

Universidad San Carlos de Guatemala
Centro Universitario de Occidente
División de Ciencias de la Ingeniería
Teoría de sistemas 1
Ing. Pedro Domingo

Byron Fernando Torres Ajxup 201731523

Manual Tecnico Hola ke hace

ID: #16

Sistema de Publicación de anuncios

Quetzaltenango 04 de Noviembre de 2024

Indice

Indice	2
Introduccion	3
Requerimientos Técnicos	4
Hardware	4
Software	4
Arquitectura del Sistema	4
Diagrama de Arquitectura	4
Diagrama de Caso de Uso	5
Instalación del Servidor Web	6
Configuración del Proyecto	6
Ejecución	6
Descripción del Código	6
Estructura de Archivos	6
Base de Datos	7
Tabla Usuarios:	7
Diagrama:	7

Introduccion

Este manual técnico describe la configuración detallada del entorno, la estructura de archivos y la implementación técnica del sistema de publicaciones. Incluye instrucciones para desarrolladores, arquitectos de sistemas y administradores que necesiten realizar ajustes o implementar mejoras en el sistema.

Requisitos del Entorno

- **Sistema Operativo:** Ubuntu
- **Backend:** PHP ejecutado sobre Apache (XAMPP)
- **Frontend:** React, manejado con Node.js v16.14.0
- **Base de Datos:** MySQL

2. Configuración del Entorno

1. Backend (PHP):

- Coloca el proyecto PHP en la carpeta
`/opt/lampp/htdocs/olaketel/proyectos.`
- Asegúrate de que los permisos de la carpeta permiten la lectura y ejecución por parte de Apache.

2. Frontend (React):

- Instala las dependencias del proyecto con:
 - i. `npm install`
 - ii. `npm run dev`

Conexión de la API y el Frontend

- La API PHP debe estar configurada para aceptar solicitudes CORS desde el frontend.
- La autenticación se maneja mediante sesiones PHP o cookies, según la configuración.
- La base URL de la API debe coincidir con la configuración del frontend para evitar errores CORS.

Configuración de la Base de Datos

- **Nombre de la Base de Datos:** `publicaciones`
- **Usuario y Permisos:** Asegúrate de que el usuario de MySQL tenga permisos de lectura, escritura y modificación en la base de datos.

Endpoints Principales del Backend

A continuación, se presentan algunos de los endpoints disponibles en el backend, especificando sus métodos HTTP, rutas, y una breve descripción de su funcionalidad:

1. Registro y Autenticación

- `POST /backend/public?action=register`: Registra a un nuevo usuario.
- `POST /backend/public?action=login`: Inicia sesión para un usuario registrado.
- `POST /backend/public?action=logout`: Finaliza la sesión actual del usuario.

2. Gestión de Publicaciones

- `POST /backend/public?action=crearpst`: Crea una nueva publicación.
- `PUT /backend/public?action=editPost`: Edita una publicación existente.
- `PUT /backend/public?action=aprovepst`: Aprueba una publicación específica.

3. Gestión de Reportes

- `POST /backend/public?action=reportar`: Genera un reporte de publicación.
- `GET /backend/public?action=getreport`: Obtiene detalles de reportes.
- `PUT /backend/public?action=ignore`: Ignora un reporte específico.
- `PUT /backend/public?action=aprove`: Aprueba reportes pendientes.

4. Interacción con Categorías

- `POST /backend/public?action=crearcst`: Crea una nueva categoría.
- `GET /backend/public?action=getcst`: Obtiene la lista de categorías.

5. Gestión de Notificaciones

- `GET /backend/public?action=getnoti&idU={idU}`: Obtiene notificaciones para un usuario en específico.

6. Detalles y Asistencias en Eventos

- `POST /backend/public?action=asistir`: Registra la asistencia de un usuario a un evento.
- `GET /backend/public?action=getdetalles&idPost={idPost}`: Obtiene los detalles de una publicación específica.

Scripts SQL

-- Tabla Categoria

```
CREATE TABLE Categoria (
```

```
    idCategoria INT PRIMARY KEY AUTO_INCREMENT, -- Identificador único de cada categoría
```

```
    titulo VARCHAR(100) NOT NULL -- Título de la categoría
```

```
);
```

-- Tabla Publicacion

```
CREATE TABLE Publicacion (
```

```
    id INT PRIMARY KEY AUTO_INCREMENT, -- Identificador único de cada publicación
```

```
    idCategoria INT, -- Clave foránea que conecta con la tabla Categoria
```

```
    titulo VARCHAR(255) NOT NULL, -- Título de la publicación
```

```
    descripcion TEXT NOT NULL, -- Descripción de la publicación
```

```
    fechaALlervarse DATE NOT NULL, -- Fecha del evento
```

```
    horaALlervarse TIME NOT NULL, -- Hora del evento
```

```
    tipo ENUM('publico', 'privado') NOT NULL, -- Tipo de evento (publico o privado)
```

```
    usuarioid INT NOT NULL, -- Identificador del usuario que creó la publicación
```

```
    aprobado BOOLEAN DEFAULT FALSE, -- Si la publicación está aprobada o no
```

```
urlImagen VARCHAR(255), -- URL de la imagen asociada
reportes BOOLEAN DEFAULT FALSE, -- Indica si la publicación tiene reportes o no
FOREIGN KEY (idCategoria) REFERENCES Categoria(idCategoria) -- Clave foránea
a la tabla Categoria
);
```

-- Insertar categorías en la tabla Categoria

```
INSERT INTO Categoria (titulo) VALUES
```

```
('Concierto'),
```

```
('Deporte'),
```

```
('Musica'),
```

```
('Ciencia'),
```

```
('Educacion'),
```

```
('Conferencia');
```

-- Insertar publicaciones en la tabla Publicacion

```
INSERT INTO Publicacion (idCategoria, titulo, descripcion, fechaALlervarse,
horaALlervarse, tipo, usuarioid, aprobado, urlImagen, )
```

```
VALUES
```

-- Publicación 1: Concierto

```
(1, 'Concierto Rock en Xela', 'Un espectacular concierto de rock con bandas locales e
internacionales.', '2024-12-15', '19:00:00', 'publico', 2, TRUE,
'https://example.com/concierto1.jpg', FALSE),
```

-- Publicación 2: Deporte

```
(2, 'Maratón Xela 10K', 'Un maratón abierto a todo el público en las calles de Xela.',
'2024-11-20', '08:00:00', 'publico', 3, FALSE, 'https://example.com/maraton.jpg', FALSE),
```

-- Publicación 3: Musica

(3, 'Festival de Jazz', 'Disfruta de una noche de jazz con artistas nacionales.', '2024-12-05', '20:00:00', 'publico', 4, TRUE, 'https://example.com/jazzfestival.jpg', FALSE),

-- Publicación 4: Ciencia

(4, 'Feria Científica de Xela', 'Evento donde se expondrán proyectos científicos de estudiantes locales.', '2024-10-30', '09:00:00', 'publico', 5, FALSE, 'https://example.com/feriacientifica.jpg', FALSE),

-- Publicación 5: Educacion

(5, 'Seminario de Innovación Educativa', 'Seminario sobre las nuevas metodologías de enseñanza en la era digital.', '2024-11-10', '14:00:00', 'privado', 6, TRUE, 'https://example.com/seminarioeducacion.jpg', FALSE),

-- Publicación 6: Conferencia

(6, 'Conferencia de Tecnología en Xela', 'Expertos en tecnología compartirán las últimas tendencias en desarrollo de software.', '2024-11-25', '10:00:00', 'privado', 7, TRUE, 'https://example.com/conferenciatecnologia.jpg', FALSE);

-- Modificar la tabla Publicacion para agregar las nuevas columnas

ALTER TABLE Publicacion

ADD COLUMN contadorReportes INT DEFAULT 0, -- Contador de reportes, inicia en 0

ADD COLUMN visible BOOLEAN DEFAULT TRUE; -- Indica si la publicación es visible, por defecto TRUE

-- Crear la tabla Reportar


```

CREATE TABLE Reportar (
    id INT PRIMARY KEY AUTO_INCREMENT, -- Identificador único para el reporte
    idUsuario INT NOT NULL, -- Usuario que realizó el reporte
    idPublicacion INT NOT NULL, -- Publicación que está siendo reportada
    aprobar BOOLEAN DEFAULT FALSE, -- Indica si el reporte fue aprobado por un
admin
    motivo TEXT NOT NULL, -- Motivo del reporte
    fechaReporte TIMESTAMP DEFAULT CURRENT_TIMESTAMP, -- Fecha en que se
hizo el reporte
    FOREIGN KEY (idUsuario) REFERENCES Usuarios(id), -- Clave foránea hacia la
tabla Usuarios
    FOREIGN KEY (idPublicacion) REFERENCES Publicacion(id) -- Clave foránea hacia
la tabla Publicacion
);

```

-- Crear la tabla Asistencia para manejar la relación de muchos a muchos entre
Usuarios y Publicaciones

```

CREATE TABLE Asistencia (
    id INT PRIMARY KEY AUTO_INCREMENT, -- ID único de la asistencia
    idU INT NOT NULL, -- Clave foránea hacia el usuario
    idP INT NOT NULL, -- Clave foránea hacia la publicación (evento)
    fechaAsistencia TIMESTAMP DEFAULT CURRENT_TIMESTAMP, -- Fecha en la que
el usuario indicó que asistirá
    visible BOOLEAN DEFAULT TRUE, -- Columna visible con valor por defecto
TRUE
    FOREIGN KEY (idU) REFERENCES usuarios(id), -- Relación con la tabla Usuarios
    FOREIGN KEY (idP) REFERENCES Publicacion(id), -- Relación con la tabla
Publicacion
    UNIQUE (idU, idP) -- Un usuario no puede "asistir" al mismo evento más de una vez

```

CONSULTA AVANZADAS, CON FUNCIONES

FUNCION PARA CONSULTAR LAS ASISTENCIAS DE UN USUARIO EN ESPECIFICO:

```
DELIMITER //
CREATE PROCEDURE obtenerAsistenciasPorUsuario(IN idUsuario INT)
BEGIN
    SELECT p.titulo, p.fechaALlervarse, p.horaALlervarse
    FROM Asistencia a
    JOIN Publicacion p ON a.idP = p.id
    WHERE a.idU = idUsuario;
END //
DELIMITER ;
```

FUNCION PARA CONSULTAR DETALLES DE REPORTES CONVINANDO TABLAS REPORTE, POST Y USUARIO

```
DELIMITER $$

-- Primero eliminamos el procedimiento existente
DROP PROCEDURE IF EXISTS obtenerDetallesReporte$$

-- Luego creamos el nuevo procedimiento
CREATE PROCEDURE obtenerDetallesReporte()
BEGIN
    SELECT
        r.id AS idReport,
```

```

        p.titulo AS titulo_publicacion,
        u.nombre AS nombre_usuario,
        r.motivo AS motivo_reporte
FROM
    Publicacion p
JOIN
    usuarios u
ON
    p.usuarioid = u.id
JOIN
    Reportar r
ON
    p.id = r.idP
WHERE
    r.visible = 1;
END $$

DELIMITER ;

```

PROCEDIMIENTO PARA OBTENER DETALLES DEL POST POR USUARIO:

```

DELIMITER $$

```

```

CREATE PROCEDURE GetUserPosts(IN userId INT)
BEGIN

```

```

SELECT
    p.titulo,
    p.lugar,
    p.fechaALlervarse,
    p.horaALlervarse,
    CASE
        WHEN p.reportes = 0 THEN 'false'
        ELSE 'true'
    END AS reporte_estado
FROM
    Publicacion p
WHERE
    p.usuarioId = userId;
END $$

DELIMITER ;

```

);

Rutas y Protección de Rutas en el Frontend

- **Protección de Rutas:** Usar React Router DOM y un middleware en el backend para verificar sesiones.
- **Rutas Condicionales:** Asegurarse de que cada rol acceda únicamente a las rutas correspondientes.

Configuración de Seguridad

- **CORS y Sesiones:** Configurar encabezados CORS y asegurar que las cookies de sesión solo sean accesibles mediante HTTP.
- **Validación de Entradas:** Validar y sanitizar todas las entradas en el backend.

Despliegue y Pruebas

- **Despliegue de Backend:** Asegúrate de que Apache esté configurado y habilitado.
- **Despliegue de Frontend:** Usa el modo de producción generado por `npm run build`.
- **Pruebas:** Ejecuta pruebas unitarias y de integración para asegurar el correcto funcionamiento de los módulos y componentes.

Diagramas:

[Todos los diagramas estan aqui](#)

Conclusión

La implementación de este sistema de publicaciones permite una gestión estructurada y eficiente de contenido y eventos, proporcionando una experiencia de usuario personalizada según el rol. Con este sistema, los usuarios pueden interactuar de manera segura y organizada gracias a la protección de rutas y la validación de autenticación implementada. Este manual técnico y de usuario ofrece los conocimientos esenciales para la configuración, el uso y el mantenimiento del sistema, y sienta las bases para futuras mejoras y