



Universidad Nacional Experimental
De los Llanos Occidentales "Ezequiel Zamora"
Unellez – Barinas
Programa Ciencias Básicas y Aplicadas
Subprograma Ingeniería Informática
Subproyecto Desarrollo de Aplicaciones II

Asociación De Karate Barinas

Docente:

España Franklin

Alumnos:

Dugarte Angely

C.I: 31.075.776

Moreno Josue

C.I:30.584.503

Pereira Wilk

C.I:30.052.584

Superlano Eduoli

C.I:30.741.644

Zambrano María

C.I: 28.421.121

Sección: T01

Barinas, 06 de mayo del 2025

Historias de Usuario

1. Como Administrador del Sistema, quiero tener un panel de control que me muestre un resumen del sistema (total de dojos, eventos, senseis, solicitudes de alumnos y usuarios registrados) para tener una visión general de la actividad de la plataforma.
2. Como Administrador del Sistema, quiero poder ver una lista de todos los dojos registrados, incluyendo los alumnos y el sensei asignado a cada uno, para poder supervisar la estructura de la asociación.
3. Como Administrador del Sistema, quiero poder crear nuevos dojos especificando su nombre, el sensei responsable y la dirección, para poder expandir la asociación y organizar nuevas ubicaciones de práctica.
4. Como Administrador del Sistema, quiero poder editar la información de un dojo existente (nombre, sensei asignado, dirección, agregar alumnos) para mantener la información actualizada.
5. Como Administrador del Sistema, quiero poder eliminar un dojo existente en caso de que sea necesario darlo de baja.
6. Como Administrador del Sistema, quiero poder ver una lista de todos los senseis registrados, incluyendo sus datos personales, para tener un directorio de los instructores de la asociación.
7. Como Administrador del Sistema, quiero poder ver los detalles de un sensei específico, incluyendo sus datos personales, el dojo al que está asignado (si aplica) y los alumnos asociados, o indicar si no tiene un dojo asignado, para tener información completa sobre cada instructor.
8. Como Administrador del Sistema, quiero poder eliminar un sensei registrado en caso de que sea necesario.
9. Como Administrador del Sistema, quiero poder registrar nuevos eventos proporcionando nombre, descripción, fecha y lugar, para poder anunciar y organizar actividades para la comunidad.
10. Como Administrador del Sistema, quiero poder ver una lista de los eventos registrados con opciones para ver los detalles, editar y eliminar, para poder gestionar el calendario de actividades de la asociación.

11. Como Administrador del Sistema, quiero poder ver los detalles de un evento específico, incluyendo la información del evento y los dojos participantes con sus alumnos y senseis, para tener una visión clara de la participación en cada actividad.
12. Como Administrador del Sistema, quiero poder editar la información de un evento existente para mantener los detalles actualizados.
13. Como Administrador del Sistema, quiero poder eliminar un evento registrado en caso de cancelación o modificación.
14. Como Administrador del Sistema, quiero poder ver una lista de las solicitudes de registro de senseis y alumnos, incluyendo el nombre del solicitante, correo electrónico, fecha de solicitud y estado, para poder gestionar el ingreso de nuevos miembros.
15. Como Administrador del Sistema, quiero poder aprobar las solicitudes de registro de senseis (basándome en la imagen del certificado) y alumnos para darles acceso a la plataforma.
16. Como Administrador del Sistema, quiero poder rechazar las solicitudes de registro de senseis y alumnos que no cumplan con los requisitos.
17. Como Sensei, quiero poder ver mis datos personales para verificar que la información en el sistema sea correcta.
18. Como Sensei, quiero poder ver el dojo al que estoy asignado para conocer mi ubicación de enseñanza.
19. Como Sensei que no está asignado a un dojo, quiero poder crear un nuevo dojo (nombre, dirección) para iniciar mi propia escuela y convertirme en el administrador del mismo.
20. Como Sensei administrador de un dojo, quiero poder agregar otros senseis a mi dojo para colaborar en la enseñanza.
21. Como Sensei administrador de un dojo, quiero poder agregar alumnos a mi dojo para gestionar a mis estudiantes.
22. Como Sensei administrador de un dojo, quiero poder crear eventos específicos para mi dojo (nombre, descripción, fecha, lugar) para organizar actividades para mis alumnos.

23. Como Sensei administrador de un dojo, quiero poder invitar a otros dojos a participar en mis eventos para fomentar la interacción entre las escuelas.
24. Como Alumno, quiero poder ver mis datos personales para verificar mi información en el sistema.
25. Como Alumno, quiero poder ver la lista de senseis asociados a mi dojo y los alumnos que también pertenecen a mi dojo para conocer a mi comunidad de práctica.
26. Como Alumno, quiero poder ver la lista de eventos en los que estoy registrado para participar para estar informado sobre las próximas actividades.
27. Como Usuario no registrado, quiero ver información general del sistema, las opciones para registrarme o iniciar sesión para poder acceder a la plataforma.
28. Como Usuario, quiero poder registrarme como alumno proporcionando mis datos para formar parte de la asociación.
29. Como Alumno registrado, quiero poder solicitar ser sensei adjuntando una imagen de mi certificado para poder optar a ser instructor.
30. Como Usuario registrado, quiero poder iniciar sesión utilizando mi correo electrónico y contraseña para acceder a mi cuenta.
31. Como Usuario registrado, quiero poder cerrar mi sesión de forma segura para proteger mi información.

Diagramas UML

Diagrama Entidad Relación

Este diagrama de Entidad-Relación modela la estructura de datos de un sistema para la "Asociación de Karate Barinas", mostrando las entidades clave (Usuario, Roles, Dojo, Evento, Solicitud, Dan, Kyu) y cómo se relacionan entre sí, incluyendo sus atributos principales y la cardinalidad de las conexiones.

Diagrama entidad relación

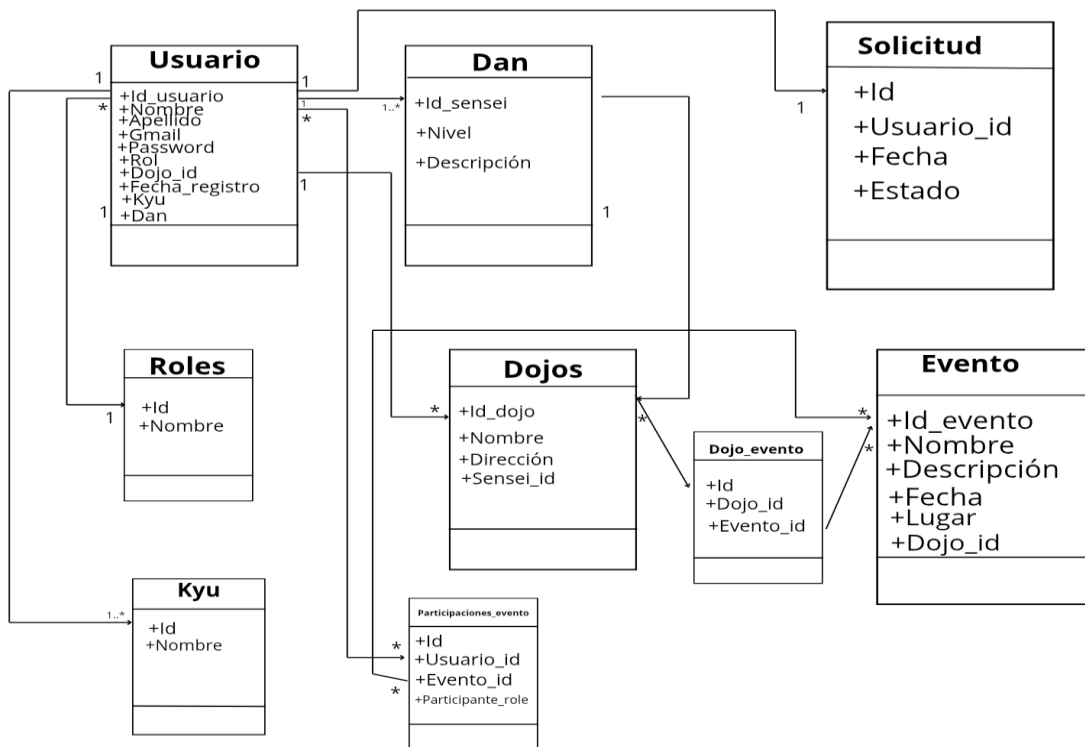


Diagrama de actividades: ilustra el flujo de interacción de los usuarios (no registrado, administrador, sensei, alumno) con el sistema de la "Asociación de Karate Barinas", detallando los procesos desde el acceso inicial y registro, pasando por la autenticación y las funcionalidades específicas disponibles para cada rol.

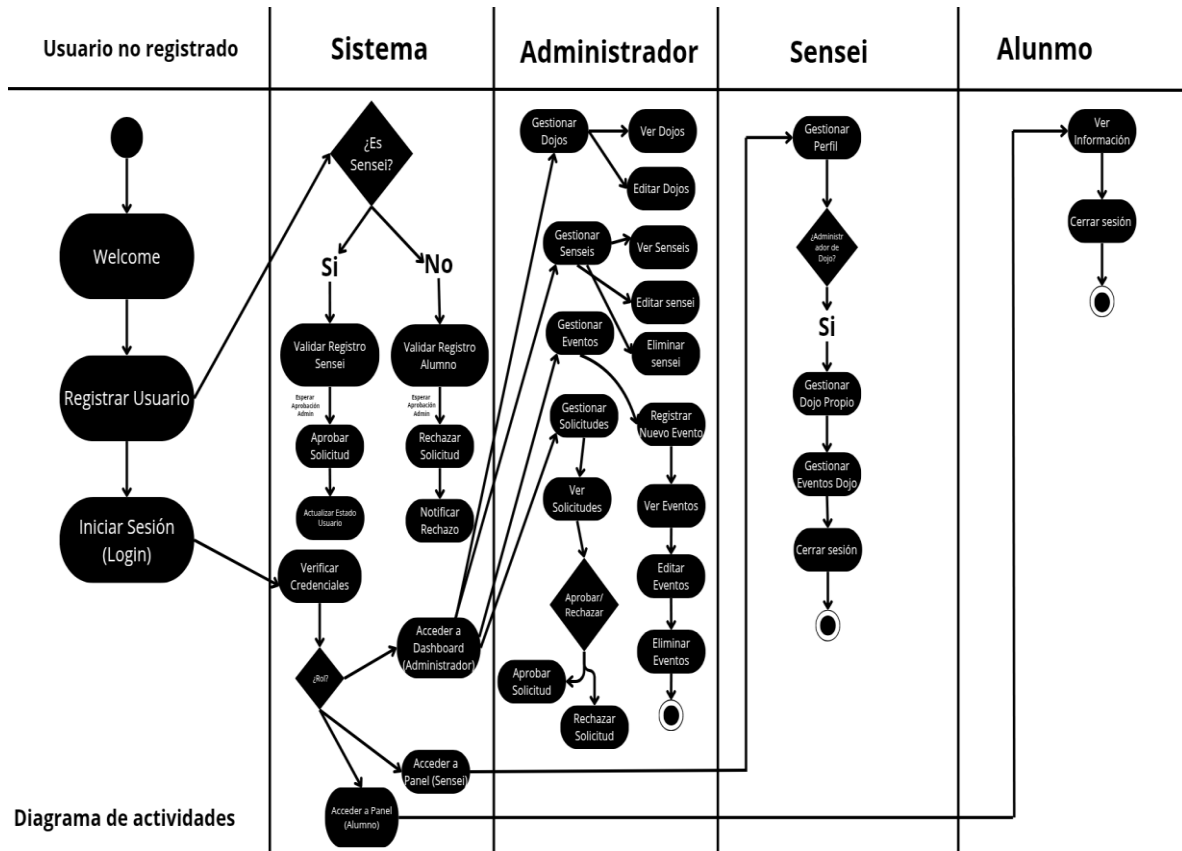


Diagrama de Caso de Uso

Diagrama de casos de uso describe las interacciones entre los diferentes actores (Usuario no registrado, Administrador, Sensei, Alumno) y el sistema "Asociación de Karate Barinas", mostrando las funcionalidades principales que el sistema ofrece para el registro, inicio de sesión y las acciones específicas disponibles para cada rol, como la gestión de dojos, senseis, eventos y la visualización de información relevante.

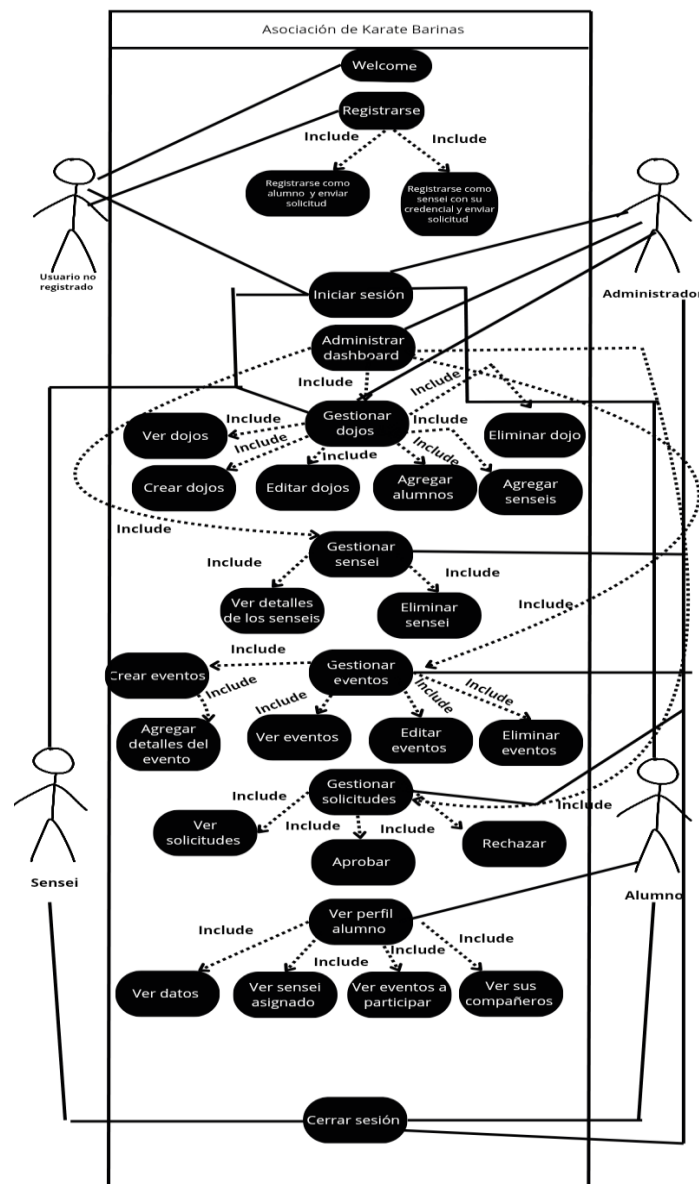


Diagrama caso de uso

Diagrama de Componentes

El Diagrama de componentes representa la estructura de alto nivel del sistema web "Asociación de Karate Barinas", mostrando las principales páginas o módulos de la interfaz de usuario y sus interconexiones, aunque carece de la representación explícita de la lógica de negocio y la base de datos.

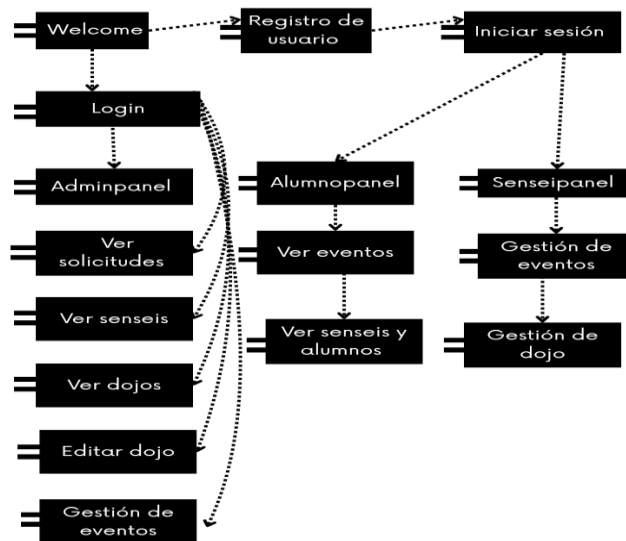


Diagrama de componentes

Diagrama de Estado

Diagrama de estado representa el comportamiento del sistema "Asociación de Karate Barinas", detallando la secuencia de acciones y decisiones desde la pantalla de inicio, pasando por la autenticación y las diferentes funcionalidades accesibles a los usuarios, mostrando el flujo de control entre las distintas partes del sistema.

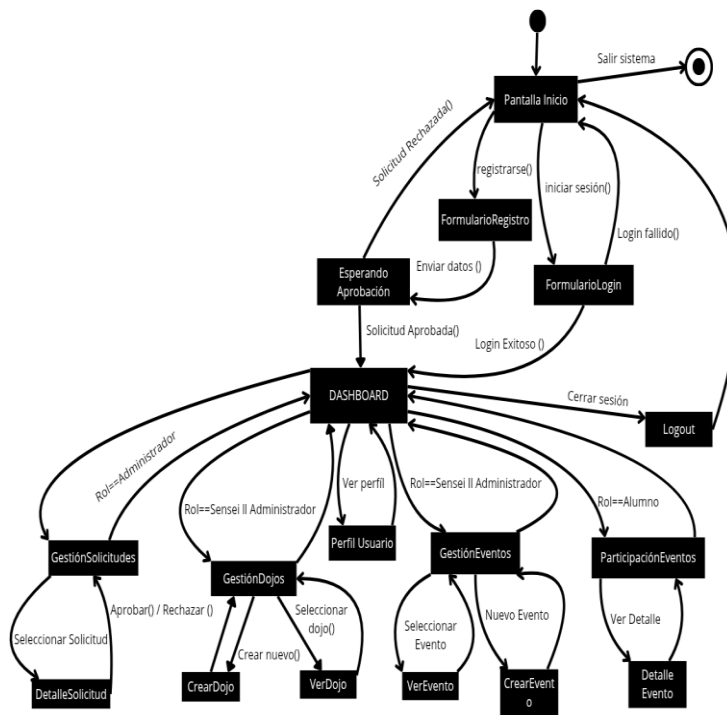
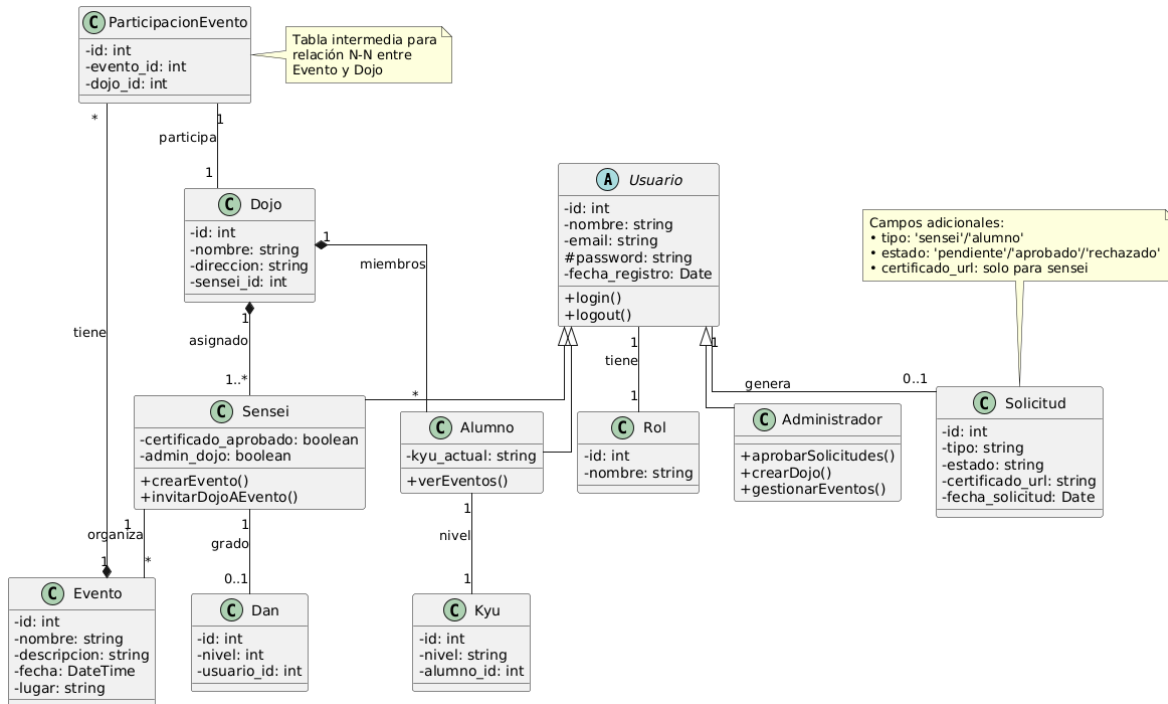


Diagrama Clase

El diagrama modela la "Asociación de Karate Barinas" con Usuarios (Administrador, Sensei, Alumno) relacionados con Dojos y Eventos. Los Administradores gestionan, los Senseis enseñan en Dojos y organizan eventos, y los Alumnos pertenecen a Dojos y asisten a eventos. También se gestionan Solicitudes de nuevos miembros.



Metodología Cascada

Fase I

Documento de Especificación de Requisitos de Software (S.R.S) del Sistema "Asociación de Karate Barinas"

Sistema: Asociación de Karate Barinas

Versión: 1.0

Fecha: 6 de mayo de 2025

Preparado por: Angely Dugarte, Eudoli Superlano, Josue Moreno, Maria Zambrano y Wilk Pereira.

1. Introducción

- **Propósito:** El sistema web "Asociación de Karate Barinas" tiene como propósito fundamental establecer una plataforma digital integral para la gestión eficiente y la conexión dinámica de la comunidad de karate en la región de Barinas. Su objetivo principal es centralizar la información crucial de dojos, senseis y alumnos, optimizar los procesos administrativos, facilitar la comunicación entre los miembros (administrador, senseis y alumnos), empoderar a los senseis en la gestión de sus dojos y eventos, mantener informados a los alumnos sobre su comunidad y eventos relevantes, y en última instancia, fortalecer la organización y promover la participación dentro de la Asociación.
- **Alcance del Producto:** El sistema web permitirá el registro de usuarios (alumnos y senseis), la gestión de dojos (creación, edición, eliminación, asignación de senseis y alumnos), la gestión de senseis (visualización, eliminación, asignación a dojos), la gestión de eventos (creación, visualización, edición, eliminación, asignación de participantes), y la gestión de solicitudes de registro. Existirán roles diferenciados para el administrador, senseis y alumnos, cada uno con funcionalidades específicas.

- Audiencia: La audiencia que va interactuar con el sistema seria el “administrador del Sistema”, persona encargadas de la gestión general de la plataforma. Su audiencia es interna y se enfoca en la supervisión, configuración y mantenimiento del sistema, así como en la aprobación de nuevos usuarios (senseis y alumnos). Los “Senseis (Maestros de Karate)”, este grupo utilizará el sistema para gestionar la información relacionada con sus dojos y alumnos (si son administradores de su dojo), así como para informarse y participar en eventos. Su audiencia incluye a otros senseis, alumnos y el administrador. Y por último los “alumnos de karate”, serán los usuarios finales que accederán al sistema para ver su información personal, conocer a otros miembros de su dojo y mantenerse informados sobre los eventos en los que pueden participar.
- Características Generales:
 - Registro y gestión de perfiles para administradores, senseis y alumnos.
 - Administración integral de dojos, incluyendo asignación de miembros.
 - Gestión de senseis y alumnos con roles y permisos definidos.
 - Creación y gestión de eventos con participación de dojos y alumnos.
 - Proceso de aprobación para nuevos senseis y alumnos.

2. Descripción General

- Perspectiva del Producto: El sistema web "Asociación de Karate Barinas" será una plataforma centralizada accesible a través de navegadores web. Facilitará la organización y gestión de la comunidad de karate en Barinas, permitiendo a los usuarios acceder a información relevante y realizar tareas específicas según su rol.
- Funciones del Producto: Las funciones principales del sistema incluyen:
 - Registro e inicio de sesión de usuarios con diferentes roles (administrador, sensei, alumno).
 - Gestión de perfiles de usuario.
 - Gestión de dojos (alta, baja, modificación, visualización de alumnos y senseis).
 - Gestión de senseis (alta a través de aprobación de certificado, baja, visualización, asignación a dojos).

- Gestión de alumnos (alta a través de aprobación, visualización, asignación a dojos).
- Gestión de eventos (creación, visualización, modificación, eliminación, asignación de participantes).
- Gestión de solicitudes de registro de usuarios.
- Tablero de control para el administrador con información general.
- Características de los Usuarios:
 - Administrador: Persona encargada de la gestión general del sistema, incluyendo la aprobación de registros de senseis y alumnos, la gestión de dojos, senseis, eventos y solicitudes.
 - Sensei: Usuario registrado que puede ser asignado a un dojo. Si no está asignado, puede crear un nuevo dojo y gestionarlo (agregar otros senseis y alumnos, crear eventos para su dojo e invitar a otros).
 - Alumno: Usuario registrado que pertenece a un dojo y puede ver su información personal, los senseis y alumnos de su dojo, y los eventos en los que participa.
- Restricciones:
 - El sistema debe ser accesible a través de navegadores web modernos.
 - Los usuarios deben tener acceso a internet y un navegador web compatible.

3. Requisitos Específicos

- Requisitos Funcionales:
 - Autenticación y Autorización:
 - Descripción: El sistema debe permitir a los usuarios registrarse e iniciar sesión de forma segura, y debe controlar el acceso a las funcionalidades según el rol del usuario.
 - Welcome: Mostrar información del sistema, opciones de inicio de sesión, registro.
 - Registro de Usuario: Permitir el registro de nuevos usuarios como "alumno" por defecto.

- Registro de Sensei: Permitir a los usuarios registrados como alumnos solicitar ser sensei adjuntando una imagen de su certificado.
- Aprobación de Sensei/Alumno: El administrador debe poder aprobar o rechazar las solicitudes de registro de senseis (basado en la imagen del certificado) y alumnos.
- Inicio de Sesión (Login): Permitir a los usuarios autenticarse utilizando su correo electrónico y contraseña.
- Cierre de Sesión: Permitir a los usuarios cerrar su sesión de forma segura.
- Panel de Administración (Dashboard):
 - Descripción: Al iniciar sesión como administrador, el sistema debe mostrar un panel con un resumen de la información del sistema.
 - Mostrar el número total de dojos, eventos, senseis, solicitudes de alumnos y usuarios registrados.
- Gestión de Dojos (Admin Panel):
 - Ver Dojos: Mostrar una lista de todos los dojos registrados, incluyendo los alumnos y el sensei asignado a cada uno.
 - Editar Dojos: Permitir al administrador:
 - Ver un formulario para crear un nuevo dojo (nombre, sensei asignado, dirección).
 - Guardar la información del nuevo dojo.
 - Ver una lista de los dojos existentes con opciones para editar y eliminar cada dojo.
 - Editar la información de un dojo existente (nombre, sensei asignado, dirección).
 - Eliminar un dojo existente.
 - Agregar alumnos al dojo.
- Gestión de Senseis (Admin Panel):
 - Ver Senseis: Mostrar una lista de todos los senseis registrados, incluyendo sus datos personales.

- Ver Detalles de Sensei: Permitir al administrador ver los detalles de un sensei específico (datos personales, dojo asignado y alumnos asociados, o indicar si no tiene dojo asignado).
- Eliminar Sensei: Permitir al administrador eliminar un sensei registrado.
- Gestión de Eventos (Admin Panel):
 - Registrar Nuevo Evento: Mostrar un formulario para registrar un nuevo evento (nombre, descripción, fecha, lugar).
 - Agregar Evento: Guardar la información del nuevo evento.
 - Ver Eventos: Mostrar una lista de los eventos registrados con opciones para ver detalles, editar y eliminar.
 - Ver Detalles de Evento: Mostrar los detalles de un evento específico (datos del evento, dojos participantes con sus alumnos y senseis).
 - Editar Evento: Permitir al administrador modificar la información de un evento existente.
 - Eliminar Evento: Permitir al administrador eliminar un evento registrado.
- Gestión de Solicitudes (Admin Panel):
 - Ver Solicitudes: Mostrar una lista de las solicitudes de registro de senseis y alumnos, incluyendo el nombre del solicitante, correo electrónico, fecha de solicitud, estado (pendiente, aprobada, rechazada) y acciones.
 - Aprobar Solicitud: Permitir al administrador aprobar una solicitud.
 - Rechazar Solicitud: Permitir al administrador rechazar una solicitud.
- Panel de Sensei:
 - Ver Datos Personales: Mostrar la información personal del sensei.
 - Ver Dojo Asignado: Mostrar el dojo al que está asignado el sensei (si lo está).
 - Crear Dojo (Si no asignado): Permitir al sensei crear un nuevo dojo (nombre, dirección) si no está asignado a uno. Al crear el dojo, el sensei se convierte en administrador de ese dojo.

- Gestionar Senseis del Dojo (Administrador de Dojo): Permitir al sensei administrador de un dojo agregar otros senseis al dojo.
- Gestionar Alumnos del Dojo (Administrador de Dojo): Permitir al sensei administrador de un dojo agregar alumnos al dojo.
- Crear Eventos del Dojo (Administrador de Dojo): Permitir al sensei administrador de un dojo crear eventos específicos para su dojo.
- Invitar a Otros Dojos (Administrador de Dojo): Permitir al sensei administrador de un dojo invitar a otros dojos a participar en sus eventos.
- Panel de Alumno:
 - Ver Datos Personales: Mostrar la información personal del alumno.
 - Ver Senseis del Dojo: Mostrar la lista de senseis asociados a su dojo y los alumnos del mismo dojo.
 - Ver Eventos Participantes: Mostrar la lista de eventos en los que el alumno está registrado para participar.
- Requisitos No Funcionales:
 - Requisitos de Rendimiento:
 - Tiempo de Respuesta: El tiempo de respuesta promedio del sistema para las operaciones más comunes (por ejemplo, inicio de sesión, carga de páginas) no debe exceder los 4 segundos.
 - Concurrencia: El sistema debe soportar muchos usuarios concurrentes sin degradación significativa del rendimiento.
 - Requisitos de Seguridad:
 - Autenticación Segura: El sistema debe utilizar métodos seguros para la autenticación de usuarios, protegiendo las contraseñas mediante hashing.
 - Autorización Basada en Roles: El acceso a las funcionalidades del sistema debe estar restringido según el rol del usuario.
 - Protección de Datos: Los datos sensibles de los usuarios deben almacenarse de forma segura.
 - Requisitos de Usabilidad:

- Interfaz Intuitiva: La interfaz de usuario debe ser clara, intuitiva y fácil de navegar para usuarios con diferentes niveles de experiencia tecnológica.
- Requisitos de Mantenibilidad:
 - Código Bien Estructurado: El código del sistema debe estar bien estructurado, documentado y seguir buenas prácticas de programación para facilitar su mantenimiento y futuras modificaciones.
 - Modularidad: El sistema debe diseñarse en módulos independientes para facilitar la actualización y el mantenimiento de componentes específicos.
 - Documentación Interna: El código debe estar comentado de forma clara y concisa.
- Requisitos de Interfaz Externa:
 - Interfaces de Usuario: El sistema será accesible a través de una interfaz gráfica de usuario basada en web, compatible con navegadores modernos.
 - Interfaces de Comunicación: El sistema utilizará el protocolo HTTP/HTTPS para la comunicación entre el cliente (navegador web) y el servidor.

4. Dependencias y Asunciones

- Dependencias:
 - Software: Navegador web moderno
 - Conectividad: Acceso a internet.
- Asunciones:
 - Plataforma de Desarrollo:(Backend) Php, (Fronted) html, css, bootstrap y JavaScript.
 - Sistema de Gestión de Bases de Datos: MySQL.
 - Entorno de Desarrollo: XAMPP para el desarrollo local.

Fase II

Diseño del Sistema "Asociación de Karate Barinas"

Diseño de Alto Nivel (Diseño Arquitectónico)

1. Arquitectura de Tres Capas:

○ Capa de Presentación (Frontend):

- Es la interfaz con la que interactúan los usuarios.
- Estará construida con HTML, CSS (Bootstrap para el diseño responsivo) y JavaScript para la interactividad.
- Se encargará de mostrar la información al usuario y de enviar las peticiones al backend.
- Tendrá diferentes interfaces para cada rol: Administrador, Sensei y Alumno.

○ Capa de Lógica de Negocio (Backend o Servidor):

- Contendrá la lógica principal de la aplicación.
- Estará desarrollada con PHP.
- Recibirá las peticiones del frontend, realizará las operaciones necesarias (validaciones, procesamiento de datos, interacción con la base de datos) y enviará las respuestas al frontend.
- Implementará las funcionalidades definidas en los requisitos funcionales (gestión de usuarios, dojos, eventos, etc.).

○ Capa de Acceso a Datos (Base de Datos):

- Se encargará de almacenar y gestionar la información del sistema.
- Utilizará MySQL como sistema de gestión de bases de datos.
- El backend (PHP) interactuará con esta capa para leer y escribir datos.

2. Diseño de Bajo Nivel (Diseño Inicial): Vamos a empezar a detallar algunos aspectos clave del diseño de bajo nivel: las interfaces de usuario iniciales y la estructura de la base de datos.
 - Diseño de la Interfaz de Usuario (UI) Inicial - Wireframes Mentales: En esta etapa, no vamos a crear diseños visuales detallados, sino a esbozar mentalmente la estructura básica de las pantallas principales para cada rol. Esto nos ayudará a entender cómo los usuarios interactuarán con el sistema.
 - Página de Bienvenida (Welcome):
 - Información general del sistema.
 - Botones para "Iniciar Sesión" y "Registrarse".
 - Formulario de Registro (Alumno):
 - Campos para nombre, apellido, correo electrónico, contraseña.
 - Formulario de Solicitud de Registro (Sensei - después de registrarse como alumno):
 - Instrucciones.
 - Campo para adjuntar la imagen del certificado.
 - Botón para "Enviar Solicitud".
 - Formulario de Inicio de Sesión (Login):
 - Campos para correo electrónico y contraseña.
 - Botón para "Iniciar Sesión".
 - Panel de Administración (Dashboard):
 - Visualización de los totales (dojos, eventos, senseis, solicitudes, usuarios).
 - Menú de navegación a la izquierda para:
 - Gestión de Dojos (Ver, Crear, Editar, Eliminar, Agregar Alumnos)
 - Gestión de Senseis (Ver, Detalles, Eliminar)

- Gestión de Eventos (Registrar Nuevo, Ver, Detalles, Editar, Eliminar)
 - Gestión de Solicitudes (Ver, Aprobar, Rechazar)
- Panel del Sensei:
 - Visualización de sus datos personales y dojo asignado (si aplica).
 - Opciones de menú (dependiendo si es administrador de un dojo o solo un sensei asignado):
 - Ver Datos Personales
 - Ver Dojo Asignado
 - Crear Dojo (si no está asignado)
 - Gestionar Senseis del Dojo (si es administrador)
 - Gestionar Alumnos del Dojo (si es administrador)
 - Crear Eventos del Dojo (si es administrador)
 - Invitar a Otros Dojos (si es administrador)
- Panel del Alumno:
 - Visualización de sus datos personales.
 - Información del dojo al que pertenece.
 - Lista de senseis y alumnos de su dojo.
 - Lista de eventos en los que participa.
- Formularios de Gestión (Ejemplos):
 - Crear/Editar Dojo: Campos para nombre, sensei asignado (lista desplegable de senseis), dirección.
 - Registrar Nuevo Evento: Campos para nombre, descripción, fecha, lugar.

3. Diseño de la Lógica del Backend (Alto Nivel):

- Peticiones del Frontend: El frontend (HTML, CSS, JavaScript) enviará peticiones HTTP (GET, POST, PUT, DELETE) a URLs específicas (endpoints) del backend (PHP) para interactuar con el sistema.
- Procesamiento en el Backend (PHP):
 - El backend recibirá las peticiones, validará los datos enviados por el frontend.

- Ejecutará la lógica de negocio necesaria para cada funcionalidad (ej: registrar un usuario, crear un dojo, etc.).
- Interactuará con la base de datos (MySQL) para leer, escribir, actualizar o eliminar información según la petición.
- Generará una respuesta (generalmente en formato JSON) y la enviará de vuelta al frontend para actualizar la interfaz.

2. Ejemplos de Interacción Backend-Base de Datos:

- Registrar Usuario: El backend recibirá los datos del formulario, los validará y los insertará en la tabla de usuarios de MySQL.
- Ver Dojos: El backend consultará la tabla dojos de MySQL y posiblemente realizará consultas relacionadas con las tablas de usuarios (para obtener el sensei asignado) y una tabla de relación dojo_alumnos para obtener los alumnos.
- Crear Evento: El backend recibirá los datos del formulario y los insertará en la tabla eventos de MySQL.
- Aprobar Solicitud: El backend actualizará el estado de la solicitud en la tabla solicitudes de MySQL y, para los senseis aprobados, podría actualizar el rol del usuario en la tabla usuarios.

4. Consideraciones de Seguridad:

- Autenticación y Autorización: Se implementará un sistema de inicio de sesión seguro basado en correo electrónico y contraseña. Los roles de usuario (Administrador, Sensei, Alumno) se verificarán en el backend para controlar el acceso a las diferentes funcionalidades y datos del sistema. Se utilizarán sesiones seguras para mantener el estado del usuario logueado.
- Validación de Datos: Todas las entradas del usuario enviadas desde el frontend serán estrictamente validadas en el backend para prevenir inyecciones SQL y otros ataques maliciosos.

- **Protección de Contraseñas:** Las contraseñas de los usuarios se almacenarán de forma segura en la base de datos utilizando funciones de hashing robustas (ej: `password_hash` en PHP).
- **Comunicación Segura:** Se recomienda el uso de HTTPS para cifrar la comunicación entre el navegador del usuario (frontend) y el servidor (backend), protegiendo la información sensible durante la transmisión.
- **Prevención de Acceso No Autorizado:** El backend implementará mecanismos para asegurar que solo los usuarios autenticados y autorizados puedan acceder a las funcionalidades y datos correspondientes a su rol.

Fase III

Implementación (Construcción) del Sistema "Asociación de Karate Barinas"

En esta fase, el equipo de desarrollo procederá a construir el sistema web "Asociación de Karate Barinas" basándose en los requisitos detallados en el Documento de Especificación de Requisitos de Software (SRS) y el diseño arquitectónico y de bajo nivel definido en la Fase II. La implementación se realizará siguiendo las tecnologías seleccionadas: PHP para el backend, HTML, CSS (con Bootstrap) y JavaScript para el frontend, y MySQL para la gestión de la base de datos.

1. Desarrollo del Frontend:

- **Implementación de la Interfaz de Usuario:** Los desarrolladores frontend codificarán las interfaces de usuario para los diferentes roles (Administrador, Sensei, Alumno) y las páginas públicas (Welcome, Registro, Login) utilizando HTML para la estructura, CSS (con Bootstrap) para el estilo y la responsividad, y JavaScript para la interactividad. Se implementarán las estructuras básicas de las pantallas esbozadas en los wireframes mentales de la Fase II.
- **Lógica del Cliente:** Se desarrollará el código JavaScript necesario para:
 - Manejar las interacciones del usuario en las diferentes páginas y formularios.

- Realizar validaciones básicas de los datos ingresados por el usuario antes de enviarlos al backend.
 - Enviar peticiones asíncronas (utilizando la Fetch API o AJAX) a las APIs del backend para realizar acciones como registrarse, iniciar sesión, crear/editar elementos, etc.
 - Recibir las respuestas del backend (probablemente en formato JSON) y actualizar dinámicamente la interfaz de usuario para mostrar la información o los resultados de las acciones.
- Integración con las APIs del Backend: Se implementarán las llamadas a los endpoints del backend (definidos en la Fase II) para cada una de las funcionalidades requeridas por las historias de usuario. Por ejemplo:
- Llamadas POST a `/api/auth/register/alumno` y `/api/auth/register/sensei` para el registro de usuarios.
 - Llamada POST a `/api/auth/login` para el inicio de sesión.
 - Llamadas GET, POST, PUT, DELETE a los endpoints de `/api/admin/dojos`, `/api/admin/senseis`, `/api/admin/eventos`, `/api/admin/solicitudes` para la gestión por parte del administrador (historias de usuario 1-16).
 - Llamadas GET a `/api/sensei/profile`, `/api/sensei/dojo` y llamadas POST a `/api/sensei/dojos` (si no está asignado), `/api/sensei/dojo/senseis`, `/api/sensei/dojo/alumnos`, `/api/sensei/dojo/eventos`, `/api/sensei/dojo/invitaciones` para las funcionalidades del sensei (historias de usuario 17-23).
 - Llamadas GET a `/api/alumno/profile`, `/api/alumno/dojo`, `/api/alumno/eventos` para las funcionalidades del alumno (historias de usuario 24-26).

2. Desarrollo del Backend:

- Implementación de la Lógica de Negocio (PHP): Se codificará la lógica principal de la aplicación en PHP para cada una de las funcionalidades descritas en el SRS y las historias de usuario. Esto incluirá:

- Gestión de Usuarios: Implementación de la lógica para el registro, inicio de sesión (verificando correo y contraseña), cierre de sesión y la gestión de los roles (administrador, sensei, alumno) y sus respectivos permisos.
 - Gestión de Dojos: Implementación de la lógica para crear, leer, actualizar y eliminar dojos, así como para asignar senseis y alumnos a los dojos.
 - Gestión de Senseis: Implementación de la lógica para visualizar la lista de senseis, ver sus detalles y eliminarlos. También se implementará la lógica para la aprobación de solicitudes de sensei por parte del administrador.
 - Gestión de Eventos: Implementación de la lógica para crear, leer, actualizar y eliminar eventos, así como para gestionar la participación de dojos y alumnos.
 - Gestión de Solicitudes: Implementación de la lógica para visualizar, aprobar y rechazar las solicitudes de registro de senseis y alumnos.
 - Paneles de Control: Implementación de la lógica para generar la información mostrada en el panel de administración, el panel del sensei y el panel del alumno, basándose en los datos almacenados en la base de datos.
- Desarrollo de las APIs (Endpoints en PHP): Se crearán los scripts PHP que responderán a las peticiones HTTP del frontend en las URLs (endpoints) diseñadas en la Fase II. Estos scripts recibirán los datos del frontend, ejecutarán la lógica de negocio necesaria y devolverán las respuestas (generalmente en formato JSON).
 - Integración con la Base de Datos (MySQL): Se utilizarán las extensiones de PHP (como MySQLi o PDO) para interactuar con la base de datos MySQL. Se escribirán las consultas SQL necesarias para:
 - Crear las tablas definidas en el esquema de la base de datos (aún por detallar completamente).
 - Implementar las operaciones CRUD (Crear, Leer, Actualizar, Eliminar) para los datos de las tablas (usuarios, dojos, eventos, solicitudes, etc.) según las necesidades de la lógica de negocio.
 - Implementación de la Seguridad (PHP): Se implementarán las medidas de seguridad en el backend:

- Verificación de las credenciales de inicio de sesión y gestión de sesiones seguras.
 - Validación exhaustiva de todos los datos recibidos del frontend para prevenir ataques.
 - Utilización de funciones de hashing seguras (como password_hash) para almacenar las contraseñas de los usuarios en la base de datos.
 - Implementación de la autorización basada en roles para asegurar que solo los usuarios autenticados y con los permisos adecuados puedan acceder a ciertas funcionalidades y datos.
 - Implementación de medidas para prevenir ataques comunes como la inyección SQL.
- Manejo de Errores y Excepciones (PHP): Se implementará un sistema robusto para el manejo de errores y excepciones en el backend. Esto incluirá el registro de errores para el seguimiento y la generación de respuestas informativas (en formato JSON) al frontend cuando ocurran problemas.

3. Pruebas Unitarias: Los desarrolladores backend escribirán pruebas unitarias utilizando frameworks de pruebas de PHP (como PHPUnit) para verificar la correcta funcionalidad de las funciones, clases y módulos individuales del código PHP. Esto asegurará que la lógica de negocio se implemente correctamente y ayudará a detectar errores tempranamente.

Fase IV: Pruebas (Testing) del Sistema "Asociación de Karate Barinas"

En esta fase, se llevará a cabo una serie de pruebas exhaustivas para asegurar la calidad, funcionalidad y rendimiento del sistema web "Asociación de Karate Barinas". El objetivo principal es identificar y corregir cualquier defecto o error que pueda existir antes de que el sistema sea entregado o puesto en producción. Se realizarán diferentes tipos de pruebas, involucrando tanto al equipo de desarrollo como a posibles usuarios finales.

1. Pruebas Unitarias (Continuación y Verificación):

- Aunque iniciadas en la Fase III, se continuará y se verificará la cobertura y los resultados de las pruebas unitarias realizadas por los desarrolladores backend. Se asegurará que cada función, clase y módulo del código PHP funcione correctamente de forma aislada, cumpliendo con las especificaciones de diseño.

2. Pruebas de Integración:

- Se realizarán pruebas para verificar la correcta interacción y comunicación entre los diferentes módulos y componentes del sistema. Esto incluye probar la integración entre el frontend (JavaScript) y el backend (APIs PHP), así como la interacción del backend con la base de datos MySQL. Se asegurará que los datos fluyan correctamente entre las capas y que las diferentes partes del sistema trabajen juntas sin problemas.
 - Verificar que el registro de un nuevo alumno a través del frontend se guarda correctamente en la base de datos a través de la API del backend.
 - Probar que la creación de un nuevo dojo por parte del administrador a través del frontend se refleja correctamente en la lista de dojos mostrada en el panel de administración.
 - Asegurar que la asignación de un sensei a un dojo en el panel de administración se actualiza correctamente en la base de datos y se visualiza en los paneles correspondientes.

3. Pruebas del Sistema (o Pruebas Funcionales):

- Se llevarán a cabo pruebas funcionales de extremo a extremo para verificar que el sistema cumpla con todos los requisitos funcionales especificados en el SRS y las

historias de usuario. Estas pruebas se realizarán desde la perspectiva del usuario final, simulando sus interacciones con la interfaz de usuario.

- Como Administrador: Probar la funcionalidad del panel de control, la visualización, creación, edición y eliminación de dojos, senseis y eventos, así como la gestión de las solicitudes de registro (aprobar y rechazar).
- Como Sensei: Probar la visualización de sus datos y dojo asignado, la creación de un nuevo dojo (si aplica), la gestión de senseis y alumnos de su dojo (si es administrador), la creación de eventos de dojo y la invitación a otros dojos.
- Como Alumno: Probar la visualización de sus datos, la información de su dojo, la lista de senseis y alumnos de su dojo, y la lista de eventos en los que está registrado.
- Como Usuario no registrado/Usuario: Probar la visualización de la información general, el registro como alumno, la solicitud para ser sensei, el inicio y cierre de sesión.

4. Pruebas de Usabilidad:

- Se realizarán pruebas con usuarios representativos (si es posible) para evaluar la facilidad de uso, la intuitividad de la interfaz y la satisfacción general con el sistema. Se identificarán posibles problemas de navegación, diseño o comprensión de las funcionalidades.
 - Pedir a un usuario simular el proceso de registro como alumno y observar si encuentra alguna dificultad.
 - Solicitar a un sensei simular la creación de un evento para su dojo y evaluar la claridad del formulario y el flujo de trabajo.
 - Pedir a un administrador simular la aprobación de una solicitud de registro de sensei y evaluar la eficiencia del proceso.

5. Pruebas de Rendimiento:

- Se realizarán pruebas para evaluar la capacidad de respuesta del sistema bajo diferentes cargas de usuarios y condiciones de uso. Se verificarán los tiempos de respuesta de las operaciones más comunes (como se especificó en los requisitos no funcionales), la concurrencia y la estabilidad del sistema bajo estrés.
 - Simular varios usuarios iniciando sesión simultáneamente para verificar si el sistema se mantiene estable y los tiempos de respuesta son aceptables.
 - Probar la carga de la lista de dojos con una gran cantidad de registros para asegurar que la página se carga dentro del tiempo de respuesta especificado.

6. Pruebas de Seguridad:

- Se realizarán pruebas específicas para identificar posibles vulnerabilidades de seguridad en el sistema, como:
 - Pruebas de inyección SQL.
 - Pruebas de scripting entre sitios (XSS).
 - Pruebas de autenticación y autorización para asegurar que los roles de usuario se apliquen correctamente y que no se pueda acceder a funcionalidades no autorizadas.
 - Pruebas de protección de datos sensibles.

7. Pruebas de Aceptación del Usuario (UAT - User Acceptance Testing): En esta etapa, se involucrará a los usuarios finales o a los stakeholders clave para que prueben el sistema en un entorno que simule el entorno de producción. El objetivo es obtener la confirmación de que el sistema cumple con sus necesidades y expectativas antes de su implementación final. Los usuarios realizarán pruebas basadas en escenarios de uso reales.

Documentación de la Fase de Pruebas:

Es crucial documentar detalladamente el proceso de pruebas, incluyendo:

- Plan de Pruebas: Define los objetivos, el alcance, los tipos de pruebas a realizar, los recursos necesarios, los criterios de éxito y los roles y responsabilidades.

- **Casos de Prueba:** Documentos que describen los pasos específicos, los datos de entrada, las condiciones previas y los resultados esperados para cada prueba funcional.
- **Registro de Defectos:** Un sistema para registrar y rastrear todos los errores o defectos encontrados durante las pruebas, incluyendo su descripción, severidad, prioridad y estado de corrección.
- **Informe de Pruebas:** Un resumen de los resultados de las pruebas, incluyendo las métricas de calidad, los defectos encontrados y su estado, y la evaluación general de la preparación del sistema para su implementación.

Fase V: Implementación (Deployment) y Despliegue del Sistema "Asociación de Karate Barinas"

En esta fase, el sistema web "Asociación de Karate Barinas", que ha sido probado exhaustivamente en la Fase IV, se implementará en el entorno de producción. Esto implica la configuración del servidor, la instalación de la aplicación y la puesta en marcha para que esté accesible a los administradores, senseis y alumnos de la asociación.

1. Preparación del Entorno de Producción:

- **Selección y Configuración del Servidor:** Se elegirá un proveedor de hosting adecuado para las necesidades del sistema en producción. Se configurará el servidor (ya sea Apache o Nginx, según la decisión tomada en la Fase II) con los requisitos necesarios para ejecutar la aplicación PHP y el servidor de base de datos MySQL. Esto incluye la instalación de los paquetes de software necesarios, la configuración de los ajustes del servidor y la optimización para el rendimiento y la seguridad en un entorno real.
- **Configuración de la Base de Datos:** Se configurará el servidor de base de datos MySQL en el entorno de producción. Se crearán las bases de datos y las tablas necesarias según el esquema diseñado en la Fase II. Se asegurará la correcta configuración de los permisos de acceso para la aplicación.
- **Configuración de Seguridad del Servidor:** Se implementarán medidas de seguridad adicionales en el servidor de producción, como la configuración de firewalls, la instalación de certificados SSL para habilitar HTTPS (como se recomendó en la Fase

II), la configuración de copias de seguridad automáticas y la implementación de otras prácticas de seguridad recomendadas para servidores web.

2. Despliegue de la Aplicación:

- **Transferencia de Archivos:** Los archivos del frontend (HTML, CSS, JavaScript, Bootstrap) y del backend (scripts PHP) se transferirán desde el entorno de desarrollo o el repositorio de control de versiones al servidor de producción. Esto se puede realizar mediante herramientas como FTP, SFTP o mediante sistemas de despliegue automatizado.
- **Configuración de la Aplicación:** Se realizarán las configuraciones necesarias en la aplicación PHP para que funcione correctamente en el entorno de producción. Esto puede incluir la configuración de la conexión a la base de datos con las credenciales del servidor de producción, la configuración de rutas de archivos y directorios, y cualquier otra variable de entorno específica del entorno de producción.
- **Pruebas de Despliegue:** Después de desplegar la aplicación, se realizarán pruebas rápidas en el entorno de producción para asegurar que la aplicación se haya instalado correctamente, que se pueda acceder a las páginas principales y que la conexión a la base de datos funcione.

3. Puesta en Marcha y Verificación:

- **Activación del Sistema:** Una vez que se hayan completado los pasos de despliegue y las pruebas iniciales sean exitosas, se activará el sistema para que esté accesible a los usuarios finales a través de la URL designada.
- **Verificación en el Entorno de Producción:** Se realizará una verificación más exhaustiva del sistema en el entorno de producción para asegurar que todas las funcionalidades sigan operando correctamente en el entorno real. Esto puede incluir la ejecución de algunos de los casos de prueba críticos de la Fase IV.

4. Comunicación y Capacitación (si es necesario):

- Comunicación a los Usuarios: Se comunicará a los administradores, senseis y alumnos de la "Asociación de Karate Barinas" la disponibilidad del nuevo sistema y se les proporcionarán las instrucciones necesarias para acceder y utilizar la plataforma.
- Capacitación (si es necesaria): Si el sistema introduce cambios significativos en los procesos o si se requiere un conocimiento específico para su uso, se proporcionará capacitación a los diferentes roles de usuario.

5. Documentación de la Implementación:

Se documentarán los pasos realizados durante el proceso de implementación, incluyendo:

- Configuración del Entorno de Producción: Detalles de la configuración del servidor, la base de datos y las medidas de seguridad implementadas.
- Proceso de Despliegue: Los pasos seguidos para transferir y configurar la aplicación en el servidor de producción.
- Guía de Despliegue (para futuras implementaciones o actualizaciones).
- Información de Acceso al Sistema para los diferentes roles de usuario.

Una vez completada esta fase, el sistema "Asociación de Karate Barinas" estará en funcionamiento y disponible para su uso por la comunidad de karate.

Fase VI: Mantenimiento del Sistema "Asociación de Karate Barinas"

En esta fase, una vez que el sistema ha sido desplegado y está en uso, se llevarán a cabo las actividades necesarias para garantizar su funcionamiento continuo, corregir errores, realizar mejoras y adaptarlo a las posibles evoluciones de las necesidades de la "Asociación de Karate Barinas". El mantenimiento es un proceso continuo que se extiende durante la vida útil del sistema.

1. Monitoreo del Sistema:

- Supervisión del Servidor y la Aplicación: Se realizará un monitoreo constante del servidor de producción para asegurar su correcto funcionamiento, verificar la disponibilidad de los recursos (CPU, memoria, disco) y detectar posibles problemas de rendimiento o caídas del sistema. También se supervisará la aplicación para identificar errores en tiempo real a través de logs y herramientas de monitoreo de aplicaciones.
- Monitoreo de la Base de Datos: Se supervisará el rendimiento de la base de datos MySQL, incluyendo la utilización de recursos, la ejecución de consultas y la detección de posibles cuellos de botella o errores.

2. Gestión de Errores y Correcciones (Mantenimiento Correctivo):

- Identificación y Reporte de Errores: Los usuarios (administradores, senseis, alumnos) podrán reportar errores o problemas que encuentren al utilizar el sistema. También se identificarán errores a través del monitoreo del sistema y los logs de la aplicación.
- Análisis y Diagnóstico: El equipo de mantenimiento analizará los errores reportados para diagnosticar la causa raíz del problema.
- Corrección de Errores: Una vez identificada la causa, se desarrollarán y probarán las correcciones necesarias en el código del frontend o del backend.
- Despliegue de Correcciones: Las correcciones probadas se desplegarán al entorno de producción siguiendo un proceso controlado para minimizar la interrupción del servicio.
- Verificación de la Corrección: Después del despliegue, se verificará que el error reportado se haya corregido correctamente y que no se hayan introducido nuevos problemas.
- Adaptación a Cambios y Nuevos Requisitos (Mantenimiento Adaptativo y Evolutivo):
- Identificación de Nuevas Necesidades: La "Asociación de Karate Barinas" podría identificar nuevas funcionalidades o la necesidad de adaptar las existentes debido a cambios en sus procesos, normativas o las necesidades de sus miembros.

- **Análisis de Requisitos:** Se analizarán estos nuevos requerimientos para comprender su alcance e impacto en el sistema existente.
- **Diseño e Implementación de Cambios:** Se diseñarán y desarrollarán las modificaciones o nuevas funcionalidades necesarias, siguiendo las fases de diseño e implementación (aunque de forma más ágil y focalizada).
- **Pruebas de los Cambios:** Se realizarán pruebas exhaustivas (unitarias, de integración y funcionales) para asegurar que los cambios implementados funcionen correctamente y no afecten negativamente la funcionalidad existente.
- **Despliegue de los Cambios:** Los cambios probados se desplegarán al entorno de producción.

4. Optimización y Mejoras (Mantenimiento Perfectivo):

- **Análisis del Rendimiento:** Se analizará el rendimiento del sistema para identificar áreas donde se puedan realizar optimizaciones para mejorar la velocidad de respuesta, la eficiencia y la escalabilidad.
- **Implementación de Optimizaciones:** Se realizarán los cambios necesarios en el código, la base de datos o la configuración del servidor para mejorar el rendimiento.
- **Mejoras de Usabilidad:** Basándose en la retroalimentación de los usuarios o en análisis de usabilidad, se podrán realizar mejoras en la interfaz de usuario para hacerla más intuitiva y fácil de usar.
- **Actualizaciones Tecnológicas:** Se realizarán las actualizaciones necesarias de las tecnologías subyacentes (PHP, MySQL, Bootstrap, JavaScript) para mantener el sistema seguro, compatible y aprovechar las nuevas funcionalidades.

5. Gestión de la Documentación:

- **Actualización de la Documentación:** Se mantendrá actualizada la documentación del sistema (SRS, diseño, guías de usuario, documentación técnica) para reflejar los cambios realizados durante el mantenimiento.

6. Soporte Técnico:

- Atención a Usuarios: Se proporcionará soporte técnico a los usuarios para resolver sus dudas, problemas o consultas relacionadas con el uso del sistema.

La Fase de Mantenimiento es crucial para la sostenibilidad y el éxito a largo plazo del sistema "Asociación de Karate Barinas". Requiere un equipo dedicado o recursos asignados para atender las necesidades continuas del sistema y sus usuarios.