Proyecto de sistema de unidad territorial para las juntas de vecinos “Vecinos Unidos”.



Integrante: Nicolas Zamorano

Ramo: Capstone

Carrera: Ingeniería en Informática

Escuela: Informática y Telecomunicaciones

Fecha: 27/09/2024

# Índice

[**Índice 1**](#_qvs33a2q5uml)

[**Información del Proyecto 2**](#_4hg1tzclpg9x)

[**Problemática a abordar 2**](#_usnkqzhqbxqg)

[**Descripción del proyecto Vecinos Unidos. 2**](#_fi9ymyf0kn1e)

[**Propósito del plan de proyecto: 4**](#_7e8iypdp1spc)

[**Metodología KanBan: 6**](#_xgbu0yltsrcn)

[**Definición de Roles y responsabilidades 8**](#_xxkrs1806q1d)

[**Costos 9**](#_c90l4jr8rui9)

[Factibilidad Técnica: 9](#_xt9x2k5iq6ra)

[Costos: 11](#_5mzldvww8jp)

[Factibilidad Operacional: 11](#_4767bbbe1lad)

[Factibilidad Legal: 12](#_cilk8ftvk6ak)

[Fundamentales para ciberseguridad 12](#_knjv2e78qi1b)

[Sobre transformación digital y acceso a la información pública 12](#_2u9yxzc2a7er)

[Leyes de conocimiento opcional 13](#_hvdmuqgixufo)

[**Resumen de riesgos 13**](#_bju4pbrx6y8s)

[**Épicas para el proyecto 15**](#_88wp1193m55h)

[1. Gestión de Usuarios 15](#_du5nlpv7crue)

[2. Gestión de Eventos Comunitarios 15](#_9a18a3ntfzaf)

[3. Gestión de Documentos Comunitarios 15](#_50acv2up3qys)

[4. Autenticación y Seguridad 16](#_mj3fefahx2pl)

[5. Gestión de la Base de Datos 16](#_r5qprdm0kcu8)

[6. Interfaz Web Responsiva 16](#_5bphxh8u2qiz)

[7. Notificaciones y Comunicaciones 17](#_o2r2ihkgebfd)

[8. Hosting y Escalabilidad 17](#_5vnz7h5guc6x)

[9. Capacitación de Usuarios 17](#_mygqof90koy5)

[10. Pruebas e Integración Continua 17](#_bde90fw499rp)

[**Priorización de Épicas 18**](#_bfkn3ov8q8pb)

[Prioridad Alta 18](#_jjqq7a4t3mcd)

[Prioridad Media 19](#_5zyipdowi9h5)

[Prioridad Baja 20](#_hwbazivletp1)

[Prioridad Posterior al Lanzamiento 20](#_vvhn3qgp4k88)

[**Justificación de la Priorización 21**](#_6jgbiji3i7y4)

[1. Registro de Usuario y Autenticación 21](#_rjkgldz2ecbr)

[**Product Backlog 27**](#_etwwp7o0rka)

[Prioridad Alta 27](#_1lz682yx34w7)

[Prioridad Media 28](#_8tvl8im7ss6i)

[Prioridad Baja 29](#_hmahtbfh4gp6)

[Prioridad Posterior al Lanzamiento 29](#_z91pqsuk394x)

[Estimaciones de Puntos de Historia 31](#_n6ne3eeql48m)

[Prioridad Alta 31](#_8v5z7qwscmjt)

[Prioridad Posterior al Lanzamiento 32](#_wjjluyblgxbs)

[**Técnica de estimación 32**](#_dbidiehuxrzp)

[Explicación de la Técnica Utilizada 32](#_p77tngi6ec5n)

[**Sprint Planning 33**](#_d5vet83adan3)

[**Definición y estimación de tareas 34**](#_bi4cnpya0hbm)

[**Historias Usuario 34**](#_n4xxmxwo99z4)

# Información del Proyecto

## Problemática a abordar

En la actualidad, las organizaciones utilizan herramientas de tecnología de la información, TI, con el propósito de mejorar en cuanto a organización, comunicación, reducción de tiempos, costos, mano de obra, energía y recursos materiales de sus proyectos, es por esto que constantemente buscan soluciones tecnológicas.

El estado, municipalidades y juntas de vecinos no son una excepción, siendo la última la más desatendida en cuanto al uso de la tecnología. Una junta de vecinos es una organización comunitaria de carácter territorial, representativa de las personas que residen en un mismo barrio, y cuyo objeto es promover el desarrollo de la comunidad, defender los intereses y velar por los derechos de los vecinos, representándolos ante las autoridades.

Este proyecto propone una solución de implementación de tecnología que se pueda adaptar a la situación de cualquier junta de vecinos de Chile y solucione las necesidades de su directorio y los vecinos que residan en la unidad vecinal que sean mayores de 14 años.

## Descripción del proyecto Vecinos Unidos.

El proyecto a desarrollar es una aplicación web de gestión de la unidad territorial, donde podrá hacer registros de los vecinos de la villa, manejo de solicitud y emisión de certificado de residencia, se publicarán eventos de la comunidad, últimas noticias de la comuna y del país, entre otros. Esta aplicación tendrá 2 niveles de usuarios definidos como presidente de la junta de vecinos y usuario normal.

Dentro de las funciones que se apoyarán con este sistema al presidente y directorio de la junta de vecinos se encuentran:

* Manejo de inscripción de vecinos en la junta de vecinos.
* Manejo de solicitud y emisión de certificados de residencia.
* Manejo de postulación de proyectos vecinales, por parte del directorio o personal autorizado, esto es, revisión, aprobación o rechazo y envío vía email de la resolución a los solicitantes.
* Envío de notificaciones y avisos a los habitantes de la unidad territorial vía aviso tipo afiche, email y/o *WhatsApp*.
* Publicación de noticias.

Para los vecinos pertenecientes a la junta, la aplicación tendrá disponibles las siguientes funcionalidades:

* Inscripción en la junta de vecinos.
* Solicitud de certificados de residencia.
* Postulación de proyectos vecinales.
* Solicitudes de los habitantes de la unidad territorial para uso de instalaciones públicas como canchas, salas, plazas, etc. esto se gestionará con un calendario.
* Inscripción de habitantes para actividades vecinales u otros, dependerá del cupo.
* Recepción de notificaciones y avisos a los habitantes de la unidad territorial vía aviso tipo afiche, email y/o *WhatsApp*.
* Visualización de noticias.

Propósito del plan de proyecto

## Propósito del plan de proyecto:

* Mejorar la gestión de la unidad territorial de una junta de vecinos.
* Implementar un sistema de gestión de usuarios.
* Implementar un sistema de adjuntar y emitir documentos.
* Desarrollar un calendario interactivo para la gestión de eventos.
* Crear un apartado de noticias para los usuarios.

Visión del Proyecto Scrum

### **Visión del Proyecto Kanban**

**Nombre del Proyecto**: Plataforma de Gestión para Juntas de Vecinos

Beneficios Esperados

* Mejora en la comunicación: Facilitar la interacción entre vecinos y la administración a través de herramientas de mensajería y notificaciones.
* Organización de Eventos: Permitir la creación, gestión y promoción de eventos comunitarios, mejorando la participación vecinal.
* Gestión Documental: Proveer un sistema para almacenar y compartir documentos importantes de la comunidad.
* Accesibilidad: Crear una aplicación responsiva que pueda ser utilizada en diferentes dispositivos, asegurando que todos los miembros puedan acceder a la información relevante.

Usuarios Objetivo

* Cualquier persona mayor o igual a 18 años que resida en las unidades vecinales.
* Miembros del directorio de las juntas de vecinos y administradores municipales.

Criterios de Éxito

1. Tasa de Adopción: Al menos el 70% de los miembros de la comunidad registrados en la plataforma dentro de los primeros seis meses.
2. Satisfacción del Usuario: Una puntuación de satisfacción del usuario de al menos 4/5 en encuestas post-uso.
3. Número de Eventos: Al menos 10 eventos gestionados a través de la plataforma en el primer año.
4. Acceso y Uso: Proporción de usuarios activos que utilizan la plataforma al menos una vez a la semana.

Alcance del Proyecto

* Desarrollo de una aplicación web responsiva con funcionalidades de gestión de usuarios, eventos y documentos.
* Implementación de un sistema de autenticación seguro.
* Capacitación para los usuarios en el uso de la plataforma.
* Pruebas y validación de la plataforma antes de su lanzamiento.

Entregables Clave

* MVP (Producto Mínimo Viable): Aplicación funcional con características básicas de gestión.
* Documentación del Usuario: Manuales y guías de uso para los miembros de la junta de vecinos.
* Capacitación: Sesiones de formación para los usuarios sobre cómo utilizar la plataforma.

Estrategia de Implementación Kanban

* Tablero Kanban: Utilizaremos un tablero Kanban para visualizar el flujo de trabajo, categorizando las tareas en columnas (Por Hacer, En Progreso, Completado).
* Límite de Trabajo en Progreso (WIP): Establecer límites para el trabajo en progreso en cada columna, lo que ayuda a evitar la sobrecarga del equipo.
* Revisión Continua: Realizaremos revisiones periódicas para ajustar las prioridades y garantizar que se estén abordando las tareas más críticas.
* Mejoras Continuas: Fomentaremos una cultura de retroalimentación constante para identificar áreas de mejora en el flujo de trabajo y la gestión de tareas.

## Alcance del proyecto

1. **Objetivo**: Implementar un sistema de gestión de usuarios.

* **Específico**: Desarrollar un sistema de registro, autenticación y administración de roles para los usuarios.
* **Medible**: El sistema debe permitir la creación, modificación y eliminación de usuarios, con al menos dos roles de acceso (administrador y usuario estándar).
* **Alcanzable**: Utilizando *Node.js/Express* y bases de datos como *PostgreSQL*, se pueden implementar librerías y *APIs* preexistentes para gestionar usuarios y autenticación.
* **Relevante**: La gestión de usuarios es clave para controlar los accesos a los documentos, eventos y noticias dentro de la aplicación.
* **Tiempo**: Este sistema debe estar completamente funcional en un plazo de 3 meses.

**2.** **Objetivo**: Implementar un sistema de adjuntar y emitir documentos

* **Específico**: Permitir a los usuarios adjuntar documentos (PDF, imágenes, etc.) y emitir documentos generados desde la plataforma.
* **Medible**: El sistema debe permitir al menos 100 cargas de documentos por usuario, con una capacidad de almacenamiento de hasta 1 GB por usuario.
* **Alcanzable**: Utilizando servicios de almacenamiento en la nube como *Firebase Storage* o *Amazon S3*.
* **Relevante**: El manejo de documentos es esencial para el propósito del proyecto, ya que permite almacenar y compartir información.
* **Tiempo**: El sistema debe estar operando en un plazo de 2 meses.

**3. Objetivo:** Desarrollar un calendario interactivo para la gestión de eventos

* **Específico**: Implementar un calendario interactivo que permita a los usuarios crear, modificar y eliminar eventos.
* **Medible**: El calendario debe permitir al menos 50 eventos por mes, con visualización semanal, mensual y diaria.
* **Alcanzable**: Usando una biblioteca como *FullCalendar* y una base de datos relacional para gestionar los eventos.
* **Relevante**: La funcionalidad del calendario es crucial para la planificación y seguimiento de eventos dentro de la aplicación.
* **Tiempo**: El calendario debe estar en funcionamiento en un plazo de 1 mes.

**4. Objetivo:** Crear un apartado de noticias para los usuarios

* **Específico**: Desarrollar un módulo de noticias donde los administradores puedan publicar artículos y los usuarios puedan visualizarlos.
* **Medible**: El sistema debe permitir la creación de al menos 10 noticias al mes, con opciones de filtro por fecha y categoría.
* **Alcanzable**: Utilizando el *framework React* para el *frontend* y *Node.js/Express* para el *backend*, junto con la base de datos seleccionada.
* **Relevante**: Un apartado de noticias permitirá mantener informados a los usuarios y es una herramienta esencial de comunicación dentro de la plataforma.
* **Tiempo**: El módulo debe estar finalizado en un plazo de 2 meses.

## **Metodología KanBan:**

La estrategia escogida para este proyecto es la metodología kanban ya que permite controlar el flujo de trabajo y manejar los tiempos dedicados a cada actividad.



## Definición de Roles y responsabilidades

| **Rol** | **Responsabilidades Clave** | **Metas** |
| --- | --- | --- |
| Product Owner (PO) | - Priorización y refinamiento continuo de historias de usuario en el tablero Kanban. | - Asegurar que las tareas completadas agreguen valor para el usuario. |
| Kanban Facilitator | - Optimización del flujo y reducción de bloqueos en el tablero. | - Mantener un flujo de trabajo constante y eficiente. |
| Desarrollador Backend | - Desarrollo de lógica de negocio, API, autenticación y conexión con PostgreSQL, y pruebas unitarias. | - Proveer un backend seguro y funcional, reduciendo tiempos de espera en su área. |
| Desarrollador Frontend | - Desarrollo y optimización de la interfaz de usuario, respondiendo a los cambios en la fila de trabajo. | - Facilitar una experiencia del usuario fluida. |
| Administrador de Base de Datos | - Diseño y optimización de la base de datos, permisos y copias de seguridad en PostgreSQL. | - Asegurar la disponibilidad y seguridad de los datos. |
| QA/Tester | - Ejecución de pruebas y documentación continua de errores en la medida que se completen las funcionalidades. | - Garantizar un producto sin errores al avanzar el flujo. |
| Administrador de Infraestructura | - Configuración y monitoreo de entorno de despliegue escalable y seguro. | - Asegurar alta disponibilidad y escalabilidad en el entorno de producción. |

## Costos

### Factibilidad Técnica:

| **Componente** | **Tecnologías/Frameworks Recomendados** | **Ventajas Técnicas** | **Factibilidad** |
| --- | --- | --- | --- |
| Frontend: Aplicación Web Responsiva | *React, Vue.js, Angular*  *UI Libraries: Bootstrap, Material UI* | - Compatibles con múltiples dispositivos (PC, tablets, móviles).  - Librerías de *UI* para crear interfaces amigables y responsivas. | Alta |
| Backend: Gestión de usuarios, eventos y documentos | *Node.js/Express*  *Python/Django* | - *Node.js/Express*: Rápido, no bloqueante y escalable.  - *Django*: Estructurado, seguro, ideal para proyectos que necesitan autenticación avanzada. | Alta |
| Gestión de Eventos con Calendario | *FullCalendar (React, Vue.js, Angular)* | - Funcionalidades listas para mostrar, crear y modificar eventos en un calendario.  - Fácil integración con base de datos. | Alta |
| Gestión de Documentos | *Amazon S3, Firebase Storage,* Almacenamiento en servidor propio | - Integración nativa con *Node.js/Django*.  - Manejo estándar de archivos (PDF, imágenes), asociado a usuarios. | Alta |
| Base de Datos | *PostgreSQL (*relacional*)* | - Escalables.  - Integración sencilla con *ORMs (Sequelize, Mongoose*) en *Node.js* o *Django ORM*.  - Herramientas robustas de administración. | Alta |
| Autenticación y Seguridad | *OAuth, JWT (JSON Web Tokens), Auth0* | - Autenticación y autorización segura.  - *Django* ofrece un sistema de autenticación integrado y confiable. | Alta |
| Hosting y Escalabilidad | *Heroku, AWS, DigitalOcean, Firebase Hosting* | - Despliegue sencillo sin necesidad de administrar servidores.  - Escalabilidad según el crecimiento de la aplicación. | Alta |

**Factibilidad técnica general**: **Alta**.

* Las tecnologías seleccionadas son ampliamente usadas en la industria, tienen una buena comunidad de soporte y son conocidas por su robustez, flexibilidad y capacidad para crecer con el proyecto.

**Escalabilidad**: El proyecto puede empezar de manera modesta con herramientas gratuitas o de bajo costo, y luego escalar a plataformas más avanzadas a medida que crece el número de usuarios y los requerimientos.

### **Costos:**

**Software:**

| **Opción** | **Escenario Económico** | **Escenario Escalable** |
| --- | --- | --- |
| **Desarrollo** | Gratis | Gratis |
| **Base de Datos** | *Firebase* gratis (limitado) o *MySQL/PostgreSQL* en servidor propio | *Amazon RDS* o *MongoDB Atlas* (USD 9-15/mes) |
| **Alojamiento** | *Heroku* gratuito o *DigitalOcean* (USD 5/mes) | *AWS EC2 o Google Cloud* (USD 10-20/mes) |
| **Almacenamiento de Documentos** | *Firebase Storage* gratuito | *Amazon S3* (USD 0.023/GB) |
| **SSL y seguridad** | Gratis (*Let's Encrypt*) | Gratis (*Let's Encryp*t) |
| **Total Aproximado** | USD 0 a USD 10/mes | USD 30 a USD 50/mes (con recursos más escalables) |

**Hardware:**

Parael acceso del usuario a la aplicación se requiere de conexión a internet

### **Factibilidad Operacional:**

| Cargo | Área | Sueldo Mensual | Sueldo por hora | Sueldo Mensual UF | Sueldo por hora UF |
| --- | --- | --- | --- | --- | --- |
| Desarrollador Full Stack | IT | $1.800.000 | $10.227 | UF 48 | UF0,27 |
| Desarrollador de Base de datos | IT | $1.348.077 | $7.660 | UF 36 | UF0,20 |
| Jefe de proyecto | IT | $1.500.000 | $8.523 | UF 40 | UF0,22 |
| Experto *QA* | IT | $1.200.000 | $6.818 | UF 32 | UF0,18 |
| Diseñador Web *UI/Ux* | IT | $645.000 | $3.665 | UF 17 | UF0,10 |
| TOTAL |  | $6.493.077 | $36.892 | UF 171 | UF0,97 |

### **Factibilidad Legal:**

#### Fundamentales para ciberseguridad

1. Ley N°21.663/2024: Esta ley tiene por objeto regular la normativa general aplicable a las acciones de ciberseguridad de los organismos del Estado, ya sea entre ellos o con entidades privadas. Asimismo, establece los requisitos mínimos para enfrentar incidentes de ciberseguridad, las atribuciones y obligaciones de los organismos del Estado, los deberes de las instituciones determinadas en la ley, como asimismo los mecanismos de control, supervisión y responsabilidad frente a infracciones.
2. Ley N°21.459/2022: Establece normas sobre delitos informáticos, deroga la Ley N° 19.223 y modifica otros cuerpos legales con el objeto de adecuarlos al Convenio de Budapest. Esta ley establece los delitos informáticos reconocidos en Chile; su conocimiento es esencial para un encargado de ciberseguridad.
3. Ley N°19.628/1999: Sobre protección de la vida privada. Esta ley establece qué es un dato personal, un dato personal sensible, y qué condiciones debe cumplir toda entidad que maneje estos datos. Algunos servicios públicos manejan datos personales o sensibles; en estos servicios, el conocimiento de esta ley para el encargado de ciberseguridad es fundamental.
4. Ley N°19.799/2002: Ley sobre documentos electrónicos, firma electrónica y servicios de certificación de dicha firma.

#### **Sobre transformación digital y acceso a la información pública**

1. Ley N°19.880/2003: Ley que establece bases de los procedimientos administrativos que rigen los actos de los órganos de la administración del Estado. Esta ley es importante porque establece condiciones sobre los plazos y la forma en que los servicios públicos deben responder frente a requerimientos de ciudadanos y de otros servicios públicos. Estas demandas usualmente deben ser satisfechas con plataformas de software, y existen consideraciones importantes de ciberseguridad que estas plataformas deben cumplir.
2. Ley N°21.180/2022: Transformación digital del Estado. Esta ley transforma varias leyes para fomentar el avance en digitalización de los procesos públicos. En particular, modifica la ley 19.880 para permitir la tramitación digital de los actos administrativos. En la práctica, no es necesario leer esta ley completa, pues en el link de la ley 19.880 en la Biblioteca del Congreso Nacional los cambios que esta ley introduce en la ley 19.880 ya aparecen reflejados.
3. Ley N°20.285/2008: Ley sobre acceso a la información pública. Esta ley es importante porque establece condiciones sobre la publicación de información que toda institución pública debe cumplir.

#### **Leyes de conocimiento opcional**

1. Ley N°19.927/2004: Ley modifica códigos penales en materia de delitos sobre pornografía infantil.
2. Ley N°17.336/1970: Sobre propiedad intelectual. Fija condiciones para la protección de los programas computacionales como propiedad intelectual de sus autores.
3. DFL 3/2021: Fija el texto refundido, coordinado y sistematizado de la Ley N° 19.496, que establece normas sobre protección de los derechos de los consumidores.

## Resumen de riesgos

| **Nombre del Riesgo** | **Descripción** | **Nivel de Impacto** | **Nivel de Ocurrencia** | **Plan de Contingencia** | **Plan de Mitigación** |
| --- | --- | --- | --- | --- | --- |
| Compatibilidad Tecnológica | Posibles problemas de compatibilidad con PostgreSQL y tecnologías usadas, dificultando la integración. | Alto | Medio | Mantener documentación detallada para identificar soluciones rápidas de compatibilidad. | Evaluar compatibilidad antes de la implementación y realizar pruebas de integración tempranas. |
| Gestión de Flujo Kanban | Riesgo de cuellos de botella si se acumulan demasiadas tareas en progreso en el flujo Kanban. | Medio | Alto | Monitorear el flujo para redistribuir tareas y evitar acumulación de trabajo. | Control de tareas en progreso en Kanban y ajuste constante del flujo. |
| Dependencia de Recursos Humanos | Dependencia en miembros clave del equipo puede afectar la continuidad si no están disponibles. | Alto | Alto | Asegurar que cada rol tenga un respaldo que pueda tomar sus tareas si es necesario. | Capacitar a múltiples miembros en áreas críticas para evitar dependencia. |
| Control de Alcance | Incremento de alcance sin ajuste de recursos o tiempo puede retrasar la entrega. | Alto | Alto | Definir alcance con precisión y establecer límites claros desde el inicio. | Revisar y ajustar el alcance periódicamente para mantener el control. |
| Seguridad y Escalabilidad | Implementar medidas de seguridad y escalabilidad puede generar costos y limitar flexibilidad. | Alto | Medio | Realizar pruebas de seguridad tempranas y ajustar según los resultados iniciales. | Implementar medidas de seguridad básicas y ajustarlas según la escalabilidad. |
| Gestión del Conocimiento | Sin gestión adecuada de documentación y capacitación, se dificultará la adopción y mantenimiento. | Medio | Medio | Plan de capacitación para los usuarios finales y documentación accesible. | Documentar cada proceso importante y capacitar a los usuarios en el uso. |

## Épicas para el proyecto

### 1. Gestión de Usuarios

**Descripción**: Permitir a los usuarios registrarse, iniciar sesión y gestionar su información personal. Esta épica garantiza que solo usuarios autorizados y mayores de 18 años puedan acceder a las funcionalidades del sistema.

* **Subtareas**:
  + Implementar formulario de registro y validación de edad.
  + Crear sistema de autenticación (login/logout).
  + Gestión de roles y permisos de usuarios.
  + Recuperación de contraseña y cambio de información personal.

### **2. Gestión de Eventos Comunitarios**

**Descripción**: Facilitar la creación, visualización y gestión de eventos comunitarios. Los usuarios podrán ver un calendario interactivo con eventos programados por la junta de vecinos.

* **Subtareas**:
  + Crear módulo de calendario interactivo.
  + Permitir a los administradores la creación y modificación de eventos.
  + Habilitar la inscripción a eventos para los usuarios.
  + Notificaciones y recordatorios de eventos.

### **3. Gestión de Documentos Comunitarios**

**Descripción**: Proveer un sistema para la publicación y gestión de documentos importantes, como actas de reuniones, estatutos y comunicados. Estos documentos serán accesibles a los usuarios autorizados.

* **Subtareas**:
  + Crear módulo de carga y descarga de documentos.
  + Definir permisos para acceso a documentos.
  + Implementar sistema de búsqueda de documentos por categorías o palabras clave.

### **4. Autenticación y Seguridad**

**Descripción**: Asegurar que el sistema sea seguro, protegiendo los datos personales de los usuarios y restringiendo el acceso a funcionalidades según los permisos establecidos.

* **Subtareas**:
  + Implementar autenticación segura (e.g., hashes de contraseñas).
  + Proteger las rutas y funcionalidades con permisos específicos (roles).
  + Implementar cifrado en la base de datos para información sensible.

### **5. Gestión de la Base de Datos**

**Descripción**: Asegurar que la base de datos esté bien estructurada y permita un acceso eficiente a la información almacenada sobre los usuarios, eventos y documentos.

* **Subtareas**:
  + Crear la estructura de la base de datos para usuarios, eventos y documentos.
  + Optimizar consultas a la base de datos.
  + Implementar backups automáticos y planes de recuperación ante fallos.

### **6. Interfaz Web Responsiva**

**Descripción**: Desarrollar una interfaz web que se adapte a diferentes dispositivos, facilitando que los usuarios puedan acceder al sistema desde computadoras, tabletas o teléfonos móviles.

* **Subtareas**:
  + Diseñar interfaz de usuario adaptativa (Responsive Design).
  + Asegurar compatibilidad con diferentes navegadores.
  + Mejorar la experiencia del usuario mediante un diseño intuitivo.

### **7. Notificaciones y Comunicaciones**

**Descripción**: Implementar un sistema de notificaciones que informe a los usuarios sobre nuevos eventos, documentos importantes o cambios en su cuenta.

* **Subtareas**:
  + Desarrollar notificaciones internas dentro de la plataforma.
  + Configurar envío de correos electrónicos para notificaciones críticas.
  + Habilitar la suscripción y cancelación de notificaciones por parte del usuario.

### **8. Hosting y Escalabilidad**

**Descripción**: Asegurar que la aplicación esté correctamente alojada y pueda escalar en función del número de usuarios y actividades sin comprometer el rendimiento.

* **Subtareas**:
  + Seleccionar el proveedor de hosting adecuado.
  + Implementar soluciones de escalabilidad (e.g., balanceo de carga).
  + Monitorear el rendimiento de la aplicación en tiempo real.

### **9. Capacitación de Usuarios**

**Descripción**: Asegurar que los usuarios sepan cómo utilizar las funcionalidades principales del sistema a través de capacitaciones, tutoriales y manuales.

* **Subtareas**:
  + Crear tutoriales en línea para los usuarios.
  + Generar documentación y manuales de usuario.
  + Ofrecer sesiones de capacitación para usuarios administradores.

### **10. Pruebas e Integración Continua**

**Descripción**: Realizar pruebas unitarias, de integración y de usuario para asegurar que todas las funcionalidades del sistema funcionan correctamente y se integran de manera adecuada.

* **Subtareas**:
  + Desarrollar un plan de pruebas unitarias para las funciones clave.
  + Ejecutar pruebas de integración para garantizar la interoperabilidad.
  + Realizar pruebas de usabilidad con usuarios reales.

## Priorización de Épicas

### Prioridad Alta

1. **Autenticación y Seguridad**
   * **Descripción**: Implementar la autenticación segura y la gestión de permisos es crucial para garantizar que solo usuarios autorizados, mayores de 18 años, puedan acceder a la plataforma.
   * **Razón**: Sin un sistema de autenticación robusto, la plataforma carecería de seguridad y control de acceso, exponiendo los datos a posibles amenazas.
   * **Dependencias**: Ninguna directa (puede desarrollarse en paralelo con la gestión de usuarios).
2. **Gestión de Usuarios**
   * **Descripción**: Esta épica se enfoca en permitir a los usuarios registrarse, iniciar sesión y gestionar su información personal.
   * **Razón**: Es la base para que el resto de las funcionalidades del sistema puedan ser utilizadas por los usuarios.
   * **Dependencias**: Depende del sistema de autenticación para permitir la gestión segura de los usuarios.
3. **Gestión de la Base de Datos**
   * **Descripción**: Asegurar que la base de datos esté bien estructurada y que los datos se almacenen de manera eficiente.
   * **Razón**: La base de datos es fundamental para que todas las demás funcionalidades (usuarios, eventos, documentos) se gestionen adecuadamente.
   * **Dependencias**: Es crucial para las demás épicas, ya que todas requieren un almacenamiento adecuado de datos.

### Prioridad Media

1. **Gestión de Eventos Comunitarios**
   * **Descripción**: Permitir la creación, visualización y gestión de eventos comunitarios, incluyendo el calendario interactivo.
   * **Razón**: Los eventos son una de las principales funcionalidades de la plataforma, ya que promueven la participación activa de la comunidad.
   * **Dependencias**: Requiere que la gestión de usuarios esté en funcionamiento para restringir la creación de eventos a usuarios autorizados.
2. **Gestión de Documentos Comunitarios**
   * **Descripción**: Facilitar la gestión y acceso a documentos relevantes para los usuarios, como actas, estatutos y comunicados.
   * **Razón**: Es importante para que los usuarios puedan acceder a documentos clave, pero depende menos de la interacción constante en comparación con los eventos.
   * **Dependencias**: Requiere que la autenticación y la gestión de usuarios estén completas para controlar el acceso a los documentos.
3. **Interfaz Web Responsiva**
   * **Descripción**: Desarrollar una interfaz adaptativa que permita el acceso desde dispositivos móviles, tabletas y computadoras.
   * **Razón**: Garantiza una buena experiencia de usuario en diferentes dispositivos, lo cual es esencial, pero no crítico en las primeras fases de desarrollo.
   * **Dependencias**: Requiere que las funcionalidades principales (usuarios, eventos, documentos) estén operativas para poder probar la interfaz en condiciones reales.

### Prioridad Baja

1. **Notificaciones y Comunicaciones**
   * **Descripción**: Implementar un sistema de notificaciones internas y externas (por correo electrónico) para alertar a los usuarios sobre nuevos eventos o documentos.
   * **Razón**: Es una mejora valiosa para la experiencia del usuario, pero la plataforma puede operar sin esta funcionalidad en las fases iniciales.
   * **Dependencias**: Depende de la gestión de eventos y documentos para enviar notificaciones pertinentes.
2. **Pruebas e Integración Continua**
   * **Descripción**: Realizar pruebas unitarias y de integración para asegurar la calidad del código y la interoperabilidad entre componentes.
   * **Razón**: Las pruebas son esenciales para mantener la calidad del software, pero las pruebas iniciales pueden ejecutarse una vez que las funcionalidades clave estén listas.
   * **Dependencias**: Depende del desarrollo de todas las funcionalidades críticas.
3. **Hosting y Escalabilidad**
   * **Descripción**: Asegurar que la aplicación esté correctamente alojada y pueda escalar conforme crece el número de usuarios.
   * **Razón**: Importante para la estabilidad del sistema en el futuro, pero en las primeras fases, es más prioritario asegurar que las funcionalidades principales estén operativas.
   * **Dependencias**: Requiere que las funcionalidades principales estén listas y probadas para desplegar el sistema completo.

### **Prioridad Posterior al Lanzamiento**

1. **Capacitación de Usuarios**

* **Descripción**: Crear tutoriales, manuales y ofrecer sesiones de capacitación para los usuarios administradores.
* **Razón**: Esta épica es útil para facilitar la adopción del sistema, pero puede implementarse después de que el sistema esté listo y funcionando.
* **Dependencias**: Depende de que el sistema esté completamente funcional para poder ofrecer una capacitación adecuada.

## Justificación de la Priorización

* Autenticación y Gestión de Usuarios son las primeras épicas porque el control de acceso y la gestión de las personas que usarán la plataforma son fundamentales para el funcionamiento seguro del sistema.
* Base de Datos es también de alta prioridad, ya que es el núcleo donde se almacenarán todos los datos del sistema.
* Las funcionalidades principales, como la gestión de eventos y documentos, siguen en prioridad media, ya que representan las acciones que los usuarios realizan con más frecuencia.
* Pruebas, notificaciones, y hosting tienen prioridad media/baja, ya que aunque son importantes para el rendimiento y la escalabilidad del sistema, no son críticas en las primeras fases.
* Finalmente, la capacitación de usuarios y el hosting son importantes, pero pueden abordarse después de que el sistema esté completamente desarrollado y probado.

Definición de Historias de Usuario

### **1. Registro de Usuario y Autenticación**

**Historia 1: Registro de un usuario mayor de 18 años**

* **Como**: usuario mayor de 18 años,
* **Quiero**: registrarme en el sistema proporcionando mi nombre, correo electrónico, y fecha de nacimiento,
* **Para**: poder acceder a las funcionalidades de la plataforma como eventos y documentos comunitarios.

**Criterios de aceptación**:

* El sistema debe validar que el usuario tiene 18 años o más.
* El usuario debe recibir un correo electrónico de confirmación para activar su cuenta.

**Historia 2: Iniciar sesión en el sistema**

* **Como**: usuario registrado,
* **Quiero**: iniciar sesión con mi correo electrónico y contraseña,
* **Para**: acceder a mi cuenta y gestionar mis datos personales, eventos y documentos.

**Criterios de aceptación**:

* El sistema debe permitir el inicio de sesión con correo y contraseña.
* El sistema debe mostrar un mensaje de error si las credenciales son incorrectas.

**Historia 3: Recuperación de contraseña**

* **Como**: usuario registrado,
* **Quiero**: recuperar mi contraseña en caso de haberla olvidado,
* **Para**: poder restablecerla y acceder nuevamente al sistema.

**Criterios de aceptación**:

* El sistema debe enviar un enlace de recuperación de contraseña al correo del usuario.
* El enlace debe permitir al usuario restablecer su contraseña de manera segura.

**2. Gestión de Eventos Comunitarios**

**Historia 4: Crear un evento comunitario**

* **Como**: administrador de la junta de vecinos,
* **Quiero**: crear un nuevo evento en el calendario comunitario,
* **Para**: que los vecinos puedan ver la información del evento y registrarse para participar.

**Criterios de aceptación**:

* El sistema debe permitir al administrador ingresar el título, descripción, fecha, hora y ubicación del evento.
* El evento debe ser visible en el calendario para todos los usuarios registrados.

**Historia 5: Ver eventos comunitarios en el calendario**

* **Como**: usuario registrado,
* **Quiero**: ver un calendario interactivo de los eventos comunitarios programados,
* **Para**: poder estar al tanto de los próximos eventos y participar si es necesario.

**Criterios de aceptación**:

* El calendario debe mostrar todos los eventos creados por el administrador.
* El usuario debe poder hacer clic en un evento para ver más detalles.

**Historia 6: Inscribirme a un evento**

* **Como**: usuario registrado,
* **Quiero**: inscribirme en un evento comunitario,
* **Para**: confirmar mi participación en el evento y recibir notificaciones sobre él.

**Criterios de aceptación**:

* El sistema debe permitir al usuario inscribirse con un solo clic.
* El usuario debe recibir una confirmación de su inscripción y la opción de cancelar si lo desea.

**3. Gestión de Documentos Comunitarios**

**Historia 7: Subir un documento para la comunidad**

* **Como**: Administrador de la junta de vecinos,
* **Quiero**: Subir documentos importantes (actas, estatutos, comunicados) a la plataforma,
* **Para**: Los vecinos tengan acceso a información relevante en todo momento.

**Criterios de aceptación**:

* El sistema debe permitir al administrador subir documentos en formato PDF u otros formatos comunes.
* Los documentos deben estar organizados por categorías y ser visibles solo para los usuarios autorizados.

**Historia 8: Descargar documentos comunitarios**

* **Como**: usuario registrado,
* **Quiero**: descargar documentos comunitarios importantes,
* **Para**: El usuario pueda revisarlos en su dispositivo cuando sea necesario.

**Criterios de aceptación**:

* El sistema debe permitir a los usuarios descargar los documentos relevantes.
* Los usuarios deben ver un listado de documentos disponibles con fechas y títulos claros.

**4. Autenticación y Seguridad**

**Historia 9: Autorización de roles y permisos**

* **Como**: administrador de la junta de vecinos,
* **Quiero**: poder asignar roles a diferentes usuarios (vecinos, miembros del directorio, administradores),
* **Para**: que solo ciertos usuarios tengan permisos para realizar tareas específicas, como crear eventos o subir documentos.

**Criterios de aceptación**:

* El sistema debe permitir asignar y cambiar roles de usuario (administrador, miembro, vecino).
* Las funcionalidades restringidas deben ser accesibles solo para los usuarios con los permisos correspondientes.

**5. Notificaciones y Comunicaciones**

**Historia 10: Recibir notificaciones sobre eventos y documentos**

* **Como**: usuario registrado,
* **Quiero**: recibir notificaciones sobre nuevos eventos y documentos importantes en la plataforma,
* **Para**: estar informado de cualquier actividad relevante en mi comunidad.

**Criterios de aceptación**:

* El sistema debe notificar al usuario por correo electrónico o en la plataforma sobre la creación de nuevos eventos o documentos.
* El usuario debe poder gestionar sus preferencias de notificación (activar/desactivar).

**6. Interfaz Web Responsiva**

**Historia 11: Acceder a la plataforma desde diferentes dispositivos**

* **Como**: Usuario registrado,
* **Quiero**: Acceder a la plataforma desde mi computadora, teléfono móvil o tableta,
* **Para**: Pueda utilizarla de manera cómoda sin importar el dispositivo.

**Criterios de aceptación**:

* La plataforma debe ser completamente responsiva y adaptarse a diferentes tamaños de pantalla.
* El diseño debe ser claro y fácil de navegar en dispositivos móviles.

**7. Pruebas e Integración Continua**

**Historia 12: Probar las funcionalidades del sistema**

* **Como**: equipo de desarrollo,
* **Quiero**: realizar pruebas unitarias e integrales de las funcionalidades del sistema,
* **Para**: asegurar que todo funcione correctamente antes del despliegue final.

**Criterios de aceptación**:

* Deben existir pruebas unitarias para cada funcionalidad crítica (autenticación, eventos, documentos).
* Deben realizarse pruebas de integración para asegurar que los componentes interactúan correctamente.

**8. Hosting y Escalabilidad**

**Historia 13: Desplegar la plataforma en un entorno escalable**

* **Como**: administrador del sistema,
* **Quiero**: que la plataforma está alojada en un entorno escalable,
* **Para**: asegurarse de que pueda manejar el crecimiento en el número de usuarios y actividades sin perder rendimiento.

**Criterios de aceptación**:

* La plataforma debe estar alojada en un servicio de hosting confiable.
* Debe ser posible escalar el servicio para manejar mayor tráfico si es necesario.

**9. Capacitación de Usuarios**

**Historia 14: Acceder a tutoriales y manuales de uso**

* **Como**: usuario nuevo en la plataforma,
* **Quiero**: tener acceso a tutoriales y manuales de uso,
* **Para**: aprender cómo utilizar todas las funcionalidades del sistema.

**Criterios de aceptación**:

* El sistema debe tener una sección de ayuda con tutoriales escritos y videos.
* El usuario debe poder acceder fácilmente a esta sección desde cualquier parte de la plataforma.

## Product Backlog

#### **Prioridad Alta**

1. **Autenticación y Seguridad**
   * Historia de usuario: Registro de un usuario mayor de 18 años
     + Crear formulario de registro.
     + Implementar validación de edad (≥ 18 años).
     + Configurar envío de correo de confirmación.
     + Crear lógica de activación de cuenta.
   * Historia de usuario: Iniciar sesión en el sistema
     + Crear formulario de inicio de sesión.
     + Implementar verificación de credenciales.
     + Desplegar mensajes de error en caso de fallos.
   * Historia de usuario: Recuperación de contraseña
     + Implementar sistema de recuperación de contraseña.
     + Configurar envío de enlace de restablecimiento.
     + Crear lógica de restablecimiento seguro de contraseña.
2. **Gestión de Usuarios**
   * Historia de usuario: Autorización de roles y permisos
   * Implementar lógica de roles (administrador, vecino, miembro del directorio).
   * Desarrollar interfaz de gestión de roles para administradores.
   * Restringir acceso a funcionalidades según el rol.
   * Historia de usuario: Inscribirme a un evento
   * Crear sistema de inscripción a eventos.
   * Desplegar confirmación de inscripción al usuario.
     + Permitir cancelación de inscripción.
3. **Gestión de la Base de Datos**
   * Historia de usuario: Estructurar la base de datos para usuarios
   * Crear tabla para almacenar datos de usuario (nombre, correo, fecha de nacimiento, roles).
   * Implementar validación y reglas de integridad referencial.
   * Probar la conexión a la base de datos con PostgreSQL.

#### **Prioridad Media**

1. **Gestión de Eventos Comunitarios**
   * Historia de usuario: Crear un evento comunitario
   * Crear formulario para la creación de eventos.
   * Implementar validaciones (título, descripción, fecha y hora).
   * Desarrollar integración con calendario.
   * Historia de usuario: Ver eventos comunitarios en el calendario
   * Desarrollar visualización de calendario interactivo.
   * Permitir clics en eventos para mostrar detalles.
   * Mostrar eventos pasados, actuales y futuros de manera diferenciada.
2. **Gestión de Documentos Comunitarios**
   * Historia de usuario: Subir un documento para la comunidad
     + Crear formulario para subir documentos.
     + Implementar validación de formatos permitidos (PDF, Word, etc.).
     + Desarrollar clasificación y categorización de documentos.
   * Historia de usuario: Descargar documentos comunitarios
     + Implementar funcionalidad de descarga de documentos.
     + Mostrar lista de documentos disponibles con títulos y fechas.
     + Validar que solo usuarios registrados puedan descargar.
3. **Interfaz Web Responsiva**
   * Historia de usuario: Acceder a la plataforma desde diferentes dispositivos
     + Desarrollar diseño responsivo que se adapte a móviles, tablets y desktops.
     + Implementar pruebas de interfaz en diferentes dispositivos.
     + Ajustar estilos CSS y mejorar la navegación en pantallas pequeñas.

#### **Prioridad Baja**

1. **Notificaciones y Comunicaciones**
   * Historia de usuario: Recibir notificaciones sobre eventos y documentos
     + Implementar sistema de notificaciones internas en la plataforma.
     + Configurar envíos de correo electrónico automáticos.
     + Permitir a los usuarios gestionar sus preferencias de notificaciones.
2. **Pruebas e Integración Continua**

* Historia de usuario: Probar las funcionalidades del sistema
  + Implementar pruebas unitarias para las principales funcionalidades (registro, eventos, documentos).
  + Realizar pruebas de integración entre componentes (usuarios, base de datos, eventos).
  + Desarrollar pruebas de carga para asegurar el rendimiento del sistema bajo presión.

1. **Hosting y Escalabilidad**

* Historia de usuario: Desplegar la plataforma en un entorno escalable
  + Seleccionar servicio de hosting adecuado.
  + Configurar servidor de base de datos y web.
  + Implementar estrategias de escalabilidad (caching, balanceo de carga, etc.).

#### **Prioridad Posterior al Lanzamiento**

1. **Capacitación de Usuarios**

* Historia de usuario: Acceder a tutoriales y manuales de uso
  + Crear sección de ayuda en la plataforma.
  + Desarrollar manuales y tutoriales en formato escrito y video.
  + Implementar enlaces a los recursos de ayuda en las páginas principales.

1. **Marcha Blanca**

* Historia de usuario: Realizar pruebas de marcha blanca con usuarios reales
  + Organizar una fase de prueba con un grupo limitado de usuarios.
  + Monitorear el uso y rendimiento del sistema.
  + Recopilar feedback y corregir errores antes del despliegue final.

**Justificación del Orden en el Product Backlog**

* Las funcionalidades de autenticación, seguridad y gestión de usuarios tienen la máxima prioridad porque garantizan que el sistema sea seguro y que solo usuarios autorizados puedan interactuar con él.
* Las funciones esenciales como la gestión de eventos, documentos y la base de datos vienen inmediatamente después, ya que representan el corazón del sistema en cuanto a su funcionalidad comunitaria.
* Las mejoras en la interfaz web responsiva y notificaciones tienen una prioridad media, ya que mejoran la experiencia del usuario, pero no son críticas para el funcionamiento inicial.
* Pruebas, integración, hosting y escalabilidad son importantes pero pueden implementarse de manera progresiva a medida que el sistema está cerca de ser lanzado o una vez esté en producción.
* Capacitación de usuarios y la marcha blanca son esenciales para la adopción del sistema, pero se llevarán a cabo una vez que el sistema esté casi completo.

Estimación de puntos de historia.

### **Estimaciones de Puntos de Historia**

#### **Prioridad Alta**

1. **Autenticación y Seguridad**
   * Registro de usuario mayor de 18 años: **8 puntos**
   * Inicio de sesión en el sistema: **5 puntos**
   * Recuperación de contraseña: **5 puntos**
2. **Gestión de Usuarios**
   * Autorización de roles y permisos: **8 puntos**
   * Inscribirse en un evento: **3 puntos**
3. **Gestión de la Base de Datos**
   * Estructuración de la base de datos para usuarios: **8 puntos**

**Prioridad Media**

1. **Gestión de Eventos Comunitarios**
   * Crear un evento comunitario: **8 puntos**
   * Ver eventos en el calendario: **5 puntos**
2. **Gestión de Documentos Comunitarios**
   * Subir un documento para la comunidad: **5 puntos**
   * Descargar documentos comunitarios: **3 puntos**
3. **Interfaz Web Responsiva**
   * Adaptación de la plataforma para dispositivos móviles: **8 puntos**

**Prioridad Baja**

1. **Notificaciones y Comunicaciones**
   * Notificaciones sobre eventos y documentos: **5 puntos**
2. **Pruebas e Integración Continua**
   * Pruebas unitarias e integración del sistema: **8 puntos**
3. **Hosting y Escalabilidad**
   * Despliegue en un entorno escalable: **13 puntos**

#### **Prioridad Posterior al Lanzamiento**

1. **Capacitación de Usuarios**
   * Creación de tutoriales y manuales: **3 puntos**
2. **Marcha Blanca**
   * Pruebas de marcha blanca con usuarios reales: **5 puntos**

**Resumen Total de Puntos**

| **Prioridad** | **Total de Puntos** |
| --- | --- |
| Prioridad Alta | 37 puntos |
| Prioridad Media | 29 puntos |
| Prioridad Baja | 26 puntos |
| Posterior al Lanzamiento | 8 puntos |
| **Total del Proyecto** | **100 puntos** |

## Técnica de estimación

La técnica de estimación utilizada para asignar puntos de historia a cada tarea fue la secuencia de Fibonacci en conjunto con la Estimación Relativa. Este método es común en metodologías ágiles y sirve para evaluar el esfuerzo y la complejidad de cada historia en relación con las demás, en lugar de medirlas en tiempo absoluto.

### **Explicación de la Técnica Utilizada**

1. **Secuencia de Fibonacci**:
   * Se asignan puntos de historia de acuerdo con la secuencia de Fibonacci (1, 2, 3, 5, 8, 13, 21, etc.).
   * Los números mayores reflejan un aumento en la incertidumbre y complejidad de las tareas. A medida que las historias aumentan en dificultad, los puntos asignados también crecen de forma no lineal.
2. **Estimación Relativa**:
   * Se estima cada historia no en términos de horas o días, sino en comparación con otras historias.
   * Por ejemplo, si el registro de usuario se considera aproximadamente el doble de complejo que ver eventos en el calendario, recibe un valor dos veces mayor en la escala de Fibonacci.
3. **Ventajas de esta Técnica**:
   * Claridad: La secuencia de Fibonacci simplifica el proceso de asignar estimaciones, sobre todo en proyectos de gran escala.
   * Adaptación al Cambio: La técnica permite ajustarse a la incertidumbre y hacer cambios en las estimaciones durante la planificación si es necesario.
   * Comparación y Priorización: Ayuda a visualizar cuáles tareas son más complejas y requieren mayor esfuerzo, facilitando la priorización en el backlog.

La combinación de la secuencia de Fibonacci y la estimación relativa permite un enfoque ágil y flexible, ideal para proyectos que se desarrollan bajo metodologías como Kanban o Scrum.

## Sprint Planning

| **Sprint** | **Historia de Usuario** | **Puntos de Historia** |
| --- | --- | --- |
| Sprint 1 | Registro de usuario mayor de 18 años | 8 |
| Sprint 1 | Inicio de sesión en el sistema | 5 |
| Sprint 1 | Recuperación de contraseña | 5 |
| Sprint 2 | Estructuración de la base de datos para usuarios | 8 |
| Sprint 2 | Autorización de roles y permisos | 8 |
| Sprint 2 | Inscripción en un evento | 3 |
| Sprint 3 | Crear un evento comunitario | 8 |
| Sprint 3 | Ver eventos en el calendario | 5 |
| Sprint 3 | Subir un documento | 5 |
| Sprint 4 | Descargar documentos comunitarios | 3 |
| Sprint 4 | Notificaciones de eventos y documentos | 5 |
| Sprint 4 | Adaptación de la plataforma para dispositivos móviles | 8 |
| Sprint 5 | Pruebas unitarias e integración | 8 |
| Sprint 5 | Despliegue en un entorno escalable | 13 |

## Definición y estimación de tareas

| **Sprint** | **Total de Puntos** | **Estimación Total (h)** |
| --- | --- | --- |
| Sprint 1 | 18 puntos | 23 horas |
| Sprint 2 | 19 puntos | 18 horas |
| Sprint 3 | 18 puntos | 18 horas |
| Sprint 4 | 16 puntos | 17 horas |
| Sprint 5 | 21 puntos | 25 horas |
| Total del Proyecto | 92 puntos | 101 horas |

## Historias Usuario

| **ID de la Historia** | **Descripción de la Historia** | **Criterios de Aceptación** | **Prioridad** | **Estimación de Puntos de Historia** | **Notas Adicionales** |
| --- | --- | --- | --- | --- | --- |
| HU-01 | Como usuario, quiero registrarme en la plataforma para poder acceder a las funcionalidades. | Formulario de registro, validación de edad, confirmación de correo. | Alta | 8 | La edad mínima para registro es de 18 años. |
| HU-02 | Como usuario, quiero iniciar sesión en la plataforma para gestionar mi cuenta. | Formulario de inicio de sesión, autenticación y manejo de sesiones. | Alta | 5 | La sesión debe durar al menos 30 minutos sin actividad. |
| HU-03 | Como usuario, quiero poder recuperar mi contraseña en caso de olvidarla. | Formulario de recuperación, envío de enlace, restablecimiento de contraseña. | Alta | 5 | Enlace de restablecimiento debe expirar en 24 horas. |
| HU-04 | Como administrador, quiero una base de datos que guarde la información de usuarios y roles. | Modelo de base de datos funcional, conexión y pruebas de integridad. | Media | 8 | Debe permitir integrarse con los sistemas existentes. |
| HU-05 | Como administrador, quiero asignar roles y permisos a los usuarios según su nivel de acceso. | Roles definidos, permisos asignados, panel de gestión de roles. | Media | 8 | Roles como Administrador, Usuario y Moderador deben estar definidos. |
| HU-06 | Como usuario, quiero inscribirme en eventos para participar en actividades comunitarias. | Inscripción exitosa, confirmación de inscripción y cancelación. | Media | 3 | Es importante poder ver detalles del evento tras inscripción. |
| HU-07 | Como administrador, quiero crear eventos para que los usuarios puedan verlos en el calendario. | Formulario de creación de eventos, validación y calendario. | Alta | 8 | Los eventos deben estar visibles en el calendario público. |
| HU-08 | Como usuario, quiero ver los eventos en un calendario para organizarme mejor. | Visualización de eventos en el calendario, detalles al hacer clic. | Media | 5 | El calendario debe mostrar eventos según la fecha de hoy. |
| HU-09 | Como usuario, quiero subir documentos a la plataforma para compartirlos con la comunidad. | Formulario de subida de documentos, validación de tipo y clasificación. | Media | 5 | Validar tipo de archivo antes de subir documentos. |
| HU-10 | Como usuario, quiero descargar documentos comunitarios para acceder a información relevante. | Lista de documentos accesible y permisos de descarga definidos. | Media | 3 | Permitir la descarga solo a usuarios registrados. |
| HU-11 | Como usuario, quiero recibir notificaciones sobre nuevos eventos y documentos importantes. | Lógica de notificaciones, correos automáticos y panel de notificaciones. | Media | 5 | Notificaciones también deben enviarse por email. |
| HU-12 | Como usuario, quiero acceder a la plataforma en mi dispositivo móvil para consultar información en cualquier lugar. | Interfaz adaptada para móviles, elementos optimizados y navegación fácil. | Baja | 8 | Debe soportar pantallas de 5-12 pulgadas adecuadamente. |
| HU-13 | Como administrador, quiero realizar pruebas para garantizar que el sistema funcione correctamente. | Pruebas unitarias completadas, pruebas de integración y carga. | Media | 8 | Incluir pruebas automatizadas para cada módulo clave. |
| HU-14 | Como administrador, quiero desplegar la plataforma en un entorno escalable para soportar más usuarios. | Servidor configurado, balanceo de carga y sistema de respaldo. | Media | 13 | Soportar al menos 1000 usuarios concurrentes sin problemas. |

\*PMOInformatica Plantilla de Product Backlog(completo)

\*PMOInformatica Plantilla de Sprint Backlog(completo)