

MARCO TEÓRICO

Conceptos de las herramientas Utilizadas

PHP:

PHP (Hypertext Preprocessor) es un lenguaje de programación de propósito general, ampliamente utilizado en el desarrollo web. Permite crear aplicaciones dinámicas e interactivas al integrarse con el HTML y trabajar en el servidor para procesar datos. En este proyecto, PHP maneja las solicitudes de usuarios, procesando datos como los de registro, gestión de publicaciones y almacenamiento de archivos.

MySQL:

MySQL es un sistema de gestión de bases de datos relacional (RDBMS) basado en SQL (Structured Query Language). En este proyecto, se utiliza para almacenar y gestionar datos estructurados, como usuarios, publicaciones y reportes. Algunas características clave de MySQL son su eficiencia en consultas, manejo de grandes volúmenes de datos y soporte para transacciones, esencial para el correcto registro y control de la información.

XAMPP:

XAMPP es un paquete de software que incluye Apache (servidor web), MySQL (base de datos), PHP y Perl, y se usa principalmente para el desarrollo local. XAMPP permite simular un servidor web en tu máquina, permitiéndote probar aplicaciones web antes de desplegarlas en producción.

Bootstrap 5:

Bootstrap es un framework de CSS para el desarrollo de interfaces de usuario responsive y optimizadas para dispositivos móviles. Bootstrap 5 permite implementar rápidamente componentes como formularios, botones, y menús, con un diseño moderno y adaptable. En este proyecto, se usa para dar estilo y mejorar la usabilidad de formularios, paneles de administración y otras interfaces de usuario.

HTML, CSS y JavaScript

- **HTML** (HyperText Markup Language): Lenguaje de marcado estándar para crear la estructura de páginas web.
- **CSS** (Cascading Style Sheets): Lenguaje de estilo usado para definir la apariencia de la página web, incluyendo colores, fuentes y diseño de los elementos.
- **JavaScript**: Lenguaje de programación que se ejecuta en el navegador y permite añadir interactividad a las páginas web, como validaciones de formulario, y funciones dinámicas de frontend.

Git y GitHub

Git es un sistema de control de versiones distribuido, y GitHub es una plataforma de hospedaje para proyectos con Git. Usar Git te permite gestionar y realizar un seguimiento de los cambios en el código, mientras que GitHub facilita la colaboración y el almacenamiento en la nube de tu proyecto.

POO (Programación Orientada a Objetos)

La POO es un paradigma de programación basado en la organización del código en "objetos" que representan entidades del mundo real. En tu proyecto, la POO se emplea para estructurar datos y funciones relacionadas, como la clase **Publicacion**, que contiene atributos y métodos para manejar las publicaciones y sus elementos, facilitando la modularidad y reutilización del código.

SQL y Procedimientos Almacenados

SQL es el lenguaje estándar para interactuar con bases de datos relacionales. Los procedimientos almacenados son conjuntos de instrucciones SQL que se almacenan y ejecutan en el servidor de base de datos, lo cual mejora la eficiencia y centraliza la lógica de negocio en la base de datos. En este proyecto, se han creado procedimientos para manejar operaciones como aprobación de publicaciones y cancelación de asistencia a eventos.

Conceptos del sistema informático:

1. Usuario Publicador

El usuario publicador es aquel que tiene la capacidad de crear y gestionar publicaciones en el sistema. Su rol está orientado a ser el generador de contenido, donde puede:

- **Crear publicaciones:** definir eventos con detalles como lugar, fecha, hora, cupo, público objetivo, y recursos multimedia (imágenes, videos, etc.).
- **Estado de privilegio automático:** se define si el usuario puede publicar sin aprobación dependiendo de la cantidad de publicaciones exitosas. Este privilegio puede cambiar si el usuario acumula más de un número específico de publicaciones o si recibe reportes que reducen su estatus.
- **Manejo de reportes y estado de baneo:** si recibe varios reportes, su cuenta puede ser limitada o incluso suspendida por el administrador.

2. Usuario Participante

El usuario participante es quien interactúa con las publicaciones y eventos creados por los publicadores. Su rol está orientado a ser un consumidor de contenido. Funciones de este tipo de usuario incluyen:

- **Visualización de publicaciones:** el participante puede ver las publicaciones, y obtener detalles como el horario, lugar, descripción y cualquier recurso visual adjunto.
- **Asistencia a eventos:** tiene la capacidad de inscribirse en eventos para asistir, y su asistencia puede estar sujeta a un cupo limitado.
- **Reporte de contenido:** el participante puede reportar publicaciones que considere inapropiadas o que incumplan con las normas del sistema, lo cual desencadena una revisión por parte del administrador.

3. Administrador

El administrador es el usuario con el mayor nivel de privilegio y es responsable de mantener la integridad y el buen funcionamiento del sistema. Sus funciones incluyen:

- **Aprobación de publicaciones:** revisa y aprueba las publicaciones de los publicadores, especialmente si son nuevos y aún no tienen privilegio automático.
- **Gestión de reportes:** verifica los reportes realizados por los participantes y toma decisiones en consecuencia, como aprobar los reportes, banear al publicador, o cambiar el estado de la publicación a "reportado."

MANUAL TÉCNICO:

1. Descripción General del Proyecto

El sistema de gestión de eventos "Ola Ke Hace" permite a los usuarios publicar eventos, participar en ellos y reportar contenidos inapropiados. Está diseñado para contar con tres tipos de usuarios (publicador, participante, y administrador) que tienen roles específicos en el sistema.

2. Entorno de Desarrollo y Herramientas Utilizadas

2.1. XAMPP

Descripción: XAMPP es una plataforma de desarrollo de software que incluye el servidor Apache, la base de datos MySQL/MariaDB, y los intérpretes para lenguajes de programación PHP y Perl.

Función en el proyecto: XAMPP se utiliza como el entorno de desarrollo local para ejecutar la aplicación en un servidor Apache y gestionar la base de datos MySQL.

2.2. MySQL

Descripción: MySQL es un sistema de gestión de bases de datos relacional (RDBMS) que usa SQL (Structured Query Language) para administrar y manipular datos.

Función en el proyecto: MySQL es la base de datos principal para almacenar la información de usuarios, publicaciones, y reportes.

2.3. PHP

Descripción: PHP es un lenguaje de programación del lado del servidor utilizado para desarrollar aplicaciones web dinámicas.

Función en el proyecto: PHP es el lenguaje principal del sistema, responsable de la lógica del servidor, manejo de solicitudes HTTP y conexión con las bases de datos MySQL.

2.4. HTML, CSS, y JavaScript

Descripción: HTML es el lenguaje estándar para estructurar contenido en la web; CSS se utiliza para el diseño visual; y JavaScript permite la interacción en el cliente.

Función en el proyecto: HTML, CSS, y JavaScript se utilizan para crear la interfaz de usuario, estructurando la información y proporcionando interactividad en la aplicación.

2.5. Bootstrap 5

Descripción: Bootstrap es un framework de diseño CSS que permite desarrollar interfaces de usuario responsivas y atractivas.

Función en el proyecto: Bootstrap 5 se usa para el diseño de la interfaz de usuario, proporcionando componentes como formularios, botones, y menús desplegables para una experiencia de usuario consistente.

2.6. Git y GitHub

Descripción: Git es un sistema de control de versiones distribuido que permite gestionar y rastrear cambios en el código fuente. GitHub es una plataforma de alojamiento de código que utiliza Git.

Función en el proyecto: Git y GitHub permiten gestionar el código del proyecto, facilitar la colaboración y realizar el seguimiento de versiones.

3. Estructura de la Base de Datos MySQL

3.1. Tablas Principales

- **usuario_registrado:** almacena la información básica de los usuarios, incluyendo su nombre de usuario, contraseña y tipo (publicador, participante).
- **usuario_publicacion:** extiende la información de los usuarios publicadores, incluyendo estadísticas de publicaciones, estado de privilegio, y nombre completo.
- **publicacion:** almacena los datos de cada publicación o evento, como lugar, fecha, hora, y estado.
- **reporte_publicacion:** registra los reportes de los participantes hacia las publicaciones, permitiendo su revisión por el administrador.

3.2. Relaciones entre las Tablas

Cada usuario registrado puede estar relacionado con una entrada en la tabla `usuario_publicacion` si tiene el rol de publicador. Las publicaciones están asociadas a un publicador, y pueden recibir múltiples reportes de diferentes participantes.

4. Funcionalidad de los Usuarios

4.1. Usuario Publicador

El publicador tiene permisos para:

- Crear y gestionar sus publicaciones.

4.2. Usuario Participante

El participante puede:

- Visualizar las publicaciones de eventos y registrarse para asistir.
- Reportar publicaciones que considere inapropiadas.

4.3. Administrador

El administrador tiene el control sobre:

- Aprobación o rechazo de publicaciones.
- Gestión de reportes y decisión de acciones como suspensión de usuarios.

5. Funcionalidades Clave del Sistema

5.1. Creación de Publicaciones

Los publicadores pueden crear eventos en el sistema mediante un formulario que captura detalles como fecha, lugar y hora. Las publicaciones pueden incluir recursos multimedia.

5.2. Gestión de Reportes

Los participantes pueden reportar publicaciones. Cuando se alcanza un cierto umbral de reportes, el sistema notifica al administrador, quien toma medidas para mantener la calidad del contenido.

5.3. Sistema de Privilegios

Los publicadores pueden ganar el privilegio automático para publicar sin aprobación previa. Este privilegio se ajusta en función de su historial y el número de reportes recibidos.

6. Configuración del Entorno de Desarrollo

6.1. Requisitos del Sistema

- **XAMPP:** Para entornos de desarrollo local
- **PHP:** Versión 8.2 o superior
- **MySQL/:** Base de datos relacional.

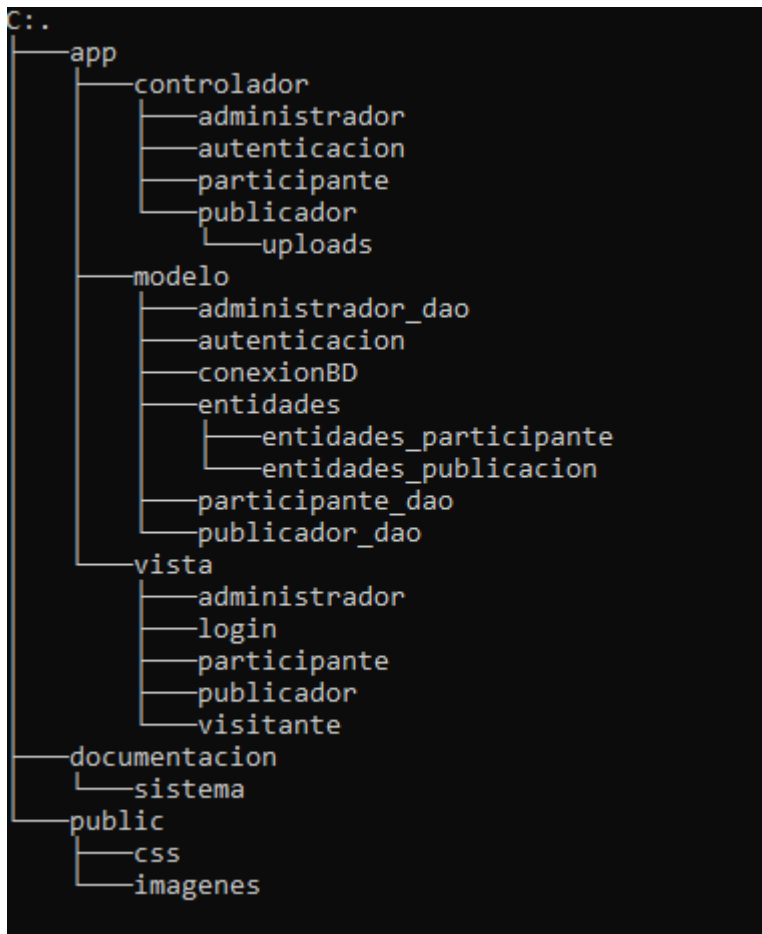
Configuración Inicial: Se deberá colocar todo el repositorio del sistema web en la carpeta 'htdocs' de XAMPP, para luego levantar el servidor ubicado en la carpeta raíz.

7. Seguridad

7.1. Hashing de Contraseñas

Las contraseñas de los usuarios están encriptadas utilizando funciones de hashing en PHP para evitar que se almacenen en texto plano.

8. Arquitectura del Sistema:



Arquitectura: La arquitectura aplicada para este sistema fue un patrón de diseño llamado MVC (Modelo,Vista,Controlador), los archivos del sistema están distribuidos de la siguiente manera, toda la lógica de programación está ubicada en la carpeta app, esta se divide en controlador, aquí se encuentra la lógica que conecta la vista con el modelo, en el modelo se encuentra la lógica para realizar las conexiones a la base de datos, y en la vista se encuentran las interfaces.

RESUMEN DE LAS CLASES PHP UTILIZADAS

1. Controladores (**app/controlador**)

Los controladores son responsables de manejar las solicitudes de los usuarios y coordinar la lógica entre las vistas y los modelos.

- **Administrador**
 - **manejoAprobaciones.php**: Controlador para gestionar la aprobación de publicaciones o reportes realizados por otros usuarios.
 - **manejoReportes.php**: Controlador para administrar y procesar los reportes hechos sobre las publicaciones.
- **Autenticación**
 - **autenticar.php**: Controlador para gestionar el inicio de sesión de los usuarios.

- `CerrarSesion.php`: Controlador para manejar el cierre de sesión.
- `procesar_registro.php`: Controlador que procesa la creación de nuevos usuarios y su registro en el sistema.
- **Participante**
 - `asistirEvento.php`: Controlador que permite a los participantes unirse a un evento.
 - `cancelarEvento.php`: Controlador para cancelar la asistencia de un participante a un evento.
 - `reportarPublicacion.php`: Controlador que permite a los participantes reportar publicaciones inapropiadas o problemáticas.
- **Publicador**
 - `guardar_publicacion.php`: Controlador para guardar nuevas publicaciones creadas por los usuarios publicadores.
 - **uploads**: Carpeta que almacena los archivos subidos por los usuarios, como imágenes, audios y documentos de las publicaciones.

2. Modelos (`app/modelo`)

Los modelos se encargan de interactuar con la base de datos y representan las entidades del sistema.

- **Administrador DAO**
 - `PublicacionesDAO.php`: Modelo para gestionar las publicaciones en el contexto del administrador. Permite consultar y actualizar el estado de las publicaciones.
- **Autenticación**
 - `registroUsuarios.php`: Modelo para gestionar el registro de nuevos usuarios en el sistema.
- **Conexión a la Base de Datos**
 - `Conexion.php`: Clase para establecer y gestionar la conexión con la base de datos.
- **Entidades**
 - **Entidades Participante y Publicación**

- `ElementoPublicacion.php`: Representa un elemento de una publicación (como texto, imagen, audio, etc.).
 - `Publicacion.php`: Representa una publicación en el sistema.
- **Participante DAO**
 - `AsistirEventoDAO.php`: Modelo que permite a los participantes unirse a eventos.
 - `ReportePublicacion.php`: Modelo que gestiona la creación y procesamiento de reportes de publicaciones.
 - `VerEventos.php`: Modelo que permite a los participantes ver los eventos disponibles.
- **Publicador DAO**
 - `CrearPublicacion.php`: Modelo para crear nuevas publicaciones en la base de datos.
 - `ObtenerPublicacion.php`: Modelo para obtener las publicaciones ya creadas y sus detalles.

3. Vistas (`app/vista`)

Las vistas son las interfaces de usuario que presentan la información y permiten la interacción con el sistema.

- **Administrador**
 - `menuAdmin.php`: Menú de navegación para el administrador.
 - `vistaAdministrador.php`: Vista principal de la interfaz de administrador.
 - `vistaAprobacionPublicacion.php`: Vista para aprobar o rechazar publicaciones reportadas.
 - `vistaReportesPublicacion.php`: Vista para revisar y administrar reportes de publicaciones.
- **Login**
 - `login.php`: Vista de inicio de sesión.
 - `registrarUsuario.php`: Formulario para registrar nuevos usuarios.
- **Participante**
 - `menuParticipante.php`: Menú de navegación para los participantes.

- `vistaNotificaciones.php`: Vista que muestra las notificaciones a los participantes.
- `vistaParticipante.php`: Vista principal de la interfaz de participante.
- **Publicador**
 - `menuPublicador.php`: Menú de navegación para los publicadores.
 - `nuevaPublicacion.php`: Formulario para crear nuevas publicaciones.
 - `vistaPublicador.php`: Vista principal de la interfaz de publicador.
- **Visitante**
 - `vistaVisitante.php`: Vista para los usuarios visitantes (usuarios no registrados) que permite explorar el sistema con acceso limitado.