



ASAMBLEA DE VOTACIONES GUSTAVO ADOLFO OSORIO

- JHON ANDERSON LUNA
- JHONATAN ANDRES CUERO CHIRINO
- JUAN ESTEBAN PEREIRA
- MIGUEL ANGEL PINEDA
- ALEJANDRO GONZALEZ SILVA

DOCUMENTO DEL PROYECTO

El sistema de votaciones para asambleas tiene como objetivo principal facilitar el proceso de votación y gestión de propuestas para reuniones y asambleas. Los usuarios podrán registrarse, iniciar sesión, crear, leer, actualizar y eliminar propuestas, votar en propuestas existentes, visualizar los resultados de las votaciones y buscar propuestas.

PROTOTIPO DE INTERFAZ DE USUARIO

Diseño visual de las páginas web del proyecto:

- **Página de bienvenida.**
- **Página de inicio de sesión.**
- **Página de registro de usuario.**
- **Página de administrador.**
- **Página de propuestas.**
- **Página de votaciones.**
- **Página de búsqueda de propuestas.**
- **Página de actualización de datos.**

MODO DE TRABAJO

- Lenguajes de programación: HTML, CSS y JavaScript (frontend), PHP (backend).
- Plataforma de alojamiento: XAMPP.
- Base de datos: MySQL con los servidores de XAMPP.
- Herramientas de desarrollo: GitHub para el control de versiones.

PRODUCT BACKLOG DEL PROYECTO

1. REGISTRO DE USUARIOS

Historia de Usuario: Yo como usuario quiero poder registrarme al iniciar sesión, con los siguientes datos (id, nombres, apellidos, edad, cargo, correo, usuario, contraseña).

Tareas:

- Diseñar la interfaz de registro para el usuario.
- Implementar el método para capturar datos (id, nombres, apellidos, edad, cargo, correo, usuario, contraseña).
- Realizar métodos de validación para cada campo (ID y Username únicos).
- Implementar el método para guardar los datos en la base de datos.

ASAMBLEA DE VOTACIONES

GUSTAVO ADOLFO OSORIO

2. INICIO DE SESIÓN

Historia de Usuario: Yo como usuario quiero que al iniciar el programa tenga la opción de iniciar sesión una vez ya registrado con mi nombre de usuario y contraseña.

Tareas:

- Diseñar la interfaz del login.
- Implementar el método para capturar datos (usuario y contraseña).
- Validar si el usuario y la contraseña existen en la base de datos y dar acceso al programa.

3. CREACIÓN DE ASAMBLEAS

Historia de Usuario: Yo administrador quiero poder crear asambleas nuevas para que los usuarios creen sus temas y propuestas.

Tareas:

- Módulo que permita crear asambleas en la sesión de administrador.
- Módulo que permita cambiar el estado de asamblea activa/cerrada.
- Módulo que permita eliminar asambleas.
- Validar que los campos no estén vacíos.
- Guardar los datos de la asamblea en la base de datos.

4. CREACIÓN DE TEMAS

Historia de Usuario: Yo como usuario quiero poder crear un tema nuevo para añadir propuestas con respecto a ese tema.

Tareas:

- Implementar un módulo que permita crear temas dentro de la Asamblea activa.
- Validar que el usuario tenga los permisos necesarios para crear un tema.
- Guardar los datos del tema en la base de datos.

5. CREACIÓN DE PROPUESTAS

Historia de Usuario: Quiero poder crear mis propuestas en un tema dentro de la asamblea con un "Título" y una "Descripción".

Tareas:

- Diseñar la interfaz para crear propuestas.
- Implementar el método para capturar título y descripción de la propuesta.
- Validar que los campos no estén vacíos.
- Guardar la propuesta en la base de datos asociada al tema y usuario correspondientes.

ASAMBLEA DE VOTACIONES

GUSTAVO ADOLFO OSORIO

6. VISUALIZACIÓN DE PROPUESTAS

Historia de Usuario: Las propuestas que yo cree las quiero visualizar dentro de un cuadro donde estén de manera ordenada.

Tareas:

- Diseñar la interfaz para listar las propuestas del usuario.
- Implementar el método para recuperar y mostrar las propuestas desde la base de datos.

7. EDICIÓN Y ELIMINACIÓN DE PROPUESTAS

Historia de Usuario: Las propuestas que yo cree quiero poder actualizarlas si tengo que hacer algún cambio y también quiero poder eliminar una propuesta mía.

Tareas:

- Implementar el método para editar título y descripción de una propuesta.
- Validar que la propuesta tenga 0 votos para permitir la modificación.
- Implementar el método para eliminar una propuesta.

8. LISTADO DE PROPUESTAS

Historia de Usuario: Yo como usuario quiero que los demás usuarios puedan ver un listado con todas las propuestas creadas en las que se incluya el título, descripción de la propuesta, el usuario que la creó y el número de votos de esta.

Tareas:

- Diseñar la interfaz para listar todas las propuestas.
- Implementar el método para recuperar y mostrar las propuestas desde la base de datos.

9. VOTACIÓN DE PROPUESTAS

Historia de Usuario: Yo como usuario quiero poder votar en cualquiera de las propuestas que existan en los diferentes temas de una Asamblea. Además yo como usuario no debo poder votar por propuestas que yo mismo cree.

Tareas:

- Implementar el método para votar por propuestas.
- Validar que el usuario no vote por sus propias propuestas.
- Actualizar la base de datos con el voto del usuario.

ASAMBLEA DE VOTACIONES

GUSTAVO ADOLFO OSORIO

10. VISUALIZACIÓN DE RESULTADOS DE VOTACIÓN

Historia de Usuario: Yo como usuario quiero que se vean los resultados de las votaciones para cada propuesta.

Tareas:

- Diseñar la interfaz para mostrar los resultados de las votaciones.
- Implementar el método para recuperar y mostrar los resultados desde la base de datos.

11. FINALIZACIÓN DE ASAMBLEAS

Historia de Usuario: Quiero que al finalizar las asambleas se muestre el número total de los votos en las propuestas y la propuesta ganadora en la asamblea.

Tareas:

- Implementar el método para calcular y mostrar el total de votos.
- Mostrar la propuesta ganadora de la asamblea.

12. BÚSQUEDA DE PROPUESTAS

Historia de Usuario: Yo como usuario quiero que se puedan buscar propuestas por título, descripción o nombre del creador de la propuesta.

Tareas:

- Implementar el método de búsqueda por diferentes filtros.
- Diseñar la interfaz para la búsqueda de propuestas.
- Mostrar los resultados de la búsqueda.

13. GESTIÓN DE DATOS PERSONALES

Historia de Usuario: Yo como usuario quiero poder actualizar mis datos personales.

Tareas:

- Diseñar la interfaz para la actualización de datos personales.
- Implementar el método para capturar y validar los datos actualizados.
- Guardar los cambios en la base de datos.

DEFINITION OF READY (DOR)

1. REGISTRO DE USUARIOS

- La base de datos debe estar implementada para almacenar la información de los usuarios.
- Los requisitos de la interfaz de usuario para la página de registro deben estar definidos, incluyendo campos obligatorios, validación de datos y mensajes de error.
- Deben definirse las acciones y mensajes que se mostrarán al usuario en caso de éxito o fallo del registro debido a la falta de datos o duplicación de información.

ASAMBLEA DE VOTACIONES

GUSTAVO ADOLFO OSORIO

2. INICIO DE SESIÓN DE USUARIOS

- Debe haber registros previos de usuarios para validar las credenciales.
- La base de datos debe estar implementada para verificar usuario y contraseña.
- Los requisitos de la interfaz de usuario para la página de inicio de sesión deben estar definidos, incluyendo campos obligatorios, validación de datos y mensajes de error en caso de credenciales incorrectas.

3. CREACIÓN DE ASAMBLEAS

- Debe estar definida la interfaz del administrador para la creación de nuevas asambleas.
- La funcionalidad de inicio de sesión debe estar completa para garantizar que solo el administrador pueda crear asambleas.
- Los permisos de administrador para crear asambleas deben estar definidos.

4. EDICIÓN Y ELIMINACIÓN DE ASAMBLEAS

- La interfaz de usuario para editar y eliminar asambleas debe estar definida.
- La funcionalidad de creación de asamblea debe estar completa para que haya asambleas que editar o eliminar.
- La validación de estado para eliminar debe estar definida.
- La funcionalidad de editar estado de asamblea debe estar definida.

5. CREACIÓN DE TEMAS

- Debe estar definida la interfaz de usuario para la creación de nuevos temas, incluyendo campos obligatorios y validaciones.
- La funcionalidad de inicio de sesión debe estar completa para garantizar que solo usuarios autenticados puedan crear temas.
- Los permisos de usuario para la creación de temas deben estar definidos.

6. CREACIÓN DE PROPUESTAS

- La interfaz de usuario para la creación de propuestas debe estar definida, incluyendo campos obligatorios y validaciones.
- La funcionalidad de inicio de sesión debe estar completa para garantizar que solo usuarios autenticados puedan crear propuestas.
- Los permisos de usuario para la creación de propuestas deben estar definidos.

7. VISUALIZACIÓN DE PROPUESTAS

- La interfaz de usuario para listar las propuestas debe estar definida.
- La funcionalidad de creación de propuestas debe estar completa para que haya propuestas que mostrar.

ASAMBLEA DE VOTACIONES

GUSTAVO ADOLFO OSORIO

8. EDICIÓN Y ELIMINACIÓN DE PROPUESTAS

- La interfaz de usuario para editar y eliminar propuestas debe estar definida.
- La funcionalidad de creación de propuestas debe estar completa para que haya propuestas que editar o eliminar.

9. LISTADO DE PROPUESTAS

- La interfaz de usuario para listar todas las propuestas debe estar definida.
- La funcionalidad de creación de propuestas debe estar completa para que haya propuestas que listar.

10. VOTACIÓN DE PROPUESTAS

- La funcionalidad de creación de propuestas debe estar completa para que haya propuestas en las cuales votar.
- La interfaz de usuario para votar debe estar definida, incluyendo restricciones para que los usuarios no puedan votar por sus propias propuestas.

11. VISUALIZACIÓN DE RESULTADOS DE VOTACIÓN

- La funcionalidad de votación debe estar completa para que existan resultados que mostrar.
- La interfaz de usuario para visualizar los resultados debe estar definida, incluyendo criterios para determinar el ganador.

12. FINALIZACIÓN DE ASAMBLAS

- La funcionalidad de votación debe estar completa para que existan resultados al finalizar la asamblea.
- La interfaz de usuario para mostrar los resultados finales debe estar definida.

13. BÚSQUEDA DE PROPUESTAS

- Los campos de búsqueda deben estar claramente definidos (título, descripción, nombre del creador).
- La interfaz de usuario para la búsqueda debe estar definida, incluyendo opciones adicionales como filtros.

14. GESTIÓN DE DATOS PERSONALES

- La interfaz de usuario para la actualización de datos personales debe estar definida.
- Los métodos para capturar y validar los datos actualizados deben estar implementados.

DEFINITION OF DONE (DOD)

1. REGISTRO DE USUARIOS

- Todos los campos obligatorios del formulario de registro (nombre, apellido, edad, cargo, correo, usuario, contraseña) deben estar presentes y ser editables.
- El sistema debe validar los datos ingresados por el usuario según los requisitos especificados.
- Debe implementarse la verificación de unicidad del nombre de usuario y la dirección de correo electrónico para evitar registros duplicados.
- Proporcionar retroalimentación clara al usuario en caso de errores durante el registro.
- Permitir la creación de usuarios de tipo moderador y miembro.

2. INICIO DE SESIÓN DE USUARIOS

- Implementar la interfaz de inicio de sesión.
- Verificar las credenciales del usuario y, si son válidas, proporcionar acceso a las funcionalidades correspondientes.

3. CREACIÓN DE ASAMBLEAS

- Implementar un módulo que permita crear asambleas.
- Validar los permisos de administrador para crear un tema.
- Guardar los datos de la asamblea en la base de datos.

4. EDICIÓN Y ELIMINACIÓN DE ASAMBLEAS

- Implementar la funcionalidad para cambiar estado de una asamblea.
- Implementar la funcionalidad para eliminar una asamblea.
- Validar que la asamblea este en estado “nueva” para permitir la eliminación.

5. CREACIÓN DE TEMAS

- Implementar un módulo que permita crear temas dentro de la Asamblea activa.
- Validar los permisos del usuario para crear un tema.
- Guardar los datos del tema en la base de datos.

6. CREACIÓN DE PROPUESTAS

- Implementar la interfaz para crear propuestas.
- Validar que los campos de título y descripción no estén vacíos.
- Guardar la propuesta en la base de datos asociada al tema y usuario correspondientes.

7. VISUALIZACIÓN DE PROPUESTAS

- Implementar la interfaz para listar las propuestas del usuario.
- Mostrar las propuestas desde la base de datos en un cuadro de manera ordenada.

8. EDICIÓN Y ELIMINACIÓN DE PROPUESTAS

- Implementar la funcionalidad para editar título y descripción de una propuesta.
- Validar que la propuesta tenga 0 votos para permitir la modificación.
- Implementar la funcionalidad para eliminar una propuesta.

9. LISTADO DE PROPUESTAS

- Implementar la interfaz para listar todas las propuestas.
- Mostrar todas las propuestas desde la base de datos, incluyendo título, descripción, usuario creador y número de votos.

10. VOTACIÓN DE PROPUESTAS

- Implementar la funcionalidad de votación para cada propuesta.
- Validar que el usuario no vote por sus propias propuestas.
- Actualizar la base de datos con el voto del usuario.

11. VISUALIZACIÓN DE RESULTADOS DE VOTACIÓN

- Implementar la interfaz para mostrar los resultados de las votaciones.
- Mostrar el número total de votos y la propuesta ganadora de la asamblea.

12. FINALIZACIÓN DE ASAMBLEAS

- Calcular y mostrar el total de votos al finalizar la asamblea.
- Mostrar la propuesta ganadora de la asamblea.

13. BÚSQUEDA DE PROPUESTAS

- Implementar el método de búsqueda por diferentes filtros.
- Diseñar la interfaz para la búsqueda de propuestas.
- Mostrar los resultados de la búsqueda.

14. GESTIÓN DE DATO PERSONALES

- Implementar la interfaz para la actualización de datos personales.
- Capturar y validar los datos actualizados.
- Guardar los cambios en la base de datos.