**Plan de Pruebas**

**“Red Vecinal”**

***Integrantes: Gabriel Cárcamo***

***Victor Navarro***

***Christopher Valenzuela***

***Fecha: 12/10/2024***

**Tabla de contenido**

Histórico de Revisiones

| Versión | Fecha | Descripción/cambio | autor |
| --- | --- | --- | --- |
| 1.0 | 12/10/2024 | Inicio de la realización del plan de pruebas, detallando su contenido y definiendo cada sección. | Christopher Valenzuela |
| 2.0 | 30/10/2024 | Finalización del plan de pruebas y ultimando detalles de algunas secciones. | Christopher Valenzuela |
|  |  |  |  |

Información del Proyecto

| Organización |  |
| --- | --- |
| Sección | 002D |
| Proyecto | Red Vecinal |
| Fecha de Inicio | 12/08/2024 |
| Fecha de Término | 09/12/2024 |
| Docente | Francisco Diaz |

Integrantes

| Rut | Nombre | Correo |
| --- | --- | --- |
| 21076959-5 | Gabriel Cárcamo | gab.carcamo@duocuc.cl |
| 26861080-4 | Victor Navarro | vic.navarro@duocuc.cl |
| 20379808-3 | Christopher Valenzuela | ch.valenzuela@duocuc.cl |

| Propósito del plan de pruebas  *Propósito, objetivo, visión que se espera de este plan de pruebas.* |
| --- |
| El propósito del plan de pruebas es asegurar la calidad, estabilidad y correcto funcionamiento del sistema web, verificando que todas las funcionalidades cumplan los requisitos, identificando errores y garantizando una experiencia segura y fácil de usar para los usuarios finales. |

| Alcance de las pruebas  *Definición de requisitos de S.W., módulos de Software a probar, Requisitos ambiente de pruebas y Documentación Referenciada, etc.* |
| --- |
| El alcance de las pruebas incluye la evaluación de todos los componentes del sistema, tanto del lado del usuario final como del administrativo. Se abarcan las siguientes áreas:   * Pruebas Funcionales:   Validar que las funcionalidades principales cumplan con los requisitos definidos, tales como:   * Registro de usuarios (vecinos y directiva). * Gestión de solicitudes de certificados de residencia. * Postulación y gestión de proyectos vecinales. * Reservas de espacios comunitarios mediante el calendario. * Envío y recepción de notificaciones y avisos. * Publicación y visualización de noticias. * Pruebas de Usabilidad: * Evaluar la facilidad de uso del sistema, asegurando que la interfaz sea intuitiva y accesible para usuarios sin experiencia técnica. * Identificar posibles mejoras en la experiencia del usuario (UX/UI). * Pruebas de Rendimiento: * Medir los tiempos de respuesta del sistema bajo condiciones normales y de carga. * Verificar que el sistema soporte al menos 100 usuarios concurrentes sin pérdida de rendimiento. * Pruebas de Seguridad: * Evaluar la protección de datos personales de los usuarios mediante autenticación y cifrado. * Asegurar que no existan vulnerabilidades comunes como inyecciones SQL o accesos no autorizados. * Pruebas de Compatibilidad: * Verificar que el sistema funcione correctamente en los principales navegadores (Chrome, Firefox, Safari, Edge). * Evaluar el desempeño en dispositivos con diferentes resoluciones y capacidades. * Pruebas de Integración: * Validar la correcta interacción entre el frontend y el backend. * Verificar la integración correcta con la base de datos relacional gestionada por Django, asegurando que las operaciones CRUD (crear, leer, actualizar y eliminar) funcionen sin errores. * Pruebas de Mantenimiento: * Asegurar que el sistema sea fácil de actualizar y mantener, garantizando que las modificaciones no afecten funcionalidades existentes. |

| Definición de roles y responsabilidades  *Roles y responsabilidades de todos los participantes en el* ***proceso de pruebas*** *de SW*. | | |
| --- | --- | --- |
| Rol | Responsabilidades | Relevancia |
| Jefe QA | Se encargará de dirigir y supervisar las pruebas | Alta |
| Analista QA | Se encargará de realizar las pruebas y mantener informado al jefe de QA | Alta |

| Tipos de pruebas a realizar  *Definir el tipo de pruebas que se debe desarrollar para este proyecto, actividades y responsables.* |
| --- |
| **Pruebas Unitarias**  **Objetivo:** Verificar que los componentes individuales del sistema funcionen correctamente de manera aislada.  **Actividades:**   * Testear funciones del backend (Django) para operaciones CRUD. * Verificar la correcta validación de datos en formularios. * Comprobar la ejecución de notificaciones y respuestas desde el servidor.   **Responsables:**   * Victor Navarro: Backend (lógica del servidor). * Christopher Valenzuela: Validaciones del frontend.   **Pruebas de Integración**  **Objetivo:** Validar la interacción entre los diferentes módulos del sistema (frontend, backend y base de datos).  **Actividades:**   * Probar la integración entre la interfaz gráfica (frontend) y la lógica del backend. * Verificar que las operaciones de la base de datos se ejecuten correctamente desde las vistas. * Asegurar la comunicación fluida entre módulos como el registro de usuarios, gestión de solicitudes y notificaciones.   **Responsables:**   * Victor Navarro y Christopher Valenzuela: Integración frontend-backend. * Gabriel Cárcamo: Integración del diseño de la interfaz.   **Pruebas Funcionales**  **Objetivo:** Confirmar que el sistema cumple con los requisitos funcionales especificados.  **Actividades:**   * Validar el flujo completo de inscripción de vecinos. * Comprobar la generación y envío de certificados. * Testear la funcionalidad de reservas en el calendario.   **Responsables:**   * Todo el equipo: Validación de cada módulo funcional.   **Pruebas de Usabilidad**  **Objetivo:** Evaluar la experiencia de usuario (UX) y la facilidad de uso del sistema.  **Actividades:**   * Realizar pruebas con usuarios finales para identificar mejoras en la interfaz. * Verificar la claridad de los menús, botones y mensajes del sistema.   **Responsables:**   * Gabriel Cárcamo: Evaluación de la experiencia del usuario. * Christopher Valenzuela: Ajustes según los comentarios de prueba.   **Pruebas de Rendimiento**  **Objetivo:** Asegurar que el sistema responda adecuadamente bajo diferentes condiciones de carga.  **Actividades:**   * Medir tiempos de respuesta en operaciones comunes (inscripción, certificados, etc.). * Simular varios usuarios concurrentes para verificar la estabilidad.   **Responsables:**   * Victor Navarro: Backend y base de datos. * Christopher Valenzuela: Interacción del frontend bajo carga.   **Pruebas de Seguridad**  **Objetivo:** Garantizar que los datos del sistema estén protegidos contra accesos no autorizados.  **Actividades:**   * Comprobar la correcta autenticación y permisos de usuarios. * Validar la protección contra inyecciones SQL y otras vulnerabilidades.   **Responsables:**   * Victor Navarro: Backend y gestión de datos. * Gabriel Cárcamo: Pruebas de seguridad en formularios.   **Pruebas de Compatibilidad**  **Objetivo:** Asegurar que el sistema funcione correctamente en diferentes navegadores y dispositivos.  **Actividades:**   * Probar la visualización en navegadores como Chrome, Firefox, Edge y Safari. * Verificar la adaptabilidad de la interfaz en distintas resoluciones.   **Responsables:**   * Christopher Valenzuela: Compatibilidad en frontend. * Gabriel Cárcamo: Verificación del diseño responsivo. |

| Estrategia y técnicas de pruebas a aplicar  *Definir las estrategias y técnicas de pruebas que se debe desarrollar para este proyecto, actividades y responsables.* |
| --- |
| | **Estrategia/Técnica** | **Actividades** | **Responsables** | | --- | --- | --- | | Estrategia Incremental | Probar módulos aislados y luego integrarlos. | Victor Navarro, Christopher Valenzuela | | Estrategia Basada en Riesgos | Priorizar pruebas en funcionalidades críticas como seguridad y datos. | Victor Navarro, Gabriel Cárcamo | | Pruebas de Caja Negra | Validar entradas/salidas sin examinar el código. | Todo el equipo, Gabriel Cárcamo | | Pruebas Exploratorias | Probar escenarios no documentados para descubrir fallos inesperados. | Todo el equipo | | Pruebas de Regresión | Repetir pruebas tras correcciones o nuevas integraciones. | Christopher Valenzuela, Victor Navarro | |

| Definición del proceso de testing  *Listar y describir todas las actividades a desarrollar en el proceso general de testing, responsables, artefactos, etc.* |
| --- |
| **Planificación de Pruebas**  **Objetivo:** Definir el alcance, estrategias y cronograma para las pruebas.  **Actividades**:   * Identificar las funcionalidades críticas: inscripción, certificados, reservas, notificaciones. * Establecer un cronograma para pruebas unitarias, de integración y funcionales. * Asignar responsables para cada tipo de prueba.   **Responsables:**   * Victor Navarro: Backend. * Christopher Valenzuela: Frontend. * Gabriel Cárcamo: UX/UI.   **Diseño de Casos de Prueba**  **Objetivo:** Crear escenarios de prueba basados en los requisitos funcionales y no funcionales.  **Actividades:**   * Escribir casos de prueba para: * Flujo de registro de vecinos. * Generación y envío de certificados. * Gestión de reservas. * Notificaciones y avisos. * Definir entradas, acciones esperadas y resultados esperados. * Herramientas: Documentos de Excel o herramientas como TestRail.   **Responsables:**   * Todo el equipo.   **Preparación del Entorno de Pruebas**  **Objetivo:** Configurar el entorno necesario para realizar las pruebas.  **Actividades:**   * Configurar el servidor de desarrollo con Django y la base de datos integrada. * Crear cuentas de prueba (administradores y vecinos). * Cargar datos iniciales para simular un entorno real (ejemplo: eventos, proyectos vecinales).   **Responsables:**   * Victor Navarro: Backend y base de datos. * Christopher Valenzuela: Frontend.   **Ejecución de Pruebas**  **Objetivo:** Probar cada funcionalidad y registrar los resultados.  **Actividades:**   * Realizar pruebas funcionales (inscripción, reservas, generación de certificados). * Validar la interfaz con pruebas de usabilidad. * Ejecutar pruebas de rendimiento con múltiples usuarios simultáneos. * Realizar pruebas de compatibilidad en diferentes navegadores. * Herramientas: Navegadores web, herramientas de carga como JMeter.   **Responsables:**   * Christopher Valenzuela: Frontend y usabilidad. * Victor Navarro: Backend y rendimiento. * Gabriel Cárcamo: Validación de diseño y experiencia.   **Gestión de Defectos**  **Objetivo:** Registrar, priorizar y corregir errores encontrados.  **Actividades:**   * Registrar los defectos en un sistema de seguimiento (Trello, Jira o similar). * Clasificar los errores según su criticidad (crítico, medio, bajo). * Verificar las correcciones con pruebas de regresión.   **Responsables:**   * Victor Navarro: Backend. * Christopher Valenzuela: Frontend.   **Evaluación y Cierre**  **Objetivo:** Analizar los resultados y determinar si el software está listo para producción.  **Actividades:**   * Generar un informe con métricas de pruebas (número de casos probados, errores detectados, etc.). * Verificar el cumplimiento de los criterios de aceptación: * Tiempo de respuesta menor a 3 segundos. * Manejo de al menos 100 usuarios simultáneos. * Funcionalidades operativas y sin errores críticos. * Realizar una reunión de revisión final.   **Responsables:**   * Todo el equipo.   **Criterios de éxito del proceso de testing:**   * El sistema pasa el 95% de las pruebas funcionales y de integración. * Se corrigen todos los errores críticos y el 90% de los errores de severidad media. * El sistema cumple con los tiempos de respuesta definidos. * El sistema es estable bajo la carga de usuarios esperada. |

| Definición de ciclos de prueba a ejecutar  *Listar y describir cantidad de ciclos de prueba a realizar en este proyecto, las tareas y actividades para cada ciclo de prueba, responsables, artefactos, etc.* |
| --- |
| | Ciclo de prueba | Objetivo | Actividades | Responsables | Duración estimada | | --- | --- | --- | --- | --- | | Ciclo 1: Pruebas Unitarias | Verificar el correcto funcionamiento de componentes individuales (funciones o módulos). | - Probar funciones CRUD del backend.  - Validar componentes de la interfaz (formularios, botones). | Victor Navarro (Backend)  Christopher Valenzuela (Frontend) | 1 semana | | Ciclo 2: Pruebas de Integración | Validar la interacción entre los módulos del sistema (frontend, backend y base de datos). | - Probar flujos completos (registro, certificados, reservas).  - Verificar el intercambio de datos entre frontend y backend. | Victor Navarro  Christopher Valenzuela | 1 semana | | Ciclo 3: Pruebas Funcionales | Garantizar que las funcionalidades cumplen con los requisitos definidos. | - Probar el registro de usuarios, generación de certificados, gestión de proyectos y reservas.  - Validar el envío y recepción de notificaciones. | Todo el equipo | 1 semana | | Ciclo 4: Pruebas de Usabilidad | Evaluar la experiencia del usuario (UX) y la facilidad de uso del sistema. | - Realizar pruebas con usuarios no técnicos.  - Analizar la navegación, diseño e intuitividad de la interfaz. | Gabriel Cárcamo (Diseño UX/UI) | 1 semana | | Ciclo 5: Pruebas de Rendimiento | Verificar el tiempo de respuesta y el comportamiento del sistema bajo carga. | - Medir tiempos de respuesta en operaciones comunes.  - Simular usuarios concurrentes (hasta 100). | Victor Navarro (Backend)  Christopher Valenzuela (Frontend) | 3 días | | Ciclo 6: Pruebas de Seguridad | Garantizar la protección de datos y prevenir vulnerabilidades comunes. | - Comprobar autenticación y permisos de usuarios.  - Validar protección contra inyecciones SQL y accesos no autorizados. | Victor Navarro (Backend)  Gabriel Cárcamo (Validación de formularios) | 3 días | | Ciclo 7: Pruebas de Compatibilidad | Asegurar el funcionamiento en diferentes navegadores y resoluciones de pantalla. | - Probar el sistema en Chrome, Firefox, Edge y Safari.  - Validar en diferentes dispositivos (PC, tablet, smartphone). | Christopher Valenzuela (Frontend)  Gabriel Cárcamo (Diseño) | 1 semana | |

| Calendarización de las actividades de pruebas  *Listado de actividades, tareas, duración, fechas, responsables, etc.* |
| --- |
|  |
| Adjuntar carta Gantt |

| Resumen de riesgos  *Listado de riesgos relacionados al desarrollo de S.W. Indicar riesgo, etapa o fase en que se presenta, la probabilidad de que ocurra, magnitud o impacto de este riesgo por etapa en el proceso.*  *Probabilidad: Alta, media, baja. (matriz de probabilidades)*  *Impacto: Alto, Significativo, Moderado, Inferior y Baja.* |
| --- |
| | **Riesgo** | **Fase** | **Probabilidad** | **Impacto** | **Acción al riesgo** | | --- | --- | --- | --- | --- | | Cambios en los requerimientos | Desarrollo | Alta | Alto | Mantener documentación actualizada y realizar reuniones semanales de seguimiento | | Problemas técnicos con la integración de servicios | implementación | Media | Media | Realizar pruebas de integración tempranas y mantener backups | | Retrasos en el cronograma | Todo el proyecto | Media | Media | Implementar metodología ágil con sprints bien definidos y buffer de tiempo | | Problemas de compatibilidad con navegadores | Testing | Baja | Baja | Realizar pruebas en múltiples navegadores y versiones | | Falta de experiencia en tecnologías específicas | Desarrollo | Media | Media | Capacitación continua y mentorías entre el equipo | |

| Clasificación de los defectos  *Definir la clasificación de los defectos según su nivel de severidad* | |
| --- | --- |
| Nivel de Severidad | Descripción |
| Crítico | Defectos que provocan el fallo completo del sistema o la incapacidad de ejecutar funcionalidades esenciales. Por ejemplo, errores en el inicio de sesión o en la conexión a la base de datos. |
| Alto | Defectos que afectan funcionalidades clave del sistema, pero que tienen soluciones alternativas temporales. Por ejemplo, fallos en el envío de formularios o en la carga de datos importantes. |
| Media | Defectos que afectan funcionalidades secundarias o de menor prioridad, como problemas en elementos visuales o errores en validaciones no críticas. |
| Bajo | Defectos que no afectan directamente la funcionalidad del sistema, como problemas menores en la interfaz de usuario (estilo, alineación, o textos). |
| Estético | Defectos estéticos o de formato que no impactan la funcionalidad, como errores tipográficos o inconsistencias en el diseño visual. |

| Definición de artefactos  *Listar y describir los artefactos que serán administrados y entregados durante este proceso de prueba.* | |
| --- | --- |
| Artefacto | Descripción |
| Plan de pruebas | Documento que define el alcance , los objetivos, las estrategias y el cronograma de las pruebas, incluye los responsables y los recursos necesarios. |
| Casos de prueba | Escenarios detallados que describen las condiciones , datos de entrada y resultados esperados para validar cada funcionalidad del sistema. |
| Registro de defectos | Lista de los errores encontrados durante el proceso de pruebas, incluyendo su descripción severidad,estado y asignación a responsables. |
| Matriz de trazabilidad | Documento que relaciona los casos de prueba con los requisitos para garantizar que todos sean validados. |
| Resultados de pruebas | Reporte que describe los resultados obtenidos en las pruebas , incluyendo casos exitosos, fallidos y detalles de los defectos encontrados. |
| Reporte final de pruebas | Documento resumen que detalla el estado del sistema tras el proceso de pruebas ,indicando si cumple con los requisitos y esta listo para producción. |
| Scripts de pruebas | Conjunto de scripts automatizados utilizados para realizar pruebas funcionales , de regresión o de rendimiento , según sea necesario. |
| Ambiente de pruebas | Configuración documentada del entorno de pruebas, incluyendo versiones de software , herramientas utilizadas y datos de prueba. |

| Condiciones de aceptación para cierre del proceso de pruebas  *Condiciones que se deben cumplir para dar término al proceso de pruebas y margen de tolerancia de aceptación de defectos.* |
| --- |
| Las condiciones de aceptación de este proyecto será que las pruebas sean realizadas con éxito reduciendo al mínimo los riesgos y que todas las funcionalidades solicitadas estén en correcto funcionamiento.. |