# Diseño del Proyecto: Mi Cadena

## 1. Objetivos del Proyecto

* **Objetivo General**: Crear una aplicación web que permita a los usuarios administrar y participar en cadenas de ahorro, facilitando la creación de cadenas, la adición de participantes y la visualización de las cadenas a las que están inscritos.
* Objetivos Específicos:
  + Desarrollar una interfaz intuitiva para la creación y gestión de cadenas de ahorro.
  + Implementar un sistema de autenticación y gestión de usuarios para proteger la privacidad y seguridad de los datos.
  + Proporcionar un panel de administración para visualizar y gestionar las cadenas y los participantes.
  + Ofrecer funcionalidades que permitan a los usuarios ver las cadenas a las que están inscritos y su estado dentro de las mismas.
  + Asegurar que la aplicación sea escalable, segura y fácil de mantener.

## 2. Requisitos Funcionales

* Autenticación y Gestión de Usuarios:
  + Los usuarios deben poder registrarse e iniciar sesión.
  + Sistema de recuperación de contraseña.
  + Verificación de correo electrónico.
* Gestión de Cadenas de Ahorro:
  + Crear una nueva cadena de ahorro definiendo parámetros como nombre, cantidad de participantes, monto de ahorro, frecuencia, etc.
  + Agregar, editar y eliminar participantes de una cadena.
  + Visualizar detalles de cada cadena, incluyendo el historial de pagos y participantes.
* Interacción del Usuario:
  + Ver las cadenas a las cuales el usuario está inscrito.
  + Ver detalles de su participación dentro de la cadena, incluyendo sus aportaciones y turno.
  + Notificaciones sobre pagos pendientes y recordatorios.
* Panel de Administración:
  + Gestión de usuarios (roles, permisos).
  + Administración de todas las cadenas creadas en la plataforma.

## 3. Requisitos Técnicos

* Tecnologías a Utilizar:
  + **Backend**: Laravel (PHP)
  + **Frontend**: Blade (Laravel) o un framework de frontend como Vue.js o React.
  + **Base de Datos**: MySQL o PostgreSQL.
  + **Autenticación**: Laravel Sanctum o Laravel Passport para la autenticación con APIs.
  + **Despliegue**: Servidor VPS con Linux, Nginx/Apache.
  + **Control de Versiones**: Git (repositorio en GitHub o GitLab).
* Integraciones:
  + Sistema de notificaciones por correo electrónico (Mailgun, SendGrid).
  + API de pagos (si se maneja la opción de pagos dentro de la plataforma).
* Seguridad:
  + Cifrado de contraseñas.
  + Protección contra ataques CSRF y XSS.
  + Auditoría de seguridad periódica.

## 4. Plan de Proyecto

### 4.1. Fase de Inicio

* **Definición de Requerimientos**: Recopilar y documentar todos los requisitos funcionales y no funcionales.
* **Estudio de Factibilidad**: Evaluar la viabilidad técnica y económica del proyecto.

### 4.2. Fase de Diseño

* **Diseño de la Base de Datos**: Crear el modelo entidad-relación (ERD) para la base de datos.
* **Diseño de la Arquitectura del Sistema**: Definir la estructura de la aplicación y los flujos de datos.
* **Diseño de Interfaz de Usuario**: Prototipado de las pantallas principales usando herramientas como Figma o Adobe XD.

### 4.3. Fase de Desarrollo

* **Configuración del Entorno de Desarrollo**: Configurar las herramientas y tecnologías necesarias.
* **Desarrollo del Backend**: Implementación de la lógica del servidor, APIs y modelo de datos.
* **Desarrollo del Frontend**: Creación de la interfaz de usuario y comunicación con el backend.

### 4.4. Fase de Pruebas

* **Pruebas Unitarias**: Verificación de la funcionalidad de componentes individuales.
* **Pruebas de Integración**: Asegurarse de que los módulos trabajen correctamente en conjunto.
* **Pruebas de Usuario**: Realizar pruebas con usuarios reales para garantizar que el sistema cumple con los requerimientos.

### 4.5. Fase de Despliegue

* **Despliegue en Producción**: Migrar la aplicación al entorno de producción.
* **Capacitación**: Entrenar a los usuarios finales y al equipo de soporte.

### 4.6. Fase de Mantenimiento

* **Soporte Post-Implementación**: Resolver cualquier problema que surja después del despliegue.
* **Actualizaciones y Mejoras**: Implementar nuevas características y mejorar las existentes basadas en el feedback de los usuarios.

#### ****5. Cronograma de Actividades****

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Fase** | **Actividad** | **Duración** | **Fecha de Inicio** | **Fecha de Fin** |
| **Inicio** | Definición de Requerimientos | 1 semana | 12/08/2024 | 18/08/2024 |
| **Inicio** | Estudio de Factibilidad | 1 semana | 19/08/2024 | 25/08/2024 |
| **Diseño** | Diseño de la Base de Datos | 2 semanas | 26/08/2024 | 08/09/2024 |
| **Diseño** | Diseño de la Arquitectura del Sistema | 2 semanas | 09/09/2024 | 22/09/2024 |
| **Diseño** | Diseño de la Interfaz de Usuario | 2 semanas | 23/09/2024 | 06/10/2024 |
| **Desarrollo** | Configuración del Entorno de Desarrollo | 1 semana | 07/10/2024 | 13/10/2024 |
| **Desarrollo** | Desarrollo del Backend | 4 semanas | 14/10/2024 | 10/11/2024 |
| **Desarrollo** | Desarrollo del Frontend | 4 semanas | 11/11/2024 | 08/12/2024 |
| **Pruebas** | Pruebas Unitarias e Integración | 2 semanas | 09/12/2024 | 22/12/2024 |
| **Pruebas** | Pruebas de Usuario | 2 semanas | 23/12/2024 | 05/01/2025 |
| **Despliegue** | Despliegue en Producción | 1 semana | 06/01/2025 | 12/01/2025 |
| **Despliegue** | Capacitación de Usuarios | 1 semana | 13/01/2025 | 19/01/2025 |
| **Mantenimiento** | Soporte Post-Implementación y Actualizaciones | Continuo | 20/01/2025 | Indefinido |