

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
DIVISIÓN DE CIENCIAS DE LA INGENIERÍA
TEORÍA DE SISTEMAS 1



Manual de Usuario y Técnico
[“Proyecto” => “Publicaciones”]

Carné:

201430801

Nombre:

Luis Basilio Baquiaux Sic

Fecha de entrega: lunes 04 de noviembre de 2024.

Contenido

Introducción.....	3
Marco Teórico.....	4
Sistema de Información y Pensamiento Sistémico.....	4
Roles y Control de Acceso.....	4
Reglas de Negocio y Automatización de Procesos.....	5
Tecnologías Web para el Desarrollo de Sistemas de Información.....	5
Manual de Usuario.....	7
Uso de la aplicación.....	7
Funcionalidades de usuario publicador.....	7
Página principal.....	7
Crear cuenta.....	7
Publicaciones.....	8
Filtrar publicaciones.....	8
Iniciar sesión.....	9
Reportar publicación.....	10
Asistir a una publicación.....	11
Mis publicaciones.....	11
Crear publicación.....	12
Notificaciones.....	13
Manual Técnico.....	15
Instalación de la aplicación.....	15
Estructura de archivos.....	15
Configuración de la base de datos.....	16
Creación de la base de datos.....	16
Creación de las tablas.....	17
Creación de los disparadores.....	18
Configuración de las variable de entorno.....	20
Iniciar aplicación.....	22
Conclusiones:.....	24

Introducción

El presente documento presenta el manual de usuario y técnico. El primero hace referencia de cómo usar la aplicación web para la gestión de publicaciones o eventos por parte de los usuarios registrados (publicadores y público en general). También existe el rol de administrador el cuál es el encargado de autorizar o rechazar publicaciones, aprobar algún reporte de una publicación de un usuario en específico.

En el manual técnico se especifica las instalaciones necesarias para que la aplicación funcione correctamente desde la descarga del proyecto hasta la configuración de la base de datos que incluye la creación de la misma, las tablas que contiene y los disparadores (triggers) que sirven para facilitar algunas funcionalidades del sistema.

Marco Teórico

Sistema de Información y Pensamiento Sistémico

Un **sistema de información** es un conjunto organizado de componentes y procesos que permite recolectar, almacenar, procesar y distribuir información, facilitando la toma de decisiones y el cumplimiento de tareas específicas dentro de una organización o comunidad. Estos sistemas se implementan para optimizar el flujo de información y mejorar la eficiencia operativa al permitir que los usuarios interactúen y gestionen datos relevantes de forma rápida y precisa.

El **pensamiento sistémico** aplicado en sistemas de información implica entender y diseñar el sistema considerando cómo sus diferentes partes interactúan y se afectan mutuamente. En este contexto, el sistema de gestión de eventos en diferentes lugares del país de Guatemala debe contemplar la interacción entre diferentes tipos de usuarios (administradores, publicadores de anuncios y usuarios registrados), la publicación y administración de eventos, y las notificaciones. Esto permite identificar los flujos de información, anticipar problemas, y establecer reglas de negocio que alineen el sistema con los objetivos del cliente, en este caso, el manejo efectivo de eventos en una comunidad local.

Roles y Control de Acceso

El **control de acceso** es un aspecto fundamental en el desarrollo de aplicaciones web, especialmente en aquellas donde existen distintos roles de usuario, cada uno con permisos y responsabilidades específicas. En este sistema de gestión de eventos, se definen tres roles principales: administrador, publicador de anuncios y usuario registrado.

- **Administrador:** Tiene acceso a todas las funcionalidades del sistema, incluyendo la aprobación de publicaciones y la gestión de reportes. Este rol permite mantener el control y la seguridad en el contenido, asegurando que sólo eventos legítimos sean visibles para el público.
- **Publicador de Anuncios:** Este usuario puede crear y gestionar sus publicaciones de eventos. Dependiendo de su reputación (evaluada a través de la cantidad de publicaciones aprobadas), algunos publicadores pueden obtener privilegios de publicación automática.
- **Usuario Registrado:** Este usuario tiene acceso a eventos específicos y puede indicar su interés en asistir a los mismos, recibir notificaciones y reportar eventos inapropiados. La definición de este rol permite un mejor control sobre la audiencia y facilita la interacción del público con el sistema.

Observación: Para simplicidad el usuario registrado y el publicador de anuncios o eventos son la misma entidad, en general un usuario puede publicar sus eventos y asistir a eventos de otros usuarios.

Reglas de Negocio y Automatización de Procesos

Las **reglas de negocio** son lineamientos específicos que definen cómo deben comportarse los procesos dentro de un sistema para que cumpla su propósito de manera efectiva y ordenada. Estas reglas permiten automatizar ciertos procesos y asegurar la integridad y el cumplimiento de políticas en el sistema. En el sistema de gestión de eventos en distintos lugares del país de Guatemala, se establecen reglas que afectan las acciones de publicación, reporte y aprobación de anuncios.

Algunas de las reglas de negocio implementadas incluyen:

- **Aprobación automática para publicadores confiables:** Los publicadores que hayan tenido al menos dos publicaciones aprobadas pueden publicar automáticamente. Esto agiliza el flujo de trabajo y facilita la inclusión de eventos sin intervención constante del administrador.
- **Sistema de reportes y manejo de contenido inapropiado:** Los usuarios registrados pueden reportar anuncios y deben indicar el motivo. Si un anuncio acumula tres reportes, se oculta de la vista general hasta que el administrador tome una acción, sea aprobar o ignorar el reporte. Esto contribuye a mantener la calidad y confiabilidad de la información en el sistema.
- **Pérdida de privilegios:** Los publicadores que sean reportados y cuyos reportes sean confirmados pierden el privilegio de publicación automática, y si siguen siendo problemáticos, pueden ser baneados del sistema. Este mecanismo de sanción protege a la comunidad y evita el abuso del sistema.

Estas reglas de negocio automatizan y controlan los procesos internos, permitiendo que el sistema responda de manera eficiente a eventos específicos sin requerir intervención humana constante, lo cual reduce errores y facilita la gestión de eventos en el tiempo.

Tecnologías Web para el Desarrollo de Sistemas de Información

Para la implementación del sistema, se utiliza el stack de tecnologías de **XAMPP**, que incluye **PHP, MySQL/MariaDB, HTML, CSS y JavaScript**. Esta combinación permite desarrollar aplicaciones web robustas y escalables, asegurando compatibilidad con los servidores del Cunoc y aprovechando herramientas de código abierto ampliamente utilizadas y respaldadas.

- **PHP:** Lenguaje de scripting del lado del servidor utilizado para gestionar la lógica del sistema, manejar las peticiones de los usuarios y comunicarse con la base de datos.
- **Laravel:** es un framework de código abierto para desarrollar aplicaciones y servicios web con PHP. Su filosofía es desarrollar código PHP de forma elegante y simple, evitando el "código espagueti".
- **MySQL:** Sistema de gestión de bases de datos relacional que almacena la información de los eventos, usuarios, y roles, permitiendo el acceso y manipulación eficiente de los datos.

- **HTML, CSS, y JavaScript:** Tecnologías front-end empleadas para la estructura, estilo e interactividad del sistema, ofreciendo una interfaz amigable y accesible para los usuarios.

Estas herramientas permiten que el sistema de gestión de eventos se desarrolle de manera modular, siguiendo una arquitectura cliente-servidor que facilita la escalabilidad, el mantenimiento y la adaptabilidad del sistema a futuro.

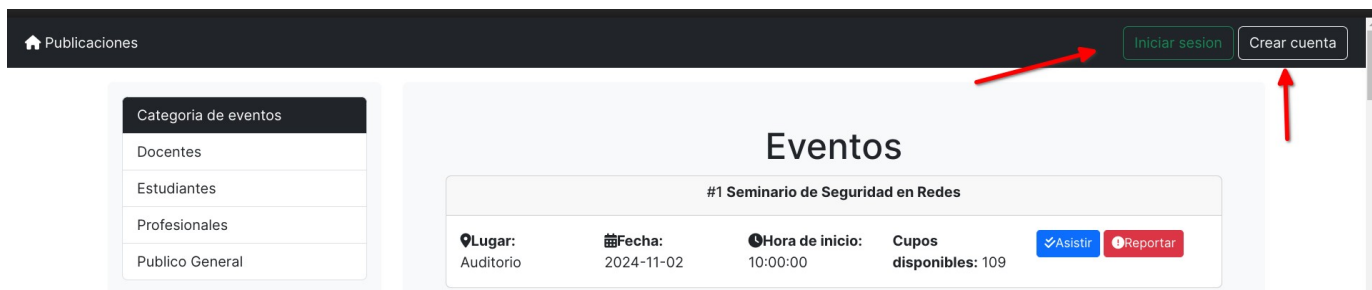
Manual de Usuario

Uso de la aplicación

Funcionalidades de usuario publicador

Página principal

Al iniciar la aplicación web mostrará por defecto las publicaciones y el menú opciones para iniciar sesión, crear cuenta.



Crear cuenta

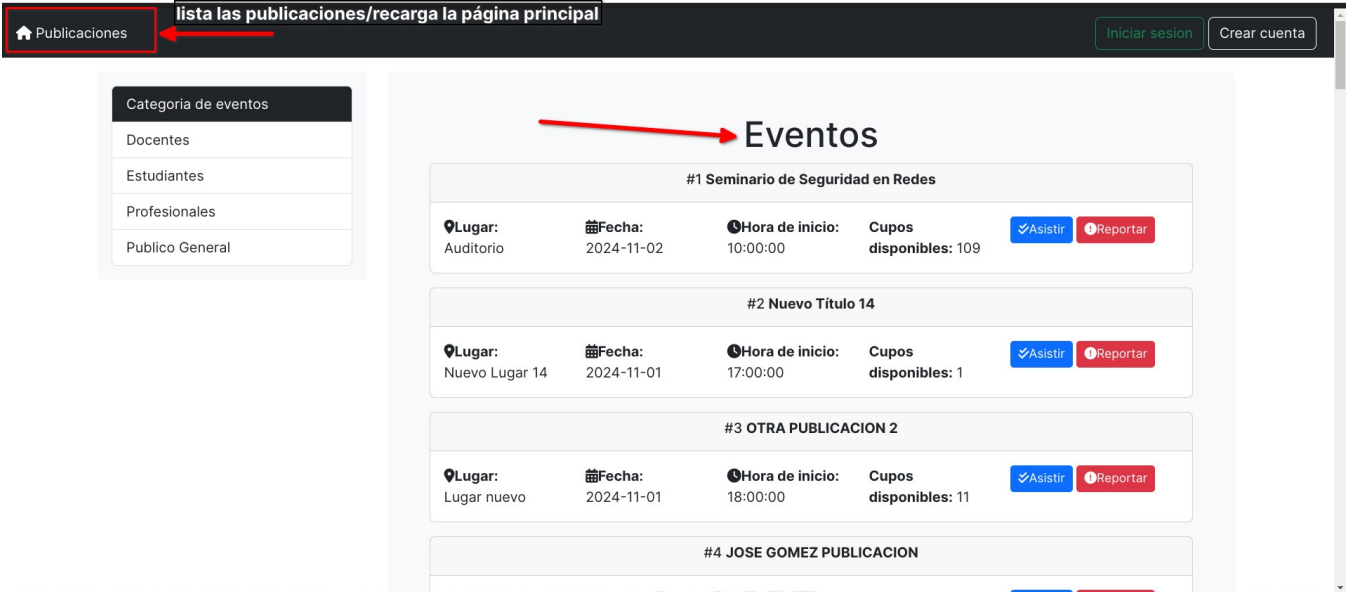
Mostrará un formulario para registrarse, deberá llenar los campos necesarios para guardar su información, después debe presionar en el botón "Crear cuenta".

A screenshot of the 'Registro' (Registration) form. The form is set against a dark background with a red-to-black gradient on the left. At the top center is a logo consisting of three stacked cubes. Below the logo, the title 'Registro' is displayed in white. The form contains several input fields arranged in two columns. The left column has fields for 'Nombre:', 'Apellido:', 'Telefono:', and 'Correo electrónico:'. The right column has fields for 'Nombre de usuario:' (containing the text 'useradmin'), 'Contraseña:', and 'Confirmar contraseña:'. At the bottom of the form is a large red button labeled 'Crear cuenta'. A callout box with a black border and the text 'Presionar para guaradr cambios' (note the typo) has a green arrow pointing to the 'Crear cuenta' button. At the very bottom of the form is a red button labeled '<< Regresar'.

El botón “Regresar” mostrará la página principal.

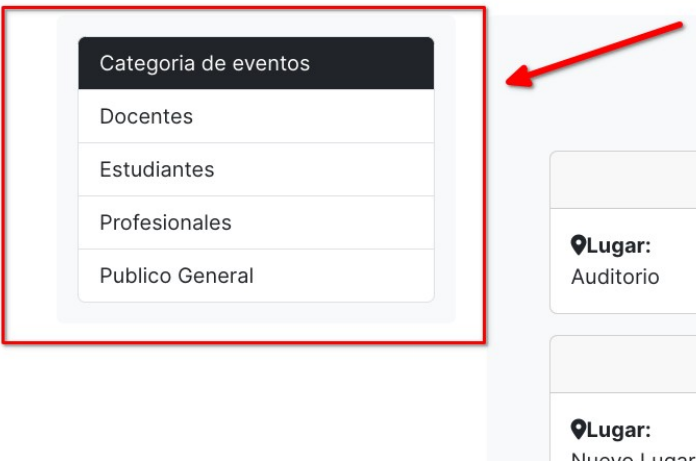
Publicaciones

En el menú aparece la opción de “Publicaciones”, su función principal es recargar la página principal para mostrar las publicaciones.



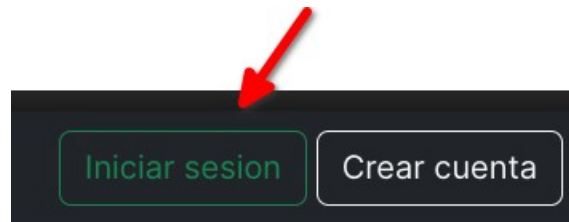
Filtrar publicaciones

El siguiente recuadro servirá para poder filtrar las publicaciones por categoría que existen en el sistema de información.



Iniciar sesión

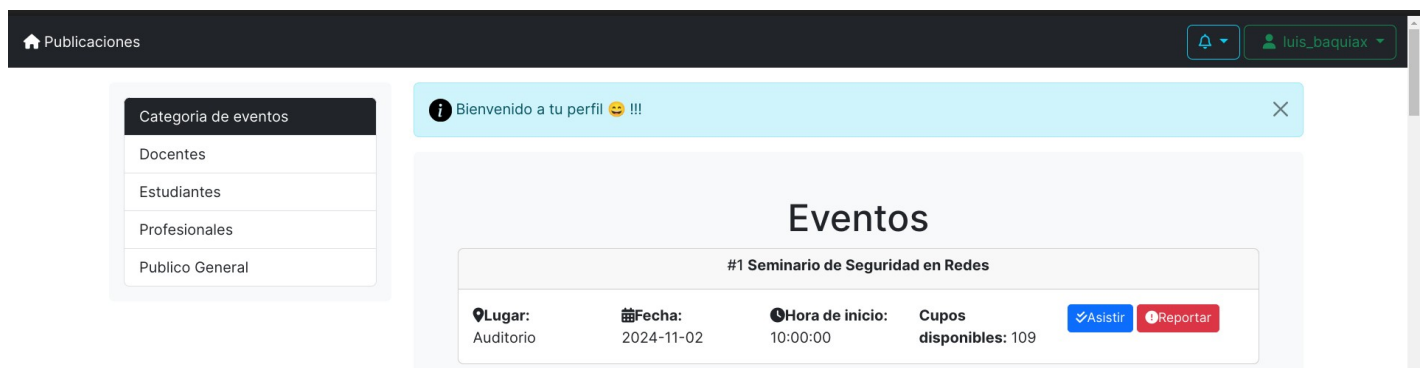
Para iniciar sesión debe presionar el botón “Iniciar sesión” el cual le mostrará la página para ingresar su nombre de usuario y contraseña.



Después de presionar el botón para iniciar sesión la aplicación le mostrará un formulario para ingresar sus credenciales.

Una imagen de un formulario de inicio de sesión sobre un fondo oscuro. En la parte superior derecha hay un logo de tres cubos blancos. El formulario tiene tres campos de entrada: 'Nombre de usuario:' con el texto 'useradmin', 'Contraseña:' con caracteres ocultos por puntos, y un botón 'Ingresar'. Hay tres flechas verdes con etiquetas rojas que indican la interacción: 'ingresr username' apunta al primer campo, 'Ingresar contraseña' apunta al segundo, y 'Presionar para inicar sesión' apunta al botón 'Ingresar'.

Al iniciar sesión se le mostrará la página de su perfil.



Reportar publicación

Para reportar una publicación deberá haber iniciado sesión, después deberá presionar el botón “Reportar”, esto

Eventos

#1 Seminario de Seguridad en Redes

Lugar:
Auditorio

Fecha:
2024-11-02

Hora de inicio:

Cupos disponibles: 109

Asistir

Reportar

El botón “Reportar” le mostrará un modal para seleccionar el motivo y confirmar el reporte.

Reportar anuncio: Seminario de Seguridad en Redes

Seleccione el motivo del reporte de esta publicación

Causa controversia

Reportar anuncio

Cancelar

Después de confirmar se mostrará un mensaje de confirmación de la acción.

i Se ha reportado la publicación exitosamente

Eventos

Mensaje cuando se ha reportado una publicación

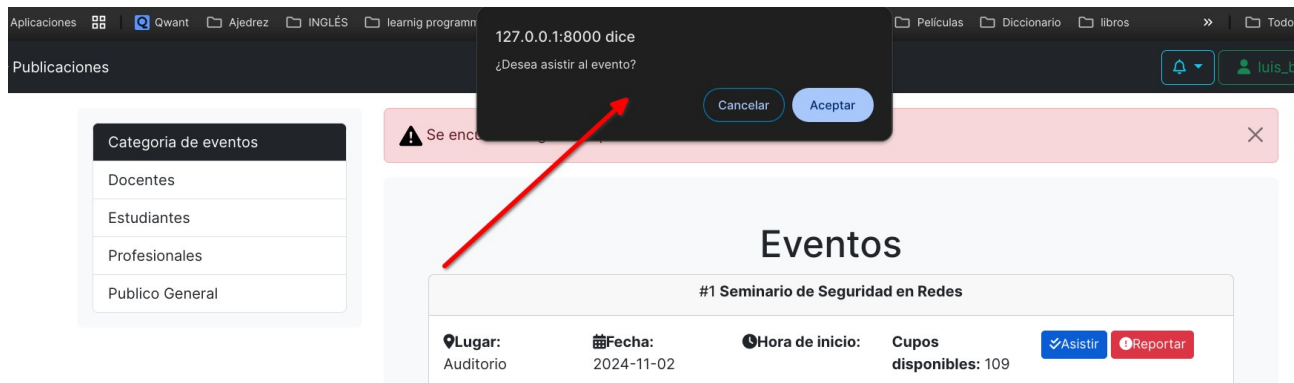
A Se encuentra registrado para este evento

Eventos

Mensaje cuando ya se había reportado anteriormente

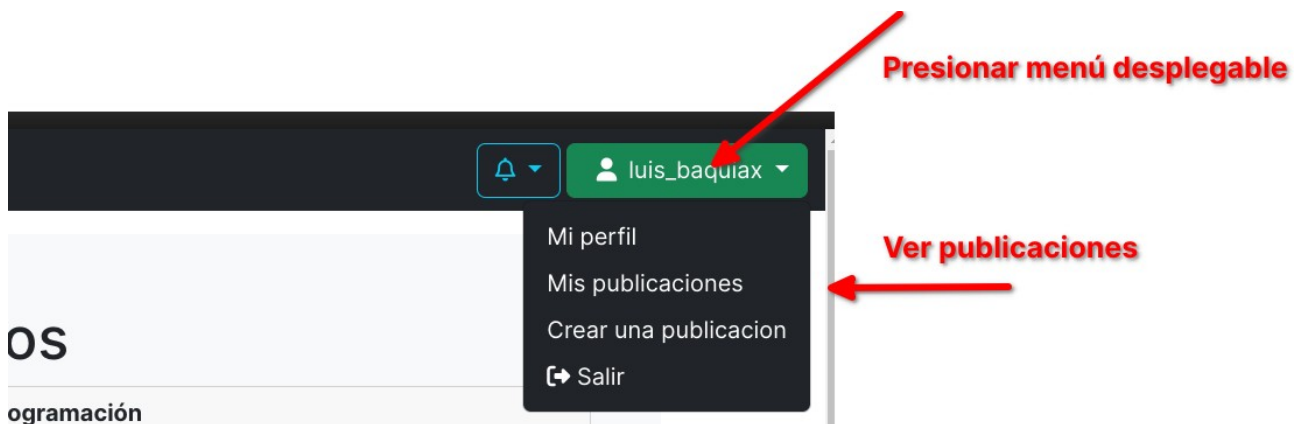
Asistir a una publicación

Para registrar asistencia a un evento se deberá presionar el botón “Asistir” el cuál le mostrará una ventana emergente para continuar la acción.



Mis publicaciones

Mostrará todas las publicaciones hechas por el usuario.



“Mis publicaciones”, mostrará las publicaciones creadas por el usuario.

Eventos

#1 Conferencia de Programación

Lugar: Sala A **Fecha:** 2024-10-01 **Hora de inicio:** 10:00:00 **Cupos disponibles:** 100 **Estado:** RECHAZADO



#2 Seminario de Ciberseguridad

Lugar: Auditorio **Fecha:** 2024-10-05 **Hora de inicio:** 10:00:00 **Cupos disponibles:** 150 **Estado:** ACEPTADO



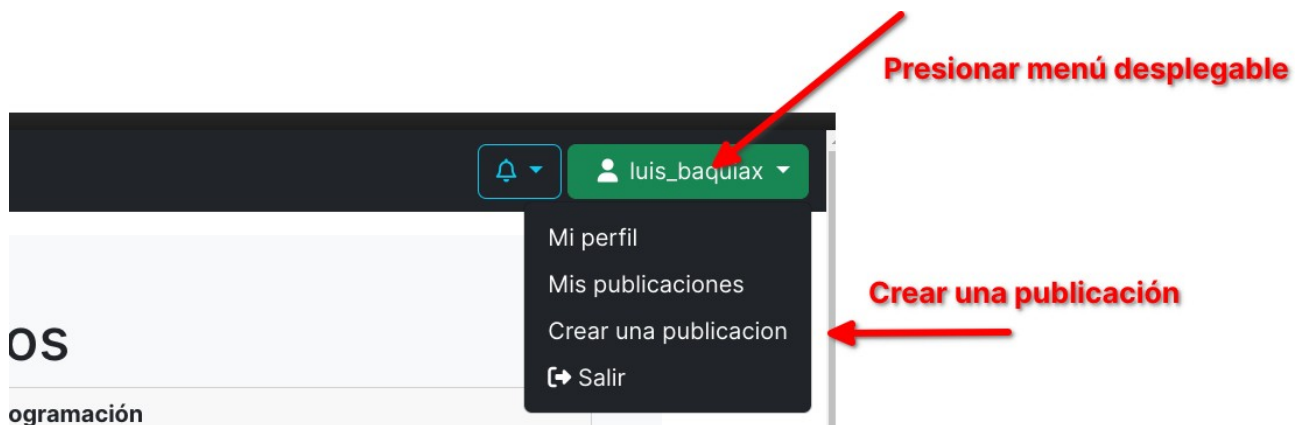
#3 Charla sobre Inteligencia Artificial

Lugar: Sala B **Fecha:** 2024-10-10 **Hora de inicio:** 10:00:00 **Cupos disponibles:** 80 **Estado:** ACEPTADO



Crear publicación

La opción “Crear una publicación” mostrará el formulario para crear una publicación.



Formulario para crear una publicación:

Creando publicación

Título:

Hora de inicio:

Lugar:

Tipo de público:

- Docentes
- Estudiantes
- Profesionales
- Publico General

Fecha:

Cupos disponibles:

Confirmar acción →

Regresar a la ventana anterior →

Debe llenar todos lo campos y pulsar el botón “Crear publicación”.

El botón “Regresar” mostrará la página principal de su perfil y las publicaciones.

Notificaciones

Para ver las notificaciones deberá pulsar sobre la “Campana” esto despliega el submenú para ver los próximos eventos.



El submenu “Eventos próximos” mostrar los eventos que están a punto de realizarse.

Eventos próximos

Título:

Seminario de Seguridad en
Redes

Lugar

📍 Auditorio

Fecha

📅 2024-11-02

Hora de inicio

🕒 10:00:00

Cuenta regresiva

Quedan 12 horas y 45
minutos

Se mostrará toda la información necesaria de los próximos eventos al cual el usuario va a asistir, mostrando el título, lugar, fecha y hora del evento. También mostrará las horas que faltan para que el evento suceda.

Manual Técnico

El siguiente apartado de este documento describe las instrucciones necesarias para la instalación de la aplicación, la configuración de la base de datos desde la creación de la misma, de las tablas que contiene, de los disparadores que usa.

Instalación de la aplicación

1. **Descargar la aplicación** o clonar la aplicación desde el link:
<https://github.com/luisbaquiaux/publicaciones>

2. **Clonar proyecto:** para clonar el proyecto debe seleccionar un directorio para clonar el proyecto y opcionalmente crear una carpeta para guardar el código fuente.

1. En windows

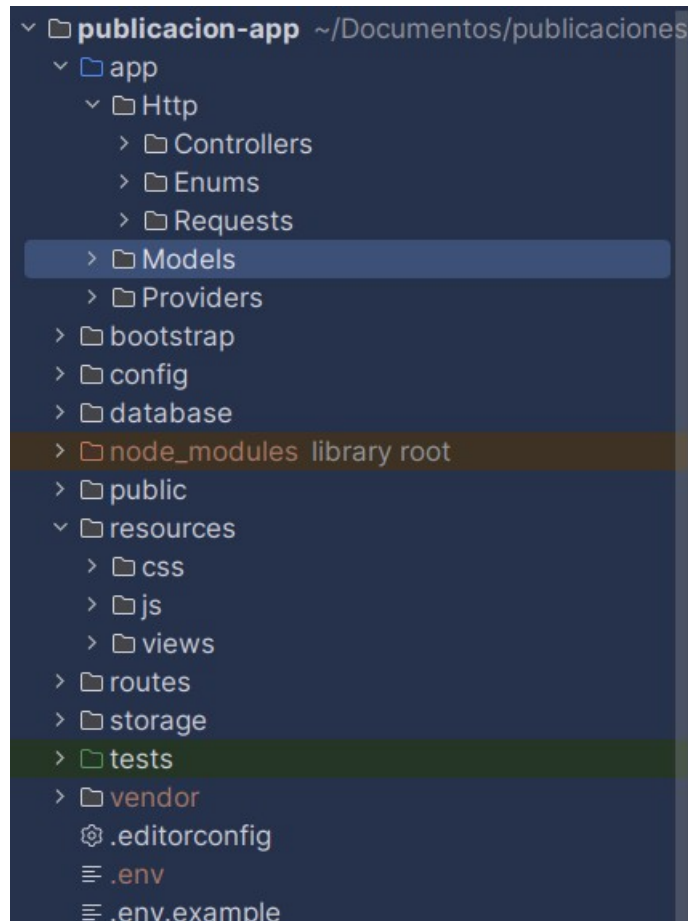
Ejecute: git clone <https://github.com/luisbaquiaux/publicaciones.git>

2. En linux

Ejecute: git clone [git@github.com:luisbaquiaux/publicaciones.git](https://github.com/luisbaquiaux/publicaciones.git)

Estructura de archivos

La siguiente imagen muestra la estructura general de los archivos del proyecto.



Descripción general de carpetas principales:

1. **app**: contiene las clases de los controladores, modelos, enums.
2. **resources**: contiene los archivos de las vistas
3. **Achivo .env**: contiene las configuraciones principalmente la de la base de datos.

Configuración de la base de datos

En el archivo “mapeo publicaciones” se encontrará todo el script para crear la base de datos y la tablas. Asimismo, se adjunta el script para crear los triggers o disparadores que servirán para automatizar algunos procesos de la lógica del negocio.

Creación de la base de datos

El siguiente script crea y usa la base de datos, pero antes verifica si no está creada, en caso que exista lo borra para reiniciar la instalación.

Ejecute el script para crear la base de datos:


```
DROP DATABASE IF EXISTS publicaciones;
CREATE DATABASE publicaciones;
USE publicaciones;
```

Creación de las tablas

Para la creación de las tablas ejecute:

```
CREATE TABLE users(
  username VARCHAR(45) NOT NULL UNIQUE,
  password TEXT NOT NULL,
  email VARCHAR(45) NOT NULL,
  rol VARCHAR(15) NOT NULL,
  nombre VARCHAR(45) NOT NULL,
  apellido VARCHAR(45) NOT NULL,
  telefono VARCHAR(8) NOT NULL,
  estado VARCHAR(15) NOT NULL,
  puede_publicar VARCHAR(5) NOT NULL,
  PRIMARY KEY(username)
);

CREATE TABLE tipo_publico(
  tipo_publico VARCHAR(45) NOT NULL PRIMARY KEY
);

CREATE TABLE publicacion(
  id INT AUTO_INCREMENT NOT NULL,
  titulo TEXT,
  lugar TEXT,
  fecha DATE NOT NULL,
  hora_inicio TIME NOT NULL,
  cupos INT NOT NULL,
  url TEXT,
  username VARCHAR(45) NOT NULL,
  estado VARCHAR(15),
  PRIMARY KEY(id),
  FOREIGN KEY(username) REFERENCES users(username)
);

CREATE TABLE publico(
  id_publicacion INT NOT NULL,
  tipo_publico VARCHAR(45) NOT NULL,
  PRIMARY KEY(id_publicacion, tipo_publico),
  FOREIGN KEY(id_publicacion) REFERENCES publicacion(id),
  FOREIGN KEY(tipo_publico) REFERENCES tipo_publico(tipo_publico)
);

CREATE TABLE asistencia(
  id INT AUTO_INCREMENT NOT NULL,
  id_publicacion INT NOT NULL,
  username VARCHAR(45) NOT NULL,
  PRIMARY KEY(id),
  FOREIGN KEY(username) REFERENCES users(username),
  FOREIGN KEY(id_publicacion) REFERENCES publicacion(id)
);
```

```

CREATE TABLE motivo(
  nombre VARCHAR(45) NOT NULL PRIMARY KEY
);

CREATE TABLE reporte(
  id INT AUTO_INCREMENT NOT NULL,
  id_publicación INT NOT NULL,
  motivo VARCHAR(45) NOT NULL,
  fecha DATE NOT NULL,
  username VARCHAR(45),
  estado VARCHAR(45),
  PRIMARY KEY(id),
  FOREIGN KEY(id_publicacion) REFERENCES publicacion(id),
  FOREIGN KEY(motivo) REFERENCES motivo(nombre),
  FOREIGN KEY(username) REFERENCES users(username)
);

```

Creación de los disparadores

En el archivo “qurys publicaciones” se encuentra el script para crear los disparadores que servirán para actualizar el estado de una publicación si tiene al menos 3 reportes, actualizar si un USUARIO puede publicar de manera automática, para banear al usuario de forma automática si ha sido reportado.

Ejecute el script para crear los disparadores:

```

// trigger para actualizar el estado de una publicacion si tiene al menos 3 reportes
DELIMITER $$

CREATE TRIGGER verificar_cantidad_reporte
AFTER INSERT ON reporte
FOR EACH ROW
BEGIN
  DECLARE num_reportes INT;

  SELECT COUNT(*) INTO num_reportes
  FROM reporte
  WHERE id_publicacion = NEW.id_publicacion;

  IF num_reportes >= 3 THEN
    UPDATE publicacion
    SET estado = 'OCULTA'
    WHERE id = NEW.id_publicacion;
  END IF;
END$$

DELIMITER ;

//trigger para actualizar si un USUARIO puede publicar de manera automatica
DELIMITER $$

```

```

CREATE TRIGGER actualizar_publicacion_automatica
AFTER INSERT ON publicacion
FOR EACH ROW
BEGIN
    DECLARE cantidad_publicacion INT;
    DECLARE puede_publicar_actual VARCHAR(10);

    SELECT puede_publicar INTO puede_publicar_actual FROM users WHERE username =
NEW.username;

    SELECT COUNT(*) INTO cantidad_publicacion
    FROM publicacion
    WHERE username = NEW.username
    AND estado = 'ACEPTADO'
    AND puede_publicar_actual != 'NUNCA';

    IF cantidad_publicacion >= 2 THEN
        UPDATE users
        SET puede_publicar = 'SI'
        WHERE username = NEW.username;
    END IF;
END$$
DELIMITER ;

```

//trigger para banear al usuario

```

DELIMITER $$
CREATE TRIGGER revisar_baneo_usuario
AFTER UPDATE ON reporte
FOR EACH ROW
BEGIN
    DECLARE estado_publicar VARCHAR(5);
    DECLARE user_name VARCHAR(255);

    SELECT u.puede_publicar, u.username INTO estado_publicar, user_name
    FROM users u
    INNER JOIN publicacion p ON p.username = u.username
    INNER JOIN reporte r ON r.id_publicacion = p.id
    WHERE r.id = NEW.id;

    IF NEW.estado = 'APROVADO' THEN
        IF estado_publicar = 'SI' THEN
            UPDATE users SET puede_publicar = 'NUNCA'
            WHERE users.username = user_name;
        ELSEIF estado_publicar = 'NO' THEN
            UPDATE users SET estado = 'BANEADO'
            WHERE users.username = user_name;
        END IF;
    END IF;
END;$$
DELIMITER ;

```

```
// notificaciones

SELECT p.id, p.titulo, p.lugar, p.fecha, p.hora_inicio
FROM publicacion p
INNER JOIN asistencia a ON a.id_publicacion = p.id
INNER JOIN users u ON u.username = a.username
WHERE u.username = 'luis_baquiax'
AND CONCAT(p.fecha, ' ', p.hora_inicio) >= NOW();

SELECT p.id, p.titulo, p.lugar, p.fecha, p.hora_inicio, TIMESTAMPDIFF(HOUR, NOW(), p.fecha) AS
horas_diferencia
FROM publicacion p
INNER JOIN asistencia a ON a.id_publicacion = p.id
INNER JOIN users u ON u.username = a.username
WHERE u.username = ?
AND p.fecha >= NOW();

SELECT p.id, p.titulo, p.lugar, p.fecha, p.hora_inicio,
TIMESTAMPDIFF(HOUR, NOW(), ADDDATE(p.fecha, INTERVAL TIME_FORMAT(p.hora_inicio, '%H:
%i:%s') HOUR_SECOND)) AS minutos_restantes
FROM publicacion p
INNER JOIN asistencia a ON a.id_publicacion = p.id
INNER JOIN users u ON u.username = a.username
WHERE u.username = ?
AND CONCAT(p.fecha, ' ', p.hora_inicio) >= NOW()

SELECT p.id, p.titulo, p.lugar, p.fecha, p.hora_inicio, TIMESTAMPDIFF(MINUTE, NOW(),
CONCAT(p.fecha, ' ', p.hora_inicio)) AS minutos_restantes
FROM publicacion p
INNER JOIN asistencia a ON a.id_publicacion = p.id
INNER JOIN users u ON u.username = a.username
WHERE u.username = 'luis_baquiax'
AND CONCAT(p.fecha, ' ', p.hora_inicio) >= NOW();
```

Configuración de las variable de entorno

Para configurar las variables de entorno es necesario abrir el archivo .env (**ver imagen**) que se encuentra en la carpeta del proyecto “publicaciones”.

Para que la conexión sea exitosa debe configurar este archivo (.env) y modificar los siguientes parámetros:

```
1
2 DB_CONNECTION=mysql
3 DB_HOST=127.0.0.1
4 DB_PORT=3306
5 DB_DATABASE=publicaciones
6 DB_USERNAME=root
7 DB_PASSWORD=luisbaquiaux1234
8
```

DB_CONNECTION=mysql (el motor de base datos)

DB_HOST=127.0.0.1 (ip del servidor)

DB_PORT=3306 (el puerto dónde correrá mysql)

DB_DATABASE=publicaciones (nombre de la base de datos)

DB_USERNAME=root (nombre de usuario, en este caso el root)

DB_PASSWORD=luisbaquiaux1234 (contraseña del usuario)

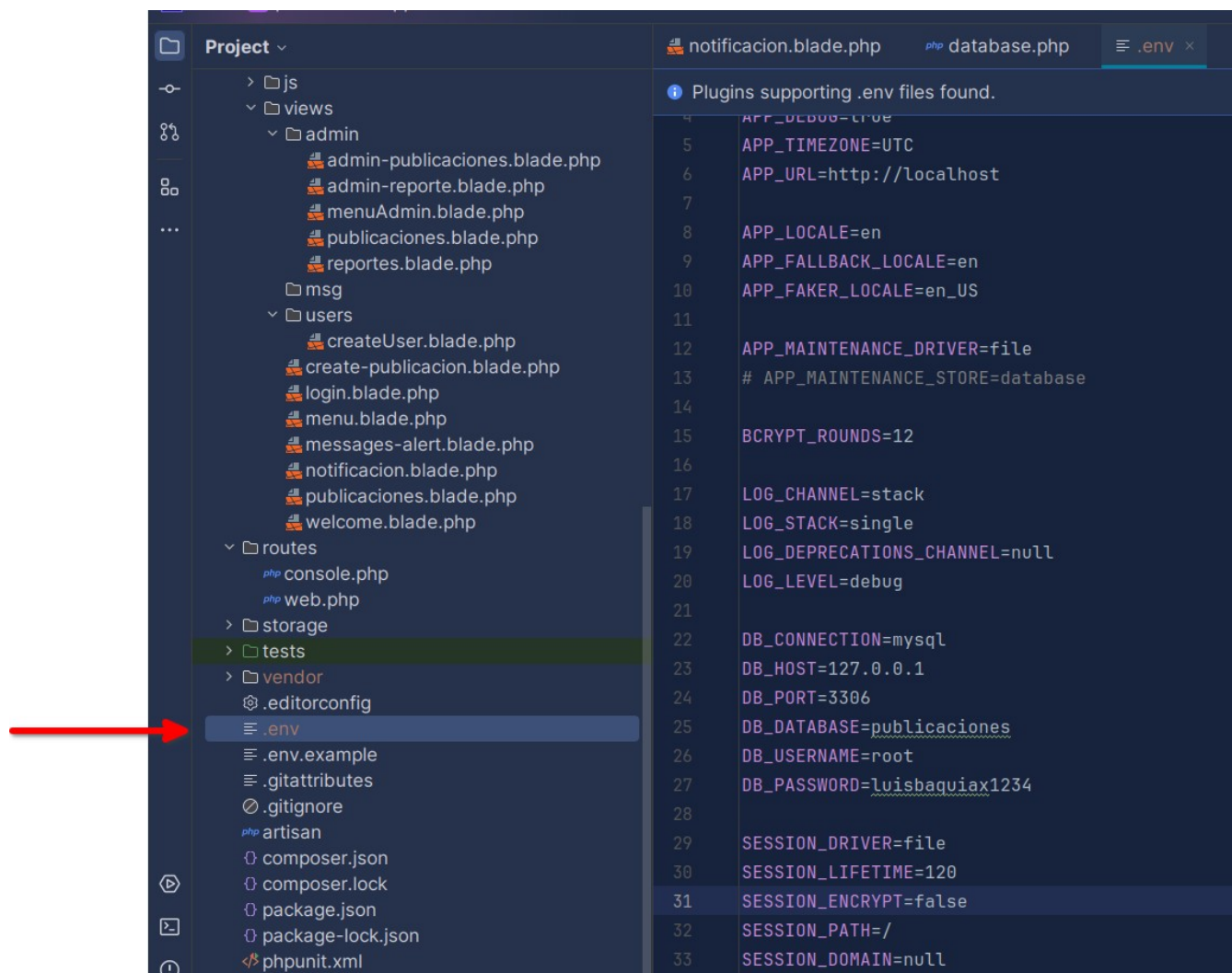


Imagen: Estructura del proyecto.

Iniciar aplicación

1. Usando servidor de desarrollo de laravel

En la carpeta de la aplicación principal “publicacion-app” ejecute: `php artisan serve`, este comando inicia un servidor web local, sin necesidad de configurar un servidor web completo como Apache u otro.

Después use la siguiente url para cargar la aplicación: <http://127.0.0.1:8000/>

2. Usando apache

1. Copiar el proyecto en la carpeta “htdocs”
2. Iniciar el servidor Apache de Xampp

3. En el navegador use la url para iniciar la aplicación: <http://localhost/publicacion-app>

Conclusiones:

1. Es de suma importancia describir las instrucciones necesarias para el uso de cualquier aplicación web dirigido hacia los usuarios finales y también es necesario describir cómo instalar e iniciar la misma, dirigido a desarrolladores.
2. Un sistema de información tiene elementos que cumplen con funciones distintas para llegar hacia un objetivo, en este contexto, tanto como la aplicación y los usuarios (administrador, publicador) tienen diferente función lo que le da vida a esta aplicación llamada “publicaciones”.