

Proyecto Fin de Ciclo 2023/2024

Social Tweet

18-04-2024

Desarrollo De Aplicaciones Web
Creado por: Iván Martínez Sánchez



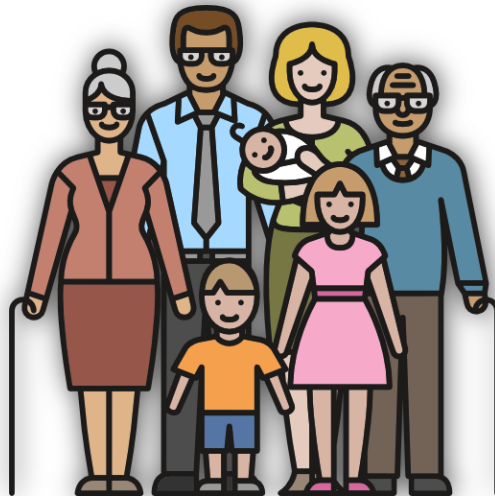
Índice

Índice.....	2
1.Introducción.....	3
A. Presentación del proyecto.....	3
B. Objetivos del proyecto.....	4
C. Justificación del proyecto	5
2. Análisis de requerimientos.....	6
A. Identificación de necesidades y requerimientos.....	6
B. Identificación de público.	7
C. Estudio de mercado y competencia.	7
3. Diseño y planificación.....	9
A. Definición de la arquitectura del proyecto.....	9
B. Diseño de la interfaz de usuario.	10
C. Planificación de las tareas y los recursos necesarios	13
4. Implementación y pruebas.....	15
A. Desarrollo de las funcionalidades del proyecto.	15
B. Pruebas unitarias y de integración.....	18
C. Corrección de errores.....	28
5. Documentación.....	30
A. Documentación técnica.....	30

1.Introducción

A. Presentación del proyecto

- **Social Tweet se trata de una red social en la cual los usuarios podrán subir sus publicaciones, ya sean sobre temas educativos, laborales o simplemente sobre su día a día.**
- **Esta web se realizará para que sea didáctica, fácil de usar y para todas las edades, simple a la vista.**



- **Como ejemplo para desarrollar esta web me he basado en Twitter o X ya que la función de esta web es similar, y mi objetivo sería recrear esta web, de ahí el nombre Social Tweet.**

B. Objetivos del proyecto

- El objetivo de este proyecto es crear una aplicación web similar a Twitter que cuente con la mayoría de las funcionalidades, que los usuarios puedan iniciar sesión y registrarse, puedan subir publicaciones, dar me gusta y guardar otras publicaciones, comentar otras publicaciones, editar su usuario...
- Otro objetivo es aplicar un panel de administrador, que podrán usar los administradores al iniciar sesión, y podrá realizar todas las funciones posibles CRUD, o sea, podrá crear, ver, editar y eliminar usuarios, publicaciones, me gustas, comentarios...

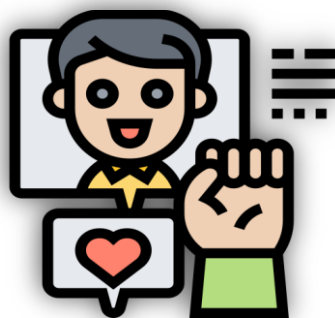


- Otros objetivos también serían, crear una aplicación web fácil de usar y de entender, como comentaba antes (para todas las edades), y que tenga un estilo responsive donde se vea correctamente en cualquier dispositivo.



C. Justificación del proyecto

- Al final me decidí en hacer este proyecto, porque desde siempre me han gustado mucho las redes sociales donde las personas pueden publicar todo tipo de cosas, y para mí es un gran reto crear una red social así de compleja, algo que me motiva y me llena de ganas para realizar este proyecto.



2. Análisis de requerimientos.

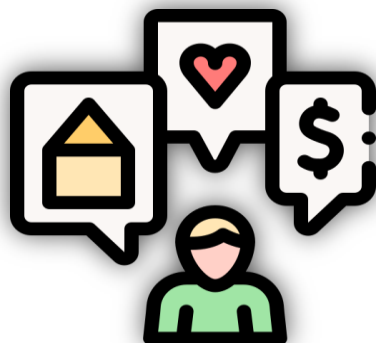
A. Identificación de necesidades y requerimientos.

Necesidades:

- Funcionalidades básicas para realizar publicaciones.
- Gestión de usuarios.
- Seguimiento de usuarios.
- Interacciones entre usuarios.
- Privacidad y seguridad.
- Búsquedas.

Requerimientos:

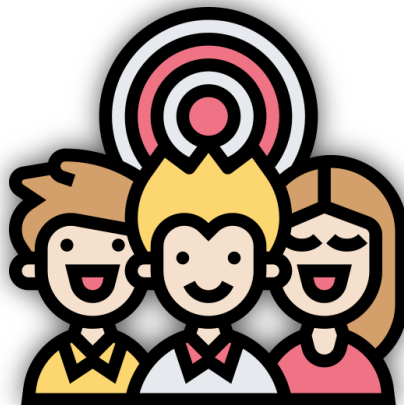
- Sistema de inicio de sesión y registro de usuarios.
- Sistema de creación de publicaciones.
- Sistema de interacción con otras publicaciones.
- Sistema de edición de usuarios.
- Panel de administrador.
- Diseño disponible para todas las plataformas (responsive).



B. Identificación de público.

La página web estará dirigida al público de cualquier ubicación geográfica, para cualquier edad y sexo. También para el público que tenga diferentes pasiones e intereses, ya que en la web se encontrarán publicaciones de diferentes temas.

Esta web busca que haya un comportamiento en línea correcto para todas las edades, sin publicaciones con temas restringidos ni +18.



C. Estudio de mercado y competencia.

- **Nuestros competidores principales serán plataformas que también se dedican a las redes sociales como serían Twitter, Facebook, Instagram o Reddit.**

-
- Las redes sociales están teniendo un gran aumento en el contenido de los usuarios, publicaciones y monetización de estas, y las plataformas están buscando nuevas formas para aumentar la participación y contenido de los usuarios y generar más visitas.



- El objetivo de la plataforma es crear una web que combine todas las mejores características de estas redes sociales, aplicando mejoras notables y una mayor seguridad para nuestros usuarios.
- Mi objetivo es que también se suba contenido de nichos específicos proporcionando así información para distintos grupos de usuarios.

3. Diseño y planificación.

A. Definición de la arquitectura del proyecto.

La arquitectura de este proyecto estará formada por el modelo vista controlador (MVC), ya que es un modelo muy usado actualmente en las aplicaciones web, donde todo se distribuye mejor y más claro.

Como servidor web se utilizará 000webhost donde se pueden alojar webs gratuitamente y me parece un muy buen servicio para alojar mi página web.

Como base de datos donde guardar nuestros datos utilizare MySQL gestionado por phpMyAdmin, ya que 000webhost te permite utilizarlo correctamente.

El lenguaje utilizado será PHP, ya que lo he utilizado este último año en el grado y me ha gustado bastante, ya que puedes utilizarlo tanto para el front-end como para el back-end y me manejo bastante bien con él.

Para la vista del usuario también se utilizará HTML, Bootstrap para aplicar estilos ya que me parece mucho más cómodo, y algo de código JavaScript para que mi web sea dinámica y también para poder usar AJAX y realizar llamadas fetch.



B. Diseño de la interfaz de usuario.

Página de Inicio de Sesión y Registro:

- Cabecera con el logotipo de la plataforma.
- Campos de entrada para correo electrónico/usuario y contraseña para iniciar sesión.
- Botón para "Iniciar Sesión".
- Enlace de "¿No tienes cuenta?" para registrarte donde te mandará a un formulario de registro con todos los campos para crear un usuario.

Página Principal (Después del Inicio de Sesión o Registro):

- **Barra de navegación con el logotipo de la plataforma, un buscador de publicaciones, un botón para agregar una nueva publicación, un enlace para editar el perfil de usuario, un enlace para ver las publicaciones guardadas, un botón para cerrar sesión y, si el usuario es administrador, un enlace para acceder al panel de administración.**
- **Sección de noticias con todas las publicaciones disponibles.**
- **Cada publicación muestra el nombre de usuario, la fecha y hora de la publicación, el contenido de la publicación, el número de me gusta, un botón para dar me gusta, un botón para guardar la publicación, un botón para comentar y, si el usuario es el autor de la publicación, un botón para editar la publicación o eliminarla.**
- **Un campo de texto para escribir un nuevo tweet con un botón para publicar.**
- **Pop up en tiempo real para nuevas interacciones, como me gusta, guardados y comentarios.**

Panel de Administración (Si el Usuario es Administrador):

- Lista de opciones, como "Crear Publicación", "Ver Publicaciones", "Editar Publicaciones", "Eliminar Publicaciones", "Ver Usuarios", "Editar Usuarios" y "Eliminar Usuarios".
- Cada opción lleva a una vista específica donde se pueden realizar las acciones correspondientes.
- La opción "Editar Usuarios" permite al administrador cambiar los roles de los usuarios, como convertirlos en administradores o desactivar sus cuentas.



C. Planificación de las tareas y los recursos necesarios

Diseño de Interfaz de Usuario:

- **Crear la interfaz del usuario para ver su estructura y flujo de la web.**

Desarrollo Front-end:

- **Implementar la interfaz de usuario utilizando tecnologías web como HTML, Bootstrap y JavaScript.**
- **Desarrollar las características de interacción en la interfaz de usuario, como la publicación de tweets, la interacción con publicaciones, la gestión de perfiles de usuario, etc.**

Desarrollo Back-end:

- **Diseñar y desarrollar la lógica del servidor y la base de datos para las funciones de la aplicación.**
- **Implementar la autenticación de usuarios, la gestión de sesiones, el almacenamiento de datos y otras funciones relacionadas con el back-end.**

Pruebas y Depuración:

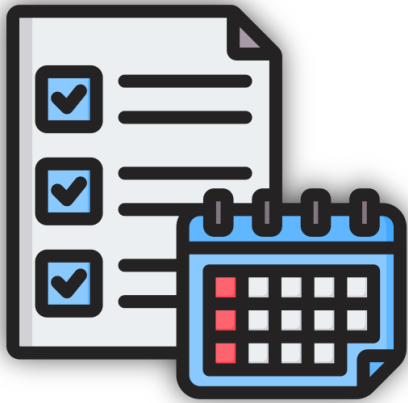
- **Realizar pruebas a la aplicación para corregir los errores y asegurar que todas las funcionalidades funcionen bien en diferentes dispositivos.**

Seguridad:

- Implementar medidas de seguridad para proteger la aplicación contra vulnerabilidades como ataques de inyección SQL, ataques de scripting entre sitios, etc.

Despliegue y Mantenimiento:

- Desplegar la aplicación.
- Realizar mantenimiento continuo de la aplicación, aplicar actualizaciones, corregir errores y añadir nuevas funcionalidades según las necesidades del usuario.



4. Implementación y pruebas.

A. Desarrollo de las funcionalidades del proyecto.

Primero he empezado desarrollando el inicio de sesión y el registro de mi página web, ya que estas dos vistas serán las principales al iniciar la web y serán públicas.

El inicio de sesión tiene validación de usuario y mensajes pop up en caso de error (rojo) o exitoso (verde), el registro es similar, una validación para comprobar que se rellenan los campos obligatorios (*) y comprobar si el nombre de usuario no existe, también cuenta con pop up igual que el inicio de sesión.

Al iniciar sesión o registrarse se guardará el usuario en una variable de sesión, se encriptará la contraseña para mayor seguridad y se creará una cookie de 1 día para recordar nuestro usuario e iniciarnos sesión automáticamente, después nos mandará al inicio.



Después seguí con el inicio donde se mostrarán todos los posts de la web en orden cronológico, estos posts tendrán el nombre de usuario, cuando se publicó, el cuerpo del post y el contador con los me gusta, guardados y mensajes.

Desde aquí el usuario podrá acceder a todos los apartados de la web, podrá acceder a ajustes de usuario o al panel de administrador (si el usuario guardado en la sesión tiene de rol “admin”), podrá acceder al apartado de posts guardados del usuario, utilizar el buscador para buscar posts por el nombre de usuario, crear, editar y eliminar posts.

Cada funcionalidad realizará lo que su nombre indica, el apartado de ajustes de usuario consiste en una vista con un formulario donde el usuario podrá editar sus datos, el panel de administrador consiste en una vista donde nos encontraremos 5 apartados Usuarios, Publicaciones, Me Gustas, Guardados y Mensajes donde podremos realizar un CRUD de estos, es decir Crear, Ver, Editar y Eliminar.



El apartado de guardados nos cargará todas nuestras publicaciones guardadas, con el usuario que ha iniciado sesión.

El buscador de usuarios consiste en un input donde introduciremos caracteres y automáticamente buscare las publicaciones de ese usuario mediante AJAX y podremos verlas, el logout eliminara la sesión actual y la cookie de sesión redirigiéndonos al inicio de sesión de nuevo.

Al añadir o editar una publicación nos mostrara una vista con un formulario donde podremos editar (si es nuestro) o añadir un post.

Debajo de cada publicación encontramos 3 iconos, el de me gusta, guardado y mensaje, al pinchar sobre el icono de me gusta o guardado se incrementará el contador y cambiará de color el icono para indicarnos que lo hemos pulsado, todo esto mediante AJAX, y al pulsar el botón de mensaje nos mostrará una vista con todos los mensajes de ese post y podremos crear un nuevo mensaje para el post.



B. Pruebas unitarias y de integración.

Para poder ver que mi aplicación web funciona correctamente realizaré pruebas unitarias, algunas de estas pruebas serian, comprobar que funciona correctamente la validación de formularios tanto de inicio de sesión como de registro, comprobar que no se puede acceder a otras vistas sin haber iniciado sesión anteriormente, comprobar que no se pueden realizar acciones no permitidas con nuestros permisos, validación al crear publicaciones, mensajes de alerta ...

Primero realicé pruebas unitarias del formulario de inicio de sesión para comprobar que este funciona correctamente, primero sin datos de entrada, mostrando una alerta ya que está vacío.



Después probé el formulario, pero sin el nombre de usuario y con contraseña y otra prueba sin la contraseña, pero con el nombre de usuario y los dos muestran una alerta.



Nombre de usuario o contraseña incorrectos



SocialTweet

Iniciar Sesión

[No tienes cuenta? Registrate](#)



Nombre de usuario o contraseña incorrectos




SocialTweet

Iniciar Sesión


[No tienes cuenta? Registrate](#)

Después inicié sesión con un usuario correcto para ver si lo valida correctamente y lanza una alerta de confirmación.



SocialTweet  Ajustes Gu Inicio de sesión con éxito Logout

 @imartinez29

 @raul

2 días atrás

holaa (Editado)

 0  0

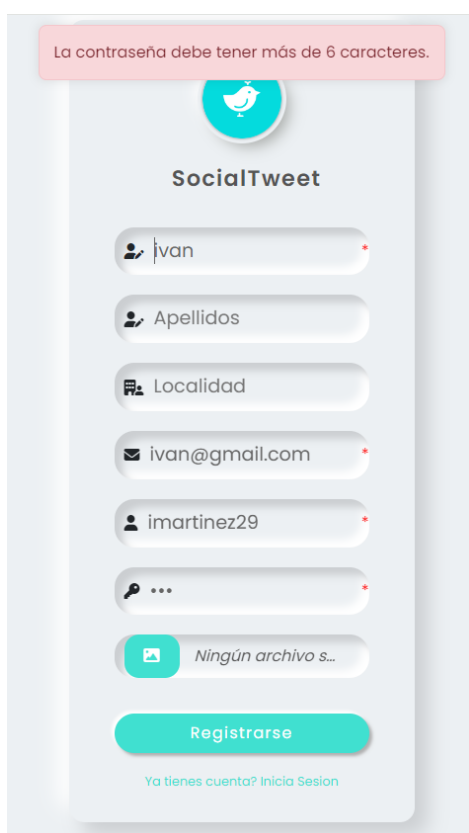
Posteriormente realicé pruebas unitarias para el formulario de registro y comprobar así su validación, probé siempre con un campo vacío obligatorio para comprobar que me lanza una alerta.

The image displays five screenshots of the SocialTweet registration form, each illustrating a different validation error triggered by leaving a required field empty. The form includes fields for Name, Surnames, Location, Email, Username, Password, and a profile picture upload. A 'Registrarse' button and a link to 'Inicia Sesión' are at the bottom of each form.

- Screenshot 1:** Alert: "El campo nombre es obligatorio." (The name field is required).
- Screenshot 2:** Alert: "El campo email es obligatorio." (The email field is required).
- Screenshot 3:** Alert: "El campo nombre de usuario es obligatorio." (The username field is required).
- Screenshot 4:** Alert: "El campo contraseña es obligatorio." (The password field is required).
- Screenshot 5:** Alert: "La foto no tiene el formato admitido, debe ser png, jpg o webp" (The photo does not have the accepted format, it must be png, jpg, or webp).

La página nos mandará una alerta de error al dejar un campo obligatorio vacío y los valores que si se insertan se mantendrán intactos en el formulario.

Después realicé otras pruebas, con una contraseña de menos de 7 caracteres y también con un nombre de usuario ya existente en la aplicación (base de datos).

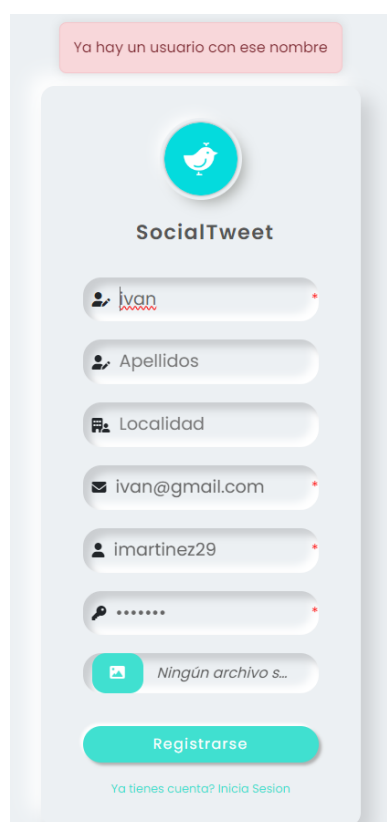


La contraseña debe tener más de 6 caracteres.

SocialTweet

Registrarse

[Ya tienes cuenta? Inicia Sesión](#)



Ya hay un usuario con ese nombre

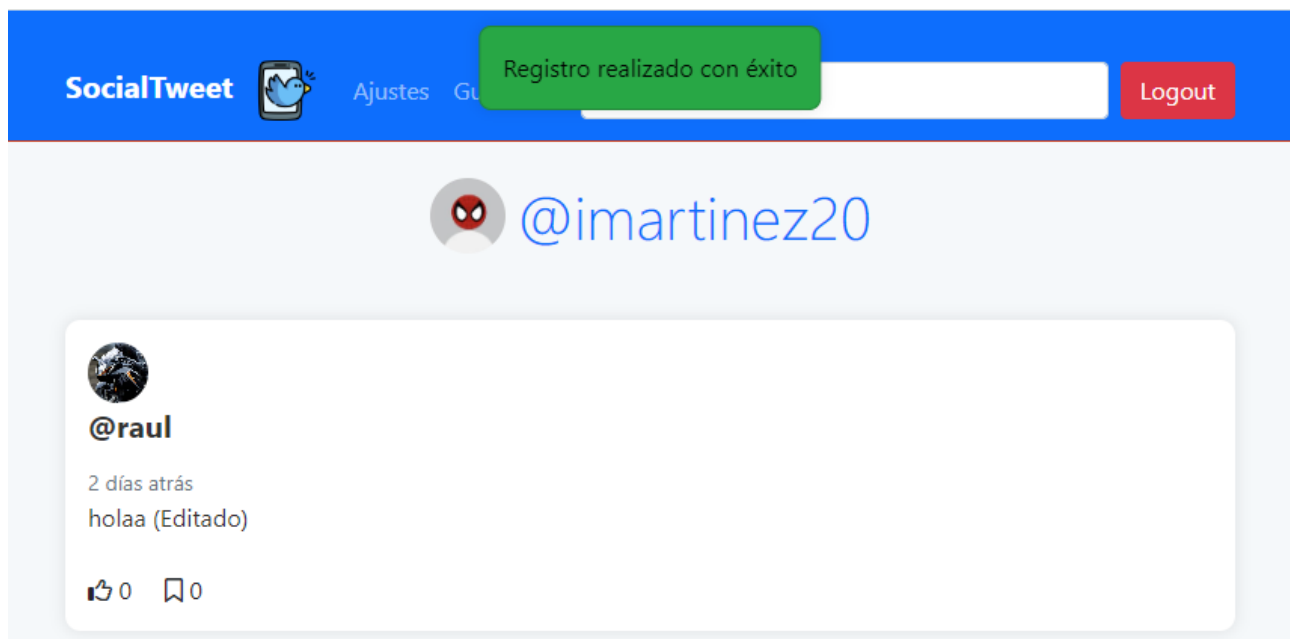
SocialTweet

Registrarse

[Ya tienes cuenta? Inicia Sesión](#)

La página nos alertará en los dos casos, ya que la contraseña debe tener más de 6 caracteres y el nombre de usuario es único, solo puede existir uno igual en la base de datos.

Después realicé un registro con un usuario correcto para ver si lo valida correctamente y lanza una alerta de confirmación.

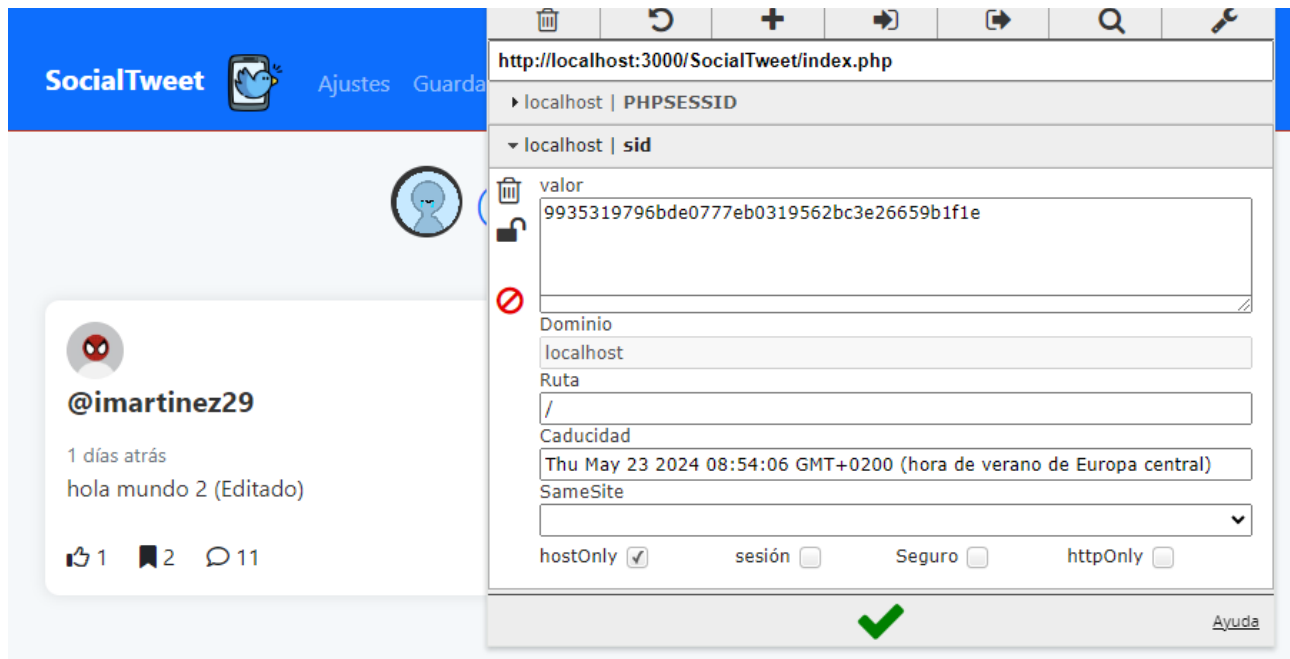


Comprobamos también como al realizar el registro nos crea un usuario en la base de datos con un hash de la imagen y la contraseña encriptada para una mayor seguridad.

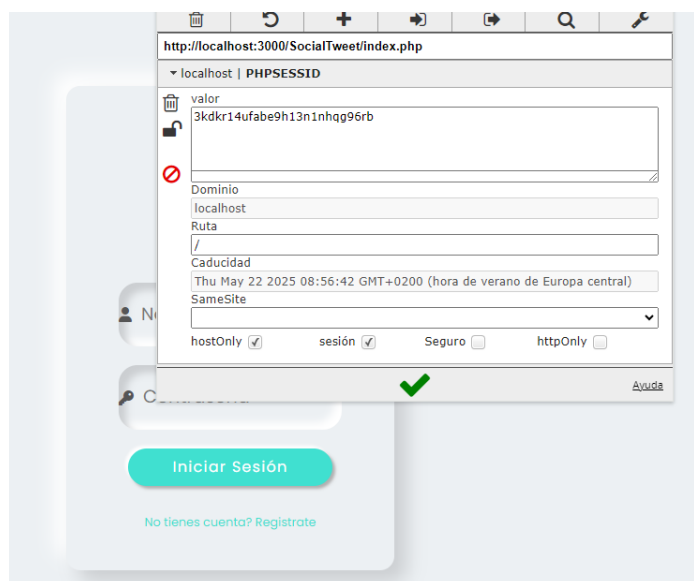
localidad	nombre	nombreusuario	password	foto	rol
ivan	imartinez20		\$2y\$10\$WCB9JryhQAzEE.MncxyC5.M/SpqclVRHmdOYzumX4hl...	a1015b28e313a81a58a48834cb65509b.jfif	NULL

Al realizar un inicio de sesión o registro este usuario es guardado en la sesión y crea una cookie (sid) durante 1 día para que nos inicie sesión automáticamente, para esto también he realizado pruebas para comprobar que funciona correctamente.

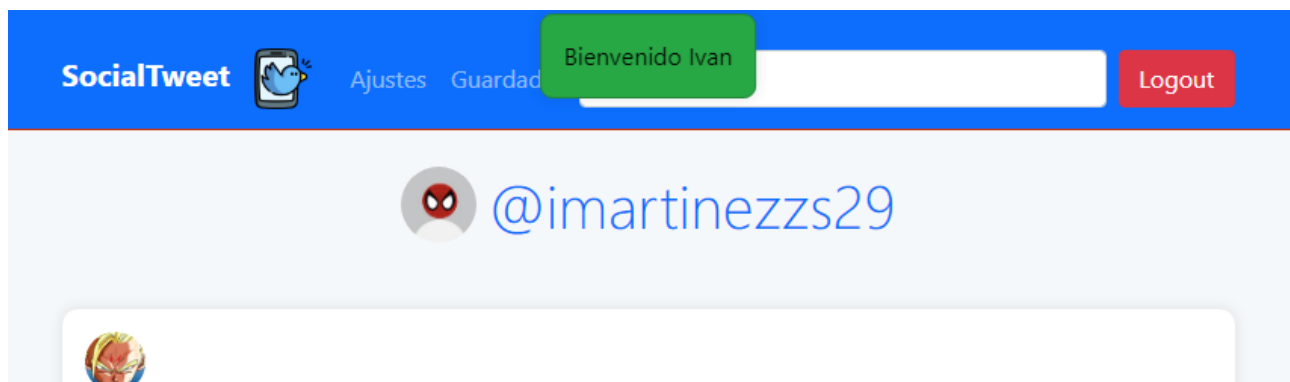
Primero he comprobado que al iniciar sesión o registrarse crea esta cookie para nosotros, esto lo comprobamos con la extensión EditThisCookie.



Podemos también comprobar como al cerrar sesión se elimina esta cookie.



La función de esta cookie es iniciarnos sesión con nuestro usuario en el caso de que cerremos nuestro navegador y queramos acceder otro día, gracias a esto no necesitaremos introducir nuestras credenciales para iniciar sesión, si cerramos el navegador y volvemos a acceder a la web nos mostrará esta alerta dándonos la bienvenida.

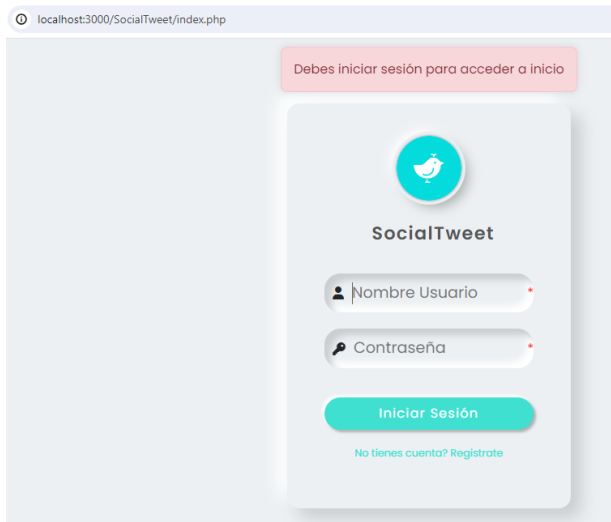


Otra de las pruebas unitarias que he realizado es comprobar la validación del formulario de publicaciones primero con el campo mensaje vacío y luego con el campo mensaje con texto.

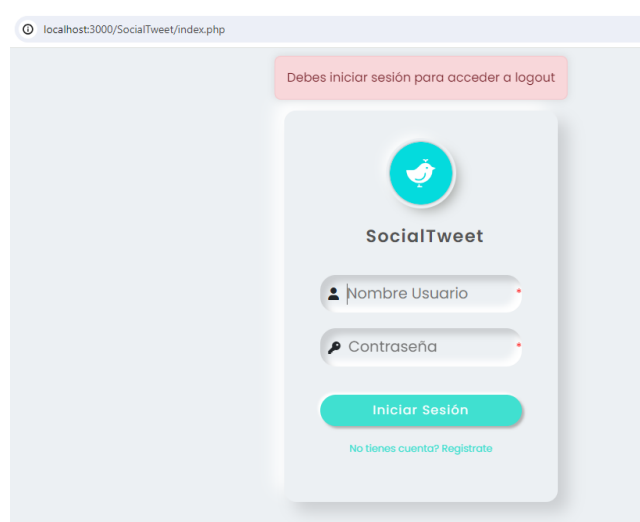


He hecho pruebas para comprobar que no me deja acceder a las rutas sin iniciar sesión, las rutas de registro e inicio de sesión después de iniciar sesión y realizar acciones que no se permiten.

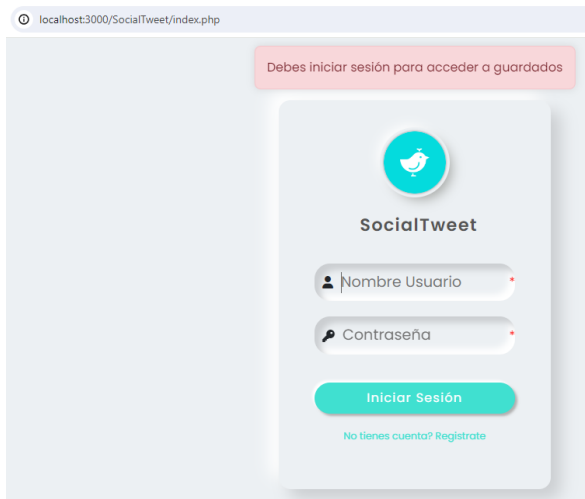
Acceso a inicio sin iniciar sesión



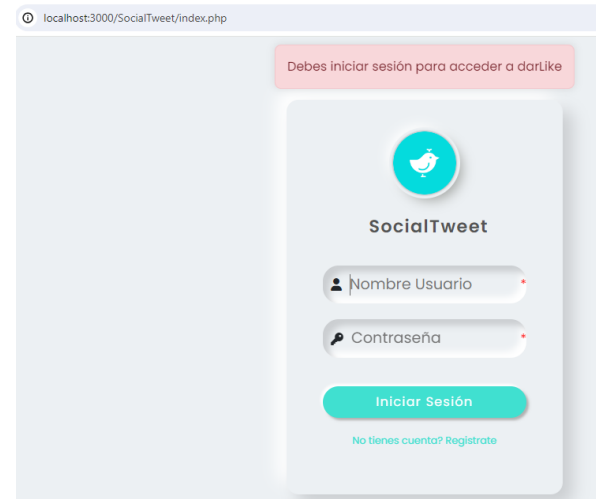
Acceso a logout sin iniciar sesión



Acceso a guardados

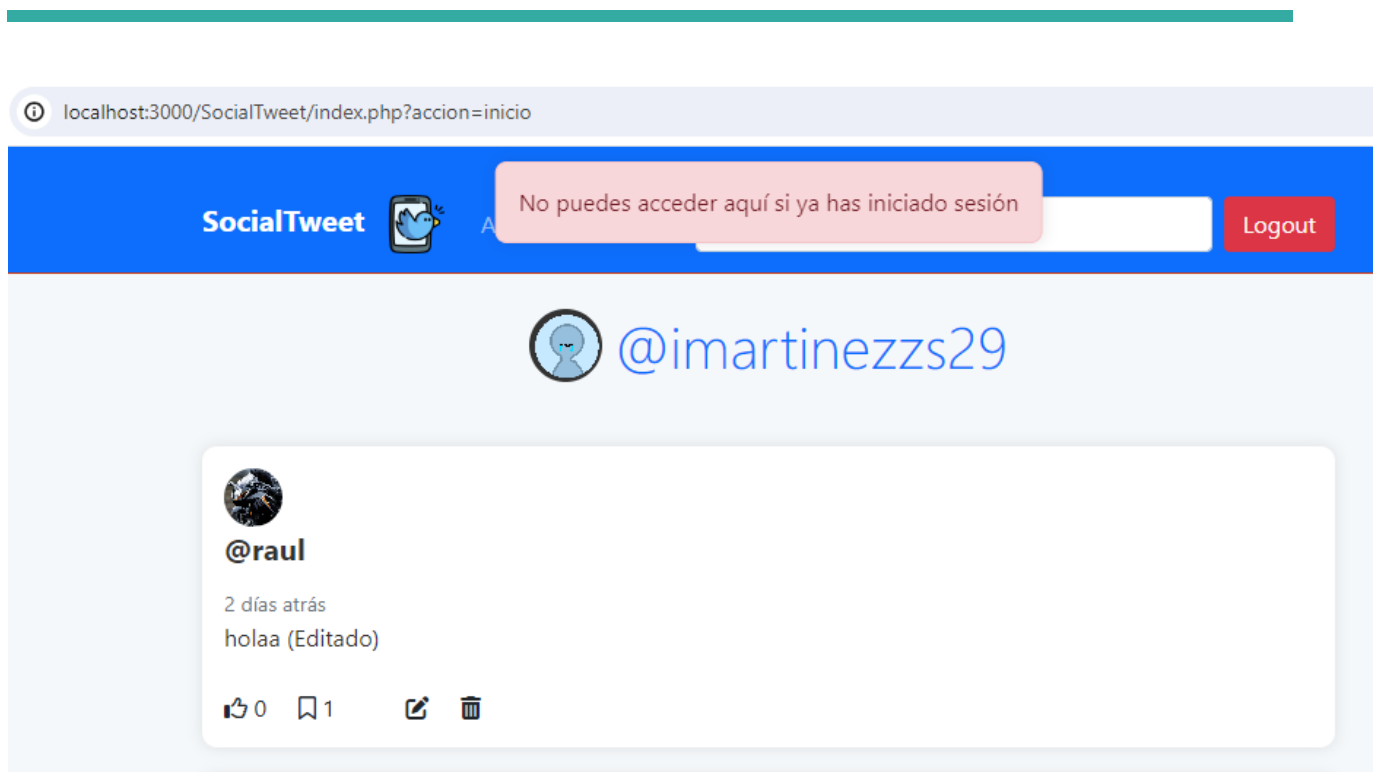


Acceso a acción darLike



Estas serían las pruebas sin haber iniciado sesión.
Ahora las pruebas habiendo iniciado sesión.

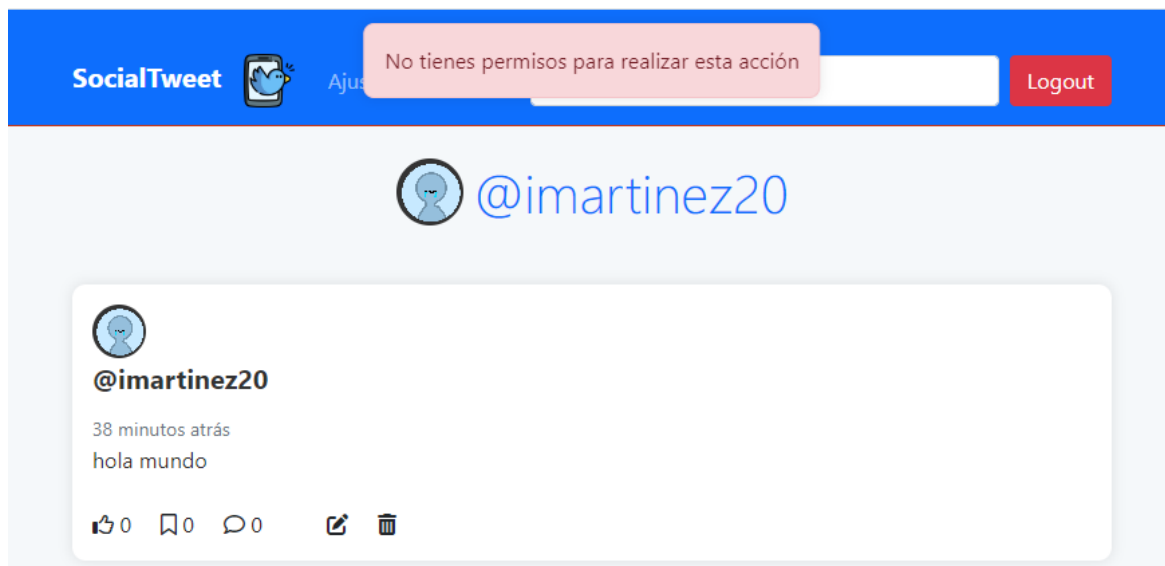
Acceso a inicio de sesión después de haber iniciado sesión



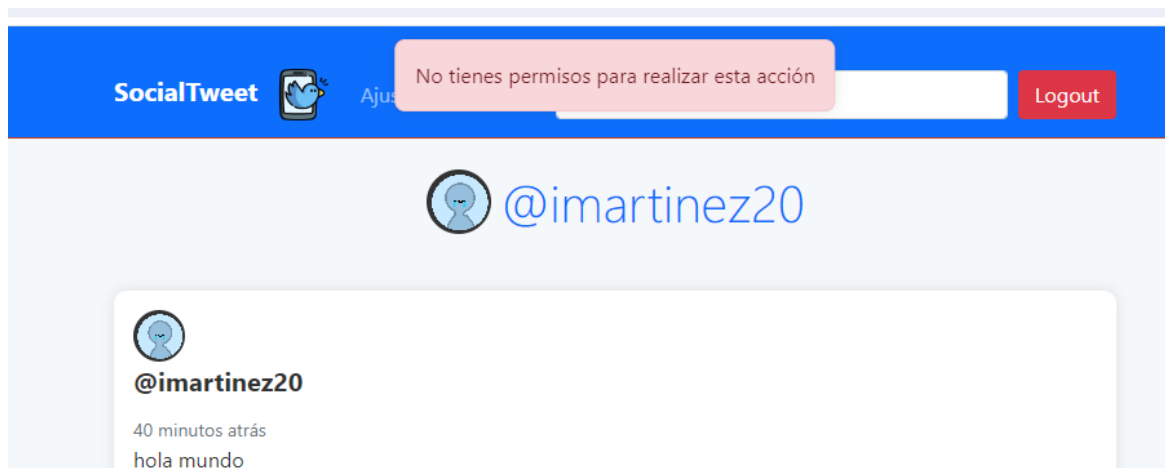
Acceso a registro después de haber iniciado sesión



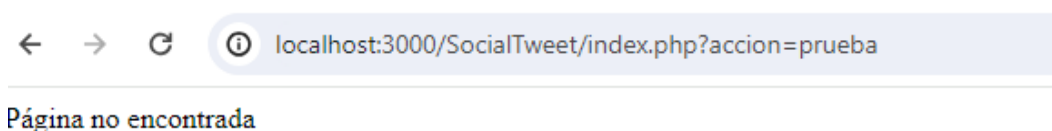
Si intentamos editar una publicación que no es nuestra



Si intentamos borrar una publicación que no es nuestra



Si intentamos acceder a una ruta que no existe



C. Corrección de errores.

Los errores que he tenido mientras desarrollaba la web han sido tres, error de cookie al registrarme, ya que no la guardaba correctamente, error de eventos en las publicaciones cargadas con el buscador input y al editar un mensaje y añadir (editado) al editar de nuevo aparecía el texto (editado).

Para solucionar el tema de las cookies he aplicado el siguiente código en el método de registro.

```
if ($iduser = $usuariosDAO->insert($usuario)) {  
    // Iniciamos sesión  
    $usuario->setIdusuario($iduser);  
    Sesion::iniciarSesion($usuario);  
    setcookie('sid', $usuario->getSid(), time() + 24 * 60 * 60, '/');  
  
    guardarMensajeC("Registro realizado con éxito");  
    header("location: index.php");  
    die();  
}
```

Para solucionar el tema de edición de publicación cambio el texto (Editado) por "" para que así no me aparezca al editar el mensaje.

```
if($id != 0){  
    $laPublicacion = $publicacionDAO->getById($id);  
    $texto = $laPublicacion->getMensaje();  
    $laPublicacion->setMensaje(str_replace(" (Editado)", "", $texto));  
    $form = "Editar";  
}
```

Para solucionar los eventos he tenido que añadirlos mediante JS con DOM (Document Object Model) con onclick en los iconos de me Gusta, Guardado y Mensaje.

```
// Icono de me gusta
const likeIcon = document.createElement("i");
likeIcon.className = publicacion.usuarioHaDadoMeGusta
    ? "fa-solid fa-thumbs-up"
    : "fa-regular fa-thumbs-up";
likeIcon.onclick = function () {
    darLike(publicacion.idPublicacion, data.usuarioActual.id, event);
};

// Contador de me gustas
const likeCount = document.createElement("span");
likeCount.style.display = "inline";
likeCount.textContent = publicacion.megustas;

// Icono de guardar
const bookmarkIcon = document.createElement("i");
bookmarkIcon.className = publicacion.usuarioHaGuardado
    ? "fa-solid fa-bookmark"
    : "fa-regular fa-bookmark";
bookmarkIcon.onclick = function () {
    darGuardado(publicacion.idPublicacion, data.usuarioActual.id, event);
};

// Contador de guardados
const bookmarkCount = document.createElement("span");
bookmarkCount.style.display = "inline";
bookmarkCount.textContent = publicacion.guardados;

// Icono de chat
const chatIcon = document.createElement("i");
chatIcon.className = "fa-regular fa-comment";
chatIcon.onclick = function () {
    openChatModal(publicacion.idPublicacion, event);
};
```

5. Documentación.

A. Documentación técnica.

-Estructura de archivos:

- | — **app/**
 - | | — **config/**
 - | | | — **config.php**: Contiene los parámetros de configuración de la base de datos.
 - | | — **controller/**
 - | | | Controlador para gestionar las funcionalidades relacionadas con los guardados.
 - | | | — **ControladorGuardados.php**:
Controlador para gestionar las funcionalidades relacionadas con los meGustas.
 - | | | — **ControladorMeGustas.php**:
Controlador para gestionar las funcionalidades relacionadas con los mensajes.
 - | | | — **ControladorMensajes.php**:
Controlador para gestionar las funcionalidades relacionadas con las publicaciones.
 - | | | — **ControladorPublicaciones.php**:
Controlador para gestionar las funcionalidades relacionadas con los usuarios.
 - | | | — **ControladorUsuarios.php**:
 - | | — **model/**
 - | | | — **ConexionDBi.php**: Archivo que maneja la conexión a la base de datos
 - | | | — **Guardado.php**: Modelo que representa un guardado en el sistema.
 - | | | — **GuardadoDAO.php**: DAO para manejar las operaciones CRUD de los guardados.
 - | | | — **MeGusta.php**: Modelo que representa un meGusta en el sistema.
 - | | | — **MeGustaDAO.php**: DAO para manejar las operaciones CRUD de los "me gusta".
 - | | | — **Mensaje.php**: Modelo que representa un mensaje en el sistema.
 - | | | — **MensajeDAO.php**: DAO para manejar las operaciones CRUD de los mensajes.
 - | | | — **Publicacion.php**: Modelo que representa una publicación en el sistema.
 - | | | — **PublicacionDAO.php**: DAO para manejar las operaciones CRUD de las publicaciones.
 - | | | — **Sesion.php**: Maneja las sesiones de usuario.
 - | | | — **Usuario.php**: Modelo que representa un usuario en el sistema.
 - | | | — **UsuarioDAO.php**: DAO para manejar las operaciones CRUD de los usuarios.
 - | | — **utils/**
 - | | | — **utils.php**: Archivo de utilidades que contiene funciones auxiliares.
 - | | — **views/**
 - | | | — **admin.php**: Vista para el panel de administración del sitio.
 - | | | — **ajustes.php**: Vista para la página de ajustes del usuario (editar).
 - | | | — **inicio.php**: Vista para la página de inicio o feed de publicaciones.
 - | | | — **login.php**: Vista para la página de inicio de sesión.
 - | | | — **mensaje.php**: Vista para (crear/editar) mensaje.

- | | └─ **publicacion.php**: Vista para (crear/editar) publicación.
- | | └─ **registro.php**: Vista para la página de registro de nuevos usuarios.
- | | └─ **sobreMi.php**: Vista para la página de información sobre mí.
- | | └─ **usuario.php**: Vista para (crear/editar) usuario.

- └─ **web/**
 - | └─ **fotosPublicaciones/**: Directorio para almacenar las fotos de las publicaciones.
 - | └─ **fotosUsuarios/**: Directorio para almacenar las fotos de perfil de los usuarios.
 - | └─ **images/**: Directorio para almacenar imágenes generales usadas en la aplicación (logotipos, íconos, etc.).
 - | └─ **js/**
 - | | └─ **ajax.js**: Archivo JavaScript que maneja las solicitudes AJAX.
 - | └─ **styles/**
 - | | └─ **Inicio.css**: Archivo CSS que contiene los estilos para la página de inicio.
 - | | └─ **LoginRegistro.css**: Archivo CSS que contiene los estilos para el login y registro.

└─ **index.php**: Archivo principal, se encarga de inicializar la aplicación, manejar las solicitudes y cargar las vistas correspondientes.

-Herramientas utilizadas:

Para el desarrollo del proyecto se ha utilizado Visual Studio Code tanto para el frontend como para el backend, ya que me siento bastante cómodo con este entorno y lo aprendí a utilizar durante el curso de DAW.



Para poder utilizar una base de datos en nuestra aplicación he usado XAMPP, el cual incluye PHPMyAdmin, y con este podremos utilizar una base de datos SQL en nuestra aplicación web, también podemos desplegar nuestra aplicación web en local con XAMPP ya que incluye Apache.



Para poder alojar mi web en un host y poder acceder a ella he utilizado 000webhost ya que me parece muy cómodo y fácil de utilizar, además es gratis y te permite utilizar PHPMyAdmin para alojar también la base de datos de tu aplicación.



La tecnología principal que he utilizado para mi proyecto es PHP, utilizada para el backend ya que se ejecuta en el lado del servidor.

Para el frontend he utilizado otras tecnologías las cuales son, HTML para construir el esqueleto de las vistas, CSS para aplicar estilos a las vistas, Bootstrap que es un framework para diseño de aplicaciones web, JavaScript para poder crear una página web dinámica e interactiva, JQuery que es una librería de JavaScript y font awesome para poder utilizar algunos iconos para mi aplicación web.



B. Documentación de usuario.