requerimientos del Proyecto .....	4
Objetivo General .....	5
Objetivos Específicos.....	5
<i>Justificación del Proyecto</i> .....	6
Alcances Esperados.....	6
Requerimientos del Proyecto .....	6
<i>Diseño del Sistema (Pre-diseño visual)</i> .....	7
Elementos de Diseño .....	7
<i>Explicación de la estructura</i> .....	8
Estructura.....	9
<i>Definición de la base de datos</i> .....	13
<i>Funciones y Triggers</i> .....	15
<i>Esquemas y Diagramas</i> .....	17
Normalización: .....	17
Conexión con la aplicación .....	18
4. Backend.....	19
1. Arquitectura del Backend.....	19
Justificación.....	19
Capas utilizadas .....	19
2. Componentes y estructura del código .....	20
Componentes .....	20
Diagrama general de flujo .....	20
Seguridad.....	21
Gestión de errores .....	21
Servicios externos.....	21
Pruebas.....	21
Escalabilidad.....	21

## **Información General del Proyecto**

El proyecto busca cumplir con las necesidades y requerimientos del sistema solicitado por el cliente, en este caso el centro educativo CoopeCEP R. L., el cual requiere una página web básica con funcionalidades como el mostrar fotografías reales del centro educativo, mostrar información de contacto, así como un formulario para que puedan contactar con el centro en caso de que deseen matricular, y finalmente que exista un inicio de sesión por parte de los profesores para el mantenimiento de su perfil y gestión de estudiantes.

## **Objetivo General**

Diseñar y desarrollar un sistema web para el Centro Educativo CoopeCEP R.L. el cual permita la visualización de la información institucional y de contacto para el público en general; la gestión de acceso seguro para los perfiles de los profesores; y la recepción de solicitudes de matrícula o información a través de un formulario en línea.

## **Objetivos Específicos**

- Diseñar la arquitectura del sistema web que permita una interacción eficiente entre los distintos tipos de usuarios (público en general, profesores, y administradores) y la base de datos del centro educativo.
- Desarrollar una interfaz gráfica intuitiva y atractiva que mejore la experiencia de todos los usuarios del sitio web.
- Implementar un backend escalable y modular, que facilite la gestión de las funcionalidades específicas para cada tipo de usuario y que permita futuras ampliaciones al sistema.
- Crear una base de datos relacional segura y consistente, que garantice la integridad, confidencialidad y disponibilidad de la información almacenada.

## **Justificación del Proyecto**

El proyecto surge como con la necesidad de brindar al centro educativo privado de CoopeCEP un sitio web con el fin de mostrar información importante que pueda asesorar a clientes de los servicios que brindan, encaminados a llegar a un nivel mayor de público.

## **Alcances Esperados**

- El sistema permitirá registrar profesores.
- Permitirá a visitantes del sitio web ponerse en contacto con la escuela mediante un formulario.
- Acceso a la información general de la escuela en tiempo real.
- Integración con módulos futuros según necesidades y ajustes del centro educativo.
- Mostrar los diversos servicios y características que ofrece el centro educativo.

## **Requerimientos del Proyecto**

- Requerimientos funcionales: Registro de profesores, gestión de la información del colegio, reportes, notificaciones a los encargados de gestionar solicitudes de información de nuevos clientes.
- Requerimientos no funcionales: Seguridad, rendimiento, escalabilidad, usabilidad, UI/UX limpio y agradable.
- Infraestructura mínima: backend compuesto por Python, Node.js, y un frontend compuesto por HTML y CSS; servidor con tabla de Profesores, base de datos MySQL; división de accesos definida para el público en general, profesores y administrador.

## ***Información General del Proyecto***

El proyecto busca cumplir con las necesidades y requerimientos del sistema solicitado por el cliente, en este caso el centro educativo CoopeCEP R. L., el cual requiere una página web básica con funcionalidades como el mostrar fotografías reales del centro educativo, mostrar información de contacto, así como un formulario para que puedan contactar con el centro en caso de que deseen matricular, y finalmente que exista un inicio de sesión por parte de los profesores para el mantenimiento de su perfil y gestión de estudiantes.

## **Objetivo General**

Diseñar y desarrollar un sistema web para el Centro Educativo CoopeCEP R.L. el cual permita la visualización de la información institucional y de contacto para el público en general; la gestión de acceso seguro para los perfiles de los profesores; y la recepción de solicitudes de matrícula o información a través de un formulario en línea.

## **Objetivos Específicos**

- Diseñar la arquitectura del sistema web que permita una interacción eficiente entre los distintos tipos de usuarios (público en general, profesores, y administradores) y la base de datos del centro educativo.
- Desarrollar una interfaz gráfica intuitiva y atractiva que mejore la experiencia de todos los usuarios del sitio web.
- Implementar un backend escalable y modular, que facilite la gestión de las funcionalidades específicas para cada tipo de usuario y que permita futuras ampliaciones al sistema.
- Crear una base de datos relacional segura y consistente, que garantice la integridad, confidencialidad y disponibilidad de la información almacenada.

## ***Justificación del Proyecto***

El proyecto surge como con la necesidad de brindar al centro educativo privado de CoopeCEP un sitio web con el fin de mostrar información importante que pueda asesorar a clientes de los servicios que brindan, encaminados a llegar a un nivel mayor de público.

### **Alcances Esperados**

- El sistema permitirá registrar profesores.
- Permitirá a visitantes del sitio web ponerse en contacto con la escuela mediante un formulario.
- Acceso a la información general de la escuela en tiempo real.
- Integración con módulos futuros según necesidades y ajustes del centro educativo.
- Mostrar los diversos servicios y características que ofrece el centro educativo.

### **Requerimientos del Proyecto**

- Requerimientos funcionales: Registro de profesores, gestión de la información del colegio, reportes, notificaciones a los encargados de gestionar solicitudes de información de nuevos clientes.
- Requerimientos no funcionales: Seguridad, rendimiento, escalabilidad, usabilidad, UI/UX limpio y agradable.
- Infraestructura mínima: backend compuesto por Python, Node.js, y un frontend compuesto por HTML y CSS; servidor con tabla de Profesores, base de datos MySQL; división de accesos definida para el público en general, profesores y administrador.

## ***Diseño del Sistema (Pre-diseño visual)***

A continuación se presenta el pre-diseño del sitio web como propuesta inicial del aspecto del sistema.

### **Elementos de Diseño**

- **Colores:** se utilizará el azul característico de las letras del logo del centro educativo, en combinación con el blanco y demás tonos de azul más oscuros:

Fondo de páginas: blue (2b78e4), white (ffffff)

Botón de WhatsApp: green 00ed3b

Botones para enviar formulario: blue (2b78e4)

- **Tipografía:** utilizaremos una fuente legible y moderna que al mismo tiempo refleje un estilo juvenil ambientado en la población estudiantil:

Letra: Roboto o Segoe UI (se define en el progreso).


- **Imágenes:** Pueden ir cambiando conforme se consolida el proyecto, para efectos de este prototipo se utilizaron imágenes generales.


Algunos de los ejemplos de las imágenes a utilizar serán las siguientes:



- **Íconos y Logo:** el botón de contacto rápido por WhatsApp tendrá su ícono. Además en el footer o pie de la página de cada sección del sitio web habrán links para la página de Facebook y YouTube del centro educativo con su respectivo ícono cada link.

WhatsApp:  

Facebook: 

YouTube: 

Logo:



**Slogan:** tanto la página principal como los footers tendrán el slogan de la institución:

“Educación para el éxito”.

## ***Explicación de la estructura***

1. “Home o Inicio”: con elementos como botones para cada sección del sitio web y el logo del centro educativo, contendrá información relevante e importante del centro como los diversos servicios que posee, dónde se encuentra ubicado, imágenes reales del centro educativo. En el final de la página tendrá un formulario para contactar al centro. Las páginas posteriores a Home tendrán un Link Bar o barra de direcciones para desplazarse a las otras páginas en lugar de los botones que hay en Home que cumple como este Link Bar.

Dirección URL: <https://www.coopecep.com/>

2. “Ponte en contacto con nosotros”: será la página donde habrá exclusivamente un formulario para contactar al centro educativo, similar al formulario de “Home”. Todos los formularios tendrán un botón para escribir por WhatsApp también como medio alternativo de comunicación. Este botón de WhatsApp redirige al programa hacia el chat del número de la institución.

Dirección URL: <https://www.coopecep.com/contacto/>

3. “Acerca de nosotros”: aquí habrá información relevante del centro educativo, cómo imágenes, texto explicativo, ya que es una página meramente informativa. Al final habrá un formulario para contactar al centro educativo. Tendrá una sección donde se puede observar en el mapa de Google la ubicación del centro educativo.

Dirección URL: <https://www.coopecep.com/about/>

4. “Programas”: contendrá los diversos niveles de educación que ofrece (materno, preescolar, primaria y secundaria), con direcciones para cada nivel el cual tendrá su propia ventana informativa. Tanto en Programas como en cada página de cada nivel, al final habrá un formulario de contacto.

Dirección URL: <https://www.coopecep.com/programas/>

5. “Inicio de sesión en Plataforma virtual”: se presenta una pantalla de Login o Inicio de sesión, .

Dirección URL: <https://www.coopecep.com/login/>

6. “Gestor para profesores”: una vez iniciada la sesión por parte de los profesores en la plataforma una vez registrados, permitirá crear listas de estudiantes y asignar datos a estos estudiantes para llevar un control de las notas y demás.

Dirección URL: <https://www.coopecep.com/profesores/>

7. “Footer o Pie de página”: es el estilo que le daremos al footer, el cual contendrá enlaces rápidos, el logo del centro educativo, la dirección de este, y enlaces a sus redes sociales, así como los números para contactar al centro. Se encontrará en todas las pestañas excepto en Login y dentro de Plataforma Virtual.



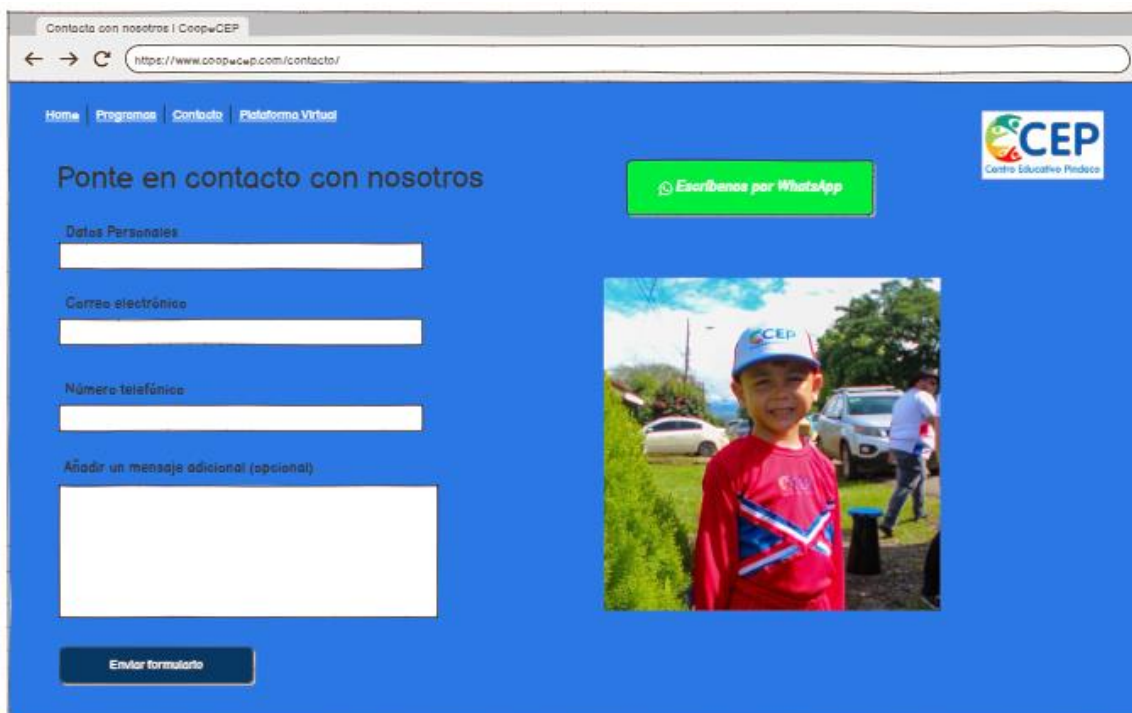
## Estructura

El pre-diseño fue diseñado mediante Balsamiq Mockups, herramienta en línea la cual permitió maquetar cada ventana del sitio web, sirviendo como una guía visual que ayudará a los usuarios y al equipo de desarrollo a comprender cómo se verá el programa antes de su implementación.

### 1. “Home o Inicio”:



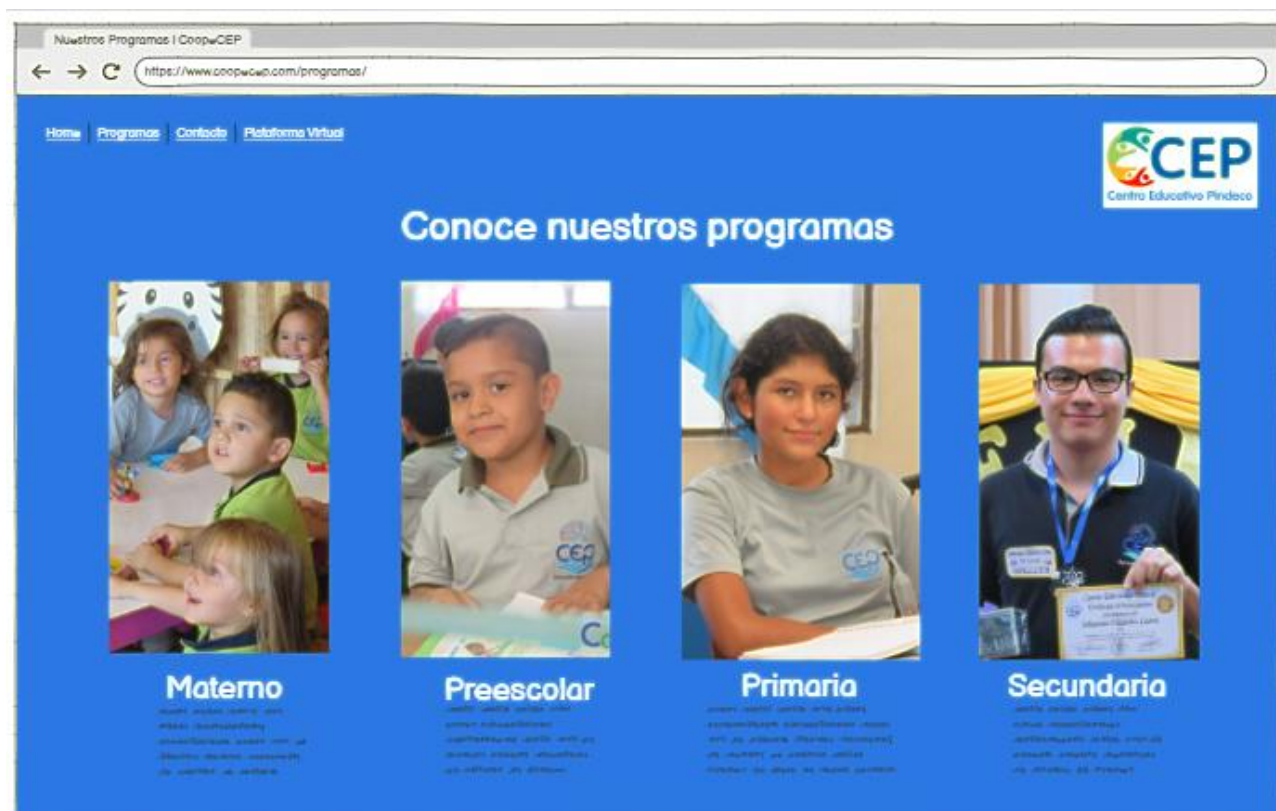
### 2. “Ponte en contacto con nosotros”:



### 3. “Acerca de nosotros”:



### 4. “Programas”:




### 5. “Login o Inicio de sesión”:

Plataforma Virtual | CoopwCEP

← → ↺ <https://www.coopwcep.com/login/>

[Home](#) | [Programas](#) | [Contacto](#) | [Plataforma Virtual](#)



**CEP**  
Centro Educativo Pindoco

## Iniciar Sesión

**Usuario**

**Contraseña**

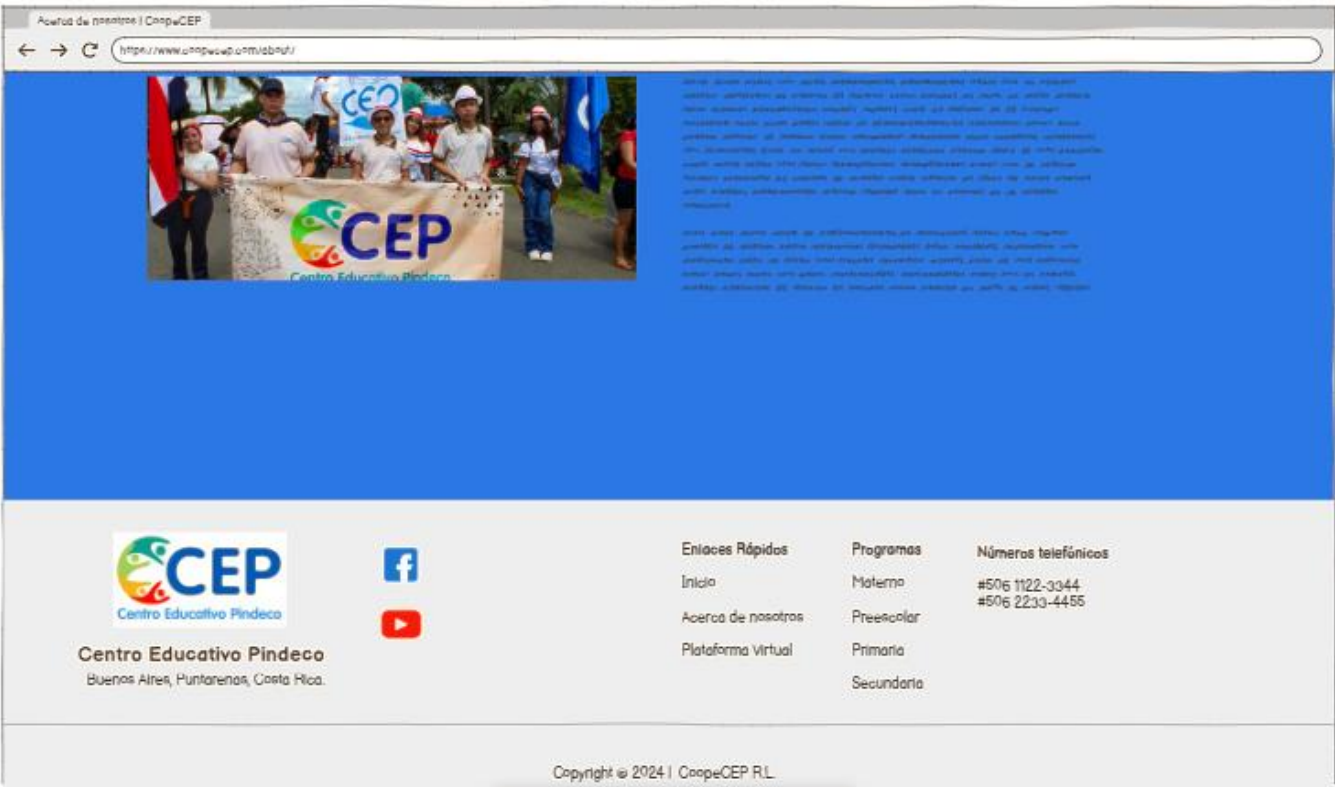
[¿Has olvidado tu contraseña?](#)

**Iniciar Sesión**

6. “Gestor para profesores ( Plataforma virtual )” :

[illegible]

7. “Footer” :





# Base de Datos

## Definición de la base de datos

La base de datos será diseñada para garantizar la integridad y consistencia de los datos del centro educativo, la cual será relacional. El propósito de la base de datos del sistema es almacenar información relacionada con el formulario de contacto, almacenamiento de los usuarios de la plataforma virtual, registrar los roles tanto Administrador como Profesor y almacenar los datos de los estudiantes para uso de los profesores.

Motor de base de datos: utilizaremos el motor de Base de datos SQL Server Management Studio 21.


### Elementos de diseño:


Los datos agrupados en formato de tablas en SQL con sus respectivos tipos de datos en columnas, así como las llaves primarias y foráneas, y campos nulos vienen a ser los siguientes:

#### **Tabla:** contacto\_form (Para uso del formulario)

- *Llaves primarias:* id\_contacto (INT, auto-incremental)

- *Nulos:* fecha\_envio acepta nulos ya que la fecha se registra automáticamente; mensaje acepta nulos ya que el mensaje adicional es opcional en el formulario.

	Column Name	Data Type	Allow Nulls
	id_contacto	int	<input type="checkbox"/>
	nombre	varchar(100)	<input type="checkbox"/>
	email	varchar(200)	<input type="checkbox"/>
	telefono	varchar(30)	<input type="checkbox"/>
	mensaje	varchar(MAX)	<input checked="" type="checkbox"/>
	fecha_envio	datetime	<input checked="" type="checkbox"/>
			<input type="checkbox"/>

	Column Name	Data Type	Allow Nulls
	id_usuario	int	<input type="checkbox"/>
	usuario	varchar(50)	<input type="checkbox"/>
	clave_hash	varbinary(MAX)	<input type="checkbox"/>
	id_rol	int	<input type="checkbox"/>
	fecha_creacion	datetime	<input checked="" type="checkbox"/>

#### **Tabla:** usuarios (Funciona con el Login)

Se utiliza para registrar los usuarios que accesan a la Plataforma Virtual.

- *Llaves primarias:* id\_usuario (INT, autoincrementable)


- *Llaves foráneas:* id\_rol (INT) la cual referencia a id\_rol (INT) de tabla roles.

**Tabla: roles (Administrador o Profesor)**

Funciona para mantener niveles y control de los accesos a la plataforma virtual

- *Llaves primarias*: id\_rol (INT, autoincrementable).

- *Laves foráneas*: id\_rol (INT) es referenciado por la llave foránea id\_rol (INT) de la tabla usuarios.

	Column Name	Data Type	Allow Nulls
	id_rol	int	<input type="checkbox"/>
	nombre_rol	varchar(50)	<input type="checkbox"/>

**Tabla: estudiantes**

Para el registro de los estudiantes por parte de los profesores dentro de la Plataforma Virtual.

- *Llaves primarias*:


id\_estudiante (INT, auto-incrementable).


Nota: No se refiere a la cédula del estudiante sino al número que se le asigna en la lista.

- *Nulos*:

fecha\_registro (DATETIME)

Debido a que se registra automáticamente, no debe de ser asignada por el profesor, por lo que se asigna permisible a nulos.

	Column Name	Data Type	Allow Nulls
	id_estudiante	int	<input type="checkbox"/>
	nombre_completo	varchar(100)	<input type="checkbox"/>
	edad	int	<input type="checkbox"/>
	direccion	varchar(200)	<input type="checkbox"/>
	aprobado	bit	<input type="checkbox"/>
	contacto_padre	varchar(50)	<input type="checkbox"/>
	fecha_registro	datetime	<input checked="" type="checkbox"/>

	Column Name	Data Type	Allow Nulls
	id_log	int	<input type="checkbox"/>
	id_estudiante	int	<input checked="" type="checkbox"/>
	nombre	varchar(100)	<input checked="" type="checkbox"/>
	fecha_eliminado	datetime	<input checked="" type="checkbox"/>

**Tabla: log\_estudiantes\_eliminados**

Utilizada por el Trigger el cual será explicado más adelante.

- *Campos nulos*:

fecha\_registro (DATETIME)

La razón es debido a que se registra automáticamente, por lo que no debe de ser asignada por el profesor, por lo que se asigna permisible a nulos.

## Funciones y Triggers

Se implementa una Function la cual automatiza el proceso de validación de usuario y clave correctos y además un Trigger el cual registra de manera automática los estudiantes eliminados en caso de seleccionar dicha opción accidentalmente en la Plataforma Virtual.

### - **Trigger:**

Registro de estudiantes eliminados.

Se crea primero la tabla de registro donde se almacenará un registro de los estudiantes eliminados, que guardará un id\_log para llevar un orden, y los datos solo necesarios a rescatar como el id del estudiante, nombre, y la fecha de eliminación (automática, documentada por la máquina).

```
-- TRIGGER: Registro de esudiantes eliminados

USE CoopeCEP_Website;
GO

-- Tabla de registro
CREATE TABLE log_estudiantes_eliminados (
  id_log INT IDENTITY(1,1) PRIMARY KEY,
  id_estudiante INT,
  nombre VARCHAR(100),
  fecha_eliminado DATETIME DEFAULT GETDATE()
);
GO

-- Creación del TRIGGER
CREATE TRIGGER trigger_estudiantes_delete
ON estudiantes
FOR DELETE
AS
BEGIN
  INSERT INTO log_estudiantes_eliminados (id_estudiante, nombre)
  SELECT id_estudiante, nombre_completo
  FROM deleted;
END;
GO
```

### - **Function:**

Check-in de credenciales de LOGIN

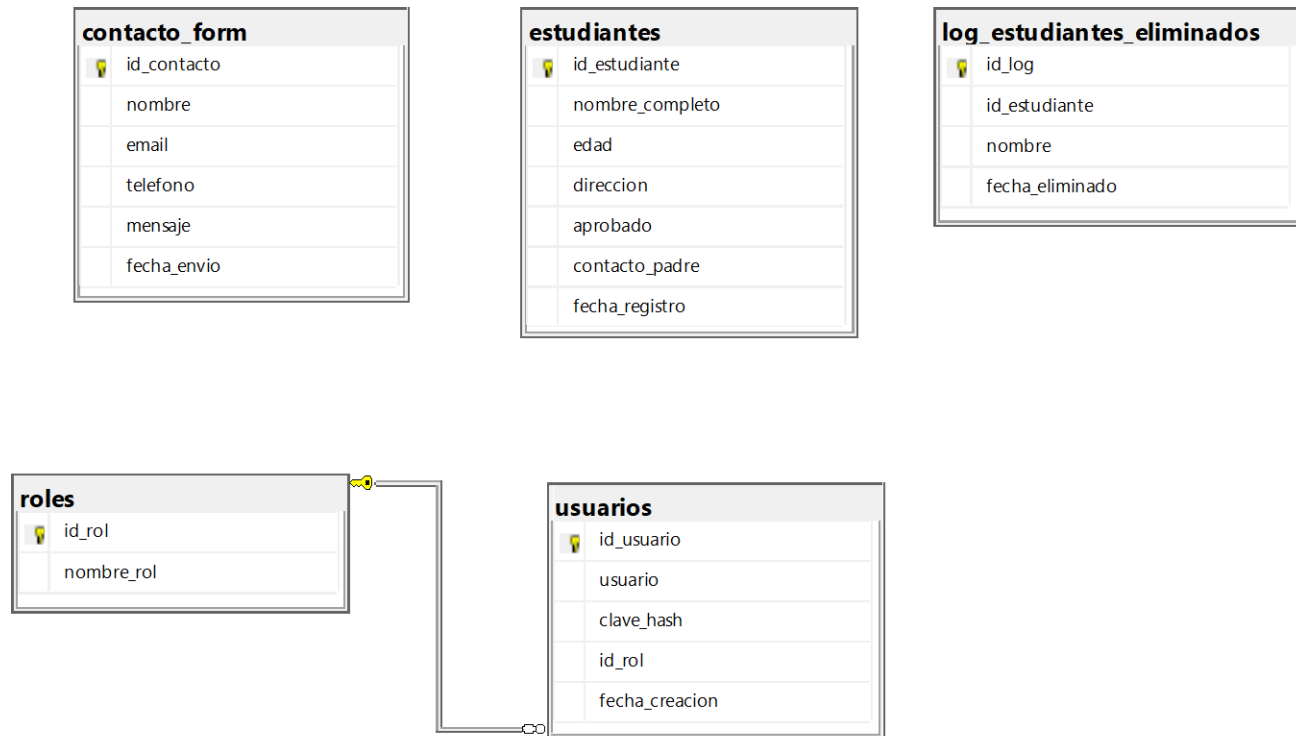
Valida si el usuario y clave son correctos, devolviendo 1 = Login válido, 0 = Login inválido. Luego realiza una comparación con la tabla y si existe el usuario, cambia el @resultado a 1.

```
-----  
-- FUNCTION: Check-in de credenciales de LOGIN  
-----  
  
USE CoopeCEP_Website;  
GO  
  
-- Creación de la FUNCTION  
-- Valida si el usuario y clave son correctos,  
-- devolviendo 1 = login válido, 0 = login inválido  
CREATE FUNCTION function_validar_login  
(  
    @usuario VARCHAR(50),  
    @clave VARCHAR(200)  
)  
RETURNS BIT  
AS  
BEGIN  
    DECLARE @resultado BIT = 0;  
    DECLARE @hash VARBINARY(MAX) = HASHBYTES('SHA2_256', @clave);  
  
    -- Comparación con la tabla y si existe el usuario, cambia el @resultado a 1  
    IF EXISTS (  
        SELECT 1 FROM usuarios  
        WHERE usuario = @usuario AND clave_hash = @hash  
    )  
        SET @resultado = 1;  
  
    RETURN @resultado;  
END;  
GO
```



## Esquemas y Diagramas

En este caso se utiliza SQL Server Management Studio v.21 para la creación y estructuración del diagrama de entidad-relación.



### ***La ausencia de llaves foráneas en la tabla de log\_estudiantes\_eliminados:***

Las tablas de LOG se consideran “historical audit tables”, y no llevan llaves foráneas. Cuando un estudiante es eliminado de la tabla Estudiantes, su fila deja de existir, y SQL Server no permite borrar datos con registros dependientes.

La columna **id\_rol** (INT) de la tabla **roles** es referenciada por la llave foránea **id\_rol** (INT) de la tabla **usuarios**.

### ● ***Normalización:***

La base de datos cumple con la Tercera Forma Natural (3FN), ya que no existen datos repetidos, no existen dependencias parciales entre columnas ni tablas, no hay dependencias transitivas, y cada dato depende solo de su respectivo ID, es decir, nada depende de un dato que no sea la clave primaria.

- ## Conexión con la aplicación

PHP normalmente usa MySQL, pero en este caso se prefirió utilizar Microsoft SQL Server, por lo que se necesita un controlador especial: Microsoft Drivers for PHP for SQL Server.

**Extensiones:** sqlsrv y pdo\_sqlsrv

Son desarrolladas por Microsoft, por lo que vendrían a ser las únicas oficiales, compatibles y seguras. Dichas extensiones permiten: conectar PHP a SQL Server; ejecutar consultas; usar procedimientos almacenados y manejar errores de SQL Server.

**Instalaciones necesarias:**

php\_sqlsrv.dll  
php\_pdo\_sqlsrv.dll

**Habilitar en php.ini (al utilizar Laragon se pueden activar manualmente):**

extension = php\_sqlsrv.dll  
extension = php\_pdo\_sqlsrv.dll

En este caso se escoge la extensión a utilizar que mayor se ajuste a apps modernas, la opción que brinde mas seguridad y flexibilidad, la cual viene a ser la **pdo\_sqlsrv**.

Un ejemplo de un código de conexión con la BD en el Backend con PHP suele verse de la siguiente manera:

```
<?php
$serverName = "localhost";
$connectionOptions = [
    "Database" => "coopeCEP_Website",
    "Uid" => "adminWeb",
    "PWD" => "contraseña",
];

try {

    $conn = new PDO("sqlsrv:Server=$serverName;Database=coopeCEP_Website",
                    $connectionOptions["Uid"],
                    $connectionOptions["PWD"]);
    $conn->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    echo "Conectado correctamente a la BD!";

} catch (PDOException $e) {

    echo "Error en la conexión: " . $e->getMessage();

}
```

## **4. Backend**

### **1. Arquitectura del Backend**

Para el desarrollo del sistema se implementó una **Arquitectura por Capas**, debido a que ofrece una separación clara entre la interfaz de usuario, la lógica de negocio y el acceso a datos. Esta arquitectura permite mantener ordenado el código, facilita la escalabilidad y mejora la mantenibilidad del sistema a largo plazo.

#### **Justificación**

- La arquitectura por capas permite un flujo claro entre la presentación (vistas), la lógica del sistema y el manejo directo de la base de datos .
- Se adapta perfectamente al uso de PHP con una estructura organizada de carpetas.

#### **Capas utilizadas**

- **Capa de Presentación:** archivos PHP dentro de app/views encargados de mostrar las páginas al usuario.
- **Capa de Lógica de Negocio:** controladores que procesan solicitudes y aplican validaciones.
- **Capa de Datos:** modelos encargados de las consultas SQL y conexión al motor de base de datos.

## 2. Componentes y estructura del código

El proyecto se organizó siguiendo un patrón inspirado en **MVC (Modelo - Vista - Controlador)**, permitiendo que cada parte del sistema tenga una responsabilidad clara.

### Componentes

- **Controladores:** reciben las peticiones, procesan datos enviados por el usuario, coordinan la lógica del sistema y deciden qué vista se mostrará.
- **Modelos:** contienen las funciones que interactúan con la base de datos con consultas SELECT, INSERT, UPDATE, DELETE.
- **Vistas:** archivos PHP que renderizan contenido HTML.
- **Servicios auxiliares:** clases de apoyo como validación, seguridad, etc.

### Diagrama general de flujo

Cliente → Rutas → Controlador → Modelo → BD → Controlador → Vista → Cliente

Este flujo permite que el backend sea ordenado, fácil de corregir y escalar.

### **3. Elementos clave del Backend**

#### **Seguridad**

- Se usa validación de datos desde el backend para evitar inyección SQL.
- Rutas de administrador protegidas mediante sesiones.

#### **Gestión de errores**

- Manejo de errores con try/catch en consultas SQL.
- Mensajes descriptivos controlados para el usuario.

#### **Servicios externos**

Aunque el proyecto funciona principalmente con recursos internos, se deja prevista la capacidad de integrar:

- APIs externas para validaciones,
- consumo de servicios web,
- almacenamiento en la nube en futuras versiones.

#### **Pruebas**

Se han realizado pruebas de:

- Unidad: validación de campos.
- Funcionales: inserción, edición y eliminación de datos.
- Integración: comunicación entre controlador / modelo /base de datos.

#### **Escalabilidad**

El backend puede crecer sin problemas porque:

- Cada módulo está separado por capas.
- Es fácil agregar nuevos controladores y modelos.
- Se puede migrar a MVC completo en el futuro sin reescribir todo.

## 4. Conexión con la base de datos y el frontend

El archivo database.php es el responsable de establecer la comunicación entre el backend en PHP y la base de datos MySQL utilizada en el proyecto.

Centralizar esta conexión permite que todos los módulos del sistema (controladores, modelos y servicios) accedan a la misma base de datos de forma segura, eficiente y consistente.

En este proyecto se utiliza MySQL, ejecutándose a través del entorno de desarrollo WAMP, que incorpora el servidor web Apache, PHP y MySQL en un mismo paquete.

El archivo define los parámetros básicos necesarios para conectarse:

- Servidor: normalmente localhost
- Usuario: por defecto en WAMP suele ser root
- Contraseña: vacía por defecto
- Nombre de la base de datos: CoopeCEP\_Website

La conexión se realiza utilizando la extensión MySQLi, mediante la clase mysqli.

```
config > database.php > ...
1  <?php
2
3  $host = "localhost";
4  $user = "root";
5  $pass = "";
6  $dbname = "CoopeCEP_Website";
7
8  $conn = new mysqli($host, $user, $pass, $dbname);
9
10 if ($conn->connect_error) {
11     die("Error de conexión: " . $conn->connect_error);
12 }
13
14 ?>
```

## 5. Capturas o fragmentos de código

Footer:

El archivo footer.php representa la sección inferior del sitio web, diseñada siguiendo el mockup correspondiente. Esta sección incluye un fondo gris, el logotipo institucional en tamaño reducido, enlaces rápidos a las secciones principales y la información básica del centro educativo.

Este archivo también se incluye en todas las vistas mediante, `<?php include`, lo que garantiza consistencia en el diseño y facilita el mantenimiento. Cualquier actualización realizada en el footer se refleja automáticamente en todas las páginas del sitio web, manteniendo orden y coherencia visual.

```
<footer class="Footer">
  Click to collapse the range.
  <!-- LOGO -->
  <div class="footer-logo">
    
  </div>

  <!-- ENLACES RÁPIDOS -->
  <div class="footer-links">
    <h3>Enlaces Rápidos</h3>
    <ul>
      <li><a href="http://localhost/CoopeCEP/app/views/home.php">Inicio</a></li>
      <li><a href="http://localhost/CoopeCEP/app/views/programas.php">Programas</a></li>
      <li><a href="http://localhost/CoopeCEP/app/views/about.php">Acerca de</a></li>
      <li><a href="http://localhost/CoopeCEP/app/views/contacto.php">Contáctanos</a></li>
      <li><a href="http://localhost/CoopeCEP/admin/index.php">Plataforma Virtual</a></li>
    </ul>
  </div>

  <!-- INFORMACIÓN -->
  <div class="footer-info">
    <h3>Contáctanos</h3>
    <p>Teléfono: 2233-4455</p>
    <p>Email: info@coopecep.cr</p>
  </div>
</div>
</footer>

</body>

</html>
```

## Header:

El archivo header.php contiene la parte superior del sitio web, incluyendo el logotipo institucional y la barra de navegación. Este archivo se incluye en todas las páginas del proyecto mediante `<?php include`, lo que permite mantener una interfaz gráfica consistente y facilita el mantenimiento del diseño.

El header incorpora la estructura básica del documento HTML, la hoja de estilos principal y el menú de navegación basado directamente en los mockups. Los elementos principales son: el logotipo del Centro Educativo, la barra de navegación con enlaces a Home, Programas, Contacto y Acerca de Nosotros, y la importación del archivo style.css. Este archivo permite estandarizar la interfaz del sitio y aplicar cambios globales sin tener que editar cada página individualmente.

```
<!-- HEADER -->

<!DOCTYPE html>

<html lang="es">

<head>

    <meta charset="UTF-8">
    <title>CoopeCEP</title>
    <meta name="viewport" content="width=device-width, initial-scale=1.0">

    <!-- Conexión a La Hoja de estilos -->
    <link rel="stylesheet" href="http://localhost/CoopeCEP/public/css/style.css">

</head>

<body>

<header class="header">
    <nav class="nav">
        <ul class="menu">
            <li><a href="http://localhost/CoopeCEP/app/views/home.php">Inicio</a></li>
            <li><a href="http://localhost/CoopeCEP/app/views/programas.php">Programas</a></li>
            <li><a href="http://localhost/CoopeCEP/app/views/about.php">Acerca de</a></li>
            <li><a href="http://localhost/CoopeCEP/app/views/contacto.php">Contáctanos</a></li>
            <li><a href="http://localhost/CoopeCEP/admin/index.php">Plataforma Virtual</a></li>
        </ul>
    </nav>
</header>
```