**Plan de Pruebas**

**Portafolio de Título**

**“Encomienda.me”**

***Fecha: [13/12/2024]***

**Tabla de contenido**

* Información general del Proyecto
* Propósito del plan de prueba
* Alcance de pruebas
* Definición de roles y responsabilidades
* Tipos de pruebas
* Estrategia y técnicas de pruebas
* Definición de proceso de testing
* Definición de ciclos de prueba
* Calendarización de las actividades de pruebas (Carta Gantt)
* Resumen de riesgos
* Clasificación de defectos
* Definición de artefactos
* Condiciones de aceptación

Histórico de Revisiones

|  |  |  |  |
| --- | --- | --- | --- |
| Versión | Fecha | Descripción/cambio | autor |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Información del Proyecto

|  |  |
| --- | --- |
| Organización | Duoc UC. Escuela de Informática y Telecomunicaciones |
| Sección | 004V |
| Proyecto (Nombre) | Encomienda.me |
| Fecha de Inicio | 13/12/2024 |
| Fecha de Término | 15/12/2024 |
| Caso N° |  |
| Patrocinador principal | Duoc UC |
| Docente | Fabián Saldaño |

Integrantes

| Rut | Nombre | Correo |
| --- | --- | --- |
| **17.921.942-5** | **Ignacio Sánchez** | **ign.sanchez@duocuc.cl** |
| **19.525.066-9** | **Agustín Sánchez** | **Agu.sanchez@duocuc.cl** |
| **20.635.907-2** | **Kevin Trujillo** | **Kevi.trujillo@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 establecer una estrategia para poder probar el producto antes de su implementación. Mediante este plan se busca lograr los siguientes objetivos:  1.-Garantizar la calidad del producto.  2.-Validar los requisitos.  3.-Minimizar los riesgos.  4.- Optimizar el rendimiento.  5.-Asegurar la usabilidad. |

|  |
| --- |
| Alcance de las pruebas  *Definición de requisitos de S.W., módulos de Software a probar, Requisitos ambiente de pruebas y Documentación Referenciada, etc.* |
| Pruebas de registro de nuevos usuarios: Verificar que el proceso de registro sea fluido y sin errores, que los campos obligatorios se validan correctamente y que se generen mensajes de error adecuados en caso de datos incorrectos o faltantes.  Pruebas de carga masiva de envíos: Asegurarse de que la carga masiva de envíos funcione correctamente, validando la correcta interpretación y procesamiento de los datos cargados, así como la generación de notificaciones o mensajes de error en caso de problemas.  Pruebas de administración de clientes: Verificar que se puedan agregar, modificar y eliminar clientes de manera adecuada, validando que los datos se actualicen correctamente en la base de datos y que las acciones de administración se realicen de acuerdo con las reglas de negocio establecidas.  Pruebas de gestión de datos comerciales y tarifas: Asegurarse de que los datos comerciales y las tarifas se almacenen y recuperen correctamente de la base de datos, y que los cálculos de tarifas se realicen según lo esperado.  Pruebas de contratación de planes de servicio: Verificar que los usuarios puedan contratar los planes de servicio deseados y que se realicen los cambios apropiados en la base de datos, incluyendo actualizaciones en los precios y características del servicio.  Pruebas de generación de informes y análisis de datos: Validar que los informes y análisis generados sean precisos, completos y se ajusten a los requisitos especificados. Esto puede incluir la verificación de los datos mostrados, los cálculos realizados y la capacidad de filtrar y visualizar la información de manera adecuada.  Pruebas de interfaz de usuario y Dashboard: Verificar la usabilidad de la interfaz de usuario, asegurándose de que sea intuitiva, fácil de usar y responda de manera adecuada a las acciones del usuario. Esto incluye pruebas de navegación, diseño de pantalla, interacciones y respuesta a eventos. |

|  |  |  |
| --- | --- | --- |
| Definición de roles y responsabilidades  *Roles y responsabilidades de todos los participantes en el* ***proceso de pruebas*** *de SW*. | | |
| Rol | Responsabilidades | Relevancia |
| Analista de Pruebas | Analizar los requisitos y especificaciones del sistema.  Diseñar casos de prueba basados en los requisitos.  Identificar los escenarios y condiciones de pruebas. | El analista de pruebas es responsable de diseñar los casos de prueba adecuados para validar los requisitos del sistema y garantizar una cobertura exhaustiva de las funcionalidades |
| Desarrollador | Corregir los defectos identificados durante las pruebas.  Colaborar con el equipo de pruebas para entender y resolver los problemas.  Realizar pruebas unitarias para garantizar la calidad del código. | Son responsables de solucionar los errores encontrados durante las pruebas y garantizar la calidad del software |
| Tester | Ejecutar pruebas funcionales para validar la usabilidad y la experiencia del usuario.  Proporcionar retroalimentación sobre la calidad y el rendimiento del software.  Identificar problemas y proponer mejoras desde la perspectiva del usuario final. | Aportar una perspectiva crítica desde la experiencia del usuario y pueden identificar problemas o mejoras que pueden pasar desapercibidos por otros roles |

|  |
| --- |
| Tipos de pruebas a realizar  *Definir el tipo de pruebas que se debe desarrollar para este proyecto, actividades y responsables.* |
| Las principales pruebas a realizar en este proyecto serán pruebas de caja negra. Este enfoque de prueba se basa en verificar el comportamiento de software desde una perspectiva externa, sin considerar la estructura interna. Se centrará en probar las funcionalidades y características del sistema basándose en los requisitos y especificaciones establecidos. |

|  |
| --- |
| Estrategia y técnicas de pruebas a aplicar  *Definir las estrategias y técnicas de pruebas que se debe desarrollar para este proyecto, actividades y responsables.* |
| Estrategias de Pruebas:  Pruebas funcionales: Validar que todas las funcionalidades del sistema cumplan con los requisitos establecidos.  Pruebas de usabilidad: evaluar la facilidad de uso de la experiencia de usuario.  Pruebas de rendimiento: Evaluar el rendimiento y capacidad del sistema bajo diferentes cargas y condiciones.  Pruebas de seguridad; Identificar vulnerabilidades y asegurar la protección de los datos sensibles.  Técnicas de pruebas:  Casos de pruebas basados en especificaciones: Diseñar los casos de pruebas a partir de los requisitos y especificaciones del sistema.  Partición de equivalencia: Agrupar conjuntos de datos en clases de equivalencia y probar un caso representativo de cada clase.  Pruebas de límites: Verificar el comportamiento del sistema en los límites de los rangos establecidos.  Pruebas de flujos de trabajo: Validar los flujos de trabajo del sistema, asegurando que las acciones se realicen en el orden correcto.  Pruebas de regresión: Volver a probar las funcionalidades existentes para asegurar que las modificaciones no hayan introducido nuevos errores.  Actividades y responsables:  -Diseño de casos de pruebas: analizar requisitos, identificar escenarios de prueba, diseñar casos de prueba y documentarlos.  Responsable: Analista de pruebas.  -Ejecución de pruebas: Ejecutar los casos de pruebas, registrar resultados y reportar cualquier defecto encontrado.  Responsable: Tester  Corrección de defectos: Corregir los defectos identificados durante las pruebas, colaborar con el equipo de pruebas para resolver el problema.  Responsable: Desarrollador. |

|  |
| --- |
| 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:  Definición de los objetivos, alcance, estrategia y recursos necesarios para las pruebas.  Identificación de los requerimientos y especificaciones de pruebas.  Creación del plan de pruebas, que incluye la definición de casos de prueba, técnicas y herramientas a utilizar.  Diseño de pruebas:  Identificación y elaboración de casos de prueba basados en los requisitos y especificaciones.  Selección de técnicas de diseño de casos de prueba, como partición de equivalencia o pruebas de límites.  Creación de los datos de prueba necesarios.  Preparación del entorno de pruebas:  Configuración del ambiente de pruebas, que puede incluir hardware, software, bases de datos y otros componentes necesarios.  Instalación y configuración del software de prueba y herramientas relacionadas.  Ejecución de pruebas:  Ejecución de los casos de prueba diseñados.  Registro de los resultados obtenidos y de cualquier defecto o error encontrado.  Seguimiento y gestión de los defectos, incluyendo su clasificación, priorización y asignación para su corrección.  Análisis y reporte de resultados:  Evaluación de los resultados de las pruebas y comparación con los criterios de aceptación.  Generación de informes de pruebas, que documentan los resultados obtenidos, los defectos encontrados y cualquier otra observación relevante.  Retesting y confirmación:  Realización de pruebas adicionales para verificar la corrección de los defectos corregidos.  Confirmación de que los problemas han sido solucionados y que el software cumple con los criterios de aceptación.  Cierre de pruebas:  Evaluación del proceso de pruebas, identificando lecciones aprendidas y áreas de mejora.  Documentación final del proceso de pruebas y de los resultados obtenidos. |

|  |
| --- |
| 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 Pruebas Unitarias:  Tareas y actividades:  Diseñar y ejecutar pruebas unitarias para verificar el correcto funcionamiento de los componentes individuales del software.  Identificar y corregir defectos a nivel de unidad.  Verificar que cada unidad cumple con su especificación y requisitos.  Responsables: Desarrolladores y analistas de pruebas.  Artefactos: Casos de prueba unitarios, informe de resultados de pruebas unitarias.  Ciclo de Pruebas de Integración:  Tareas y actividades:  Diseñar y ejecutar pruebas de integración para validar la interacción y comunicación entre los diferentes módulos o componentes del sistema.  Identificar y corregir problemas de interoperabilidad y comunicación entre módulos.  Verificar que los datos se transfieran correctamente entre los componentes integrados.  Responsables: Analistas de Pruebas, Desarrolladores.  Artefactos: Casos de prueba de integración, informe de resultados de pruebas de integración.  Ciclo de Pruebas de Sistema:  Tareas y actividades:  Diseñar y ejecutar pruebas de sistema para validar el comportamiento global del software en el contexto del entorno de producción simulado.  Verificar que todas las funcionalidades del sistema funcionen correctamente en conjunto.  Identificar y corregir problemas de configuración o instalación del sistema.  Responsables: Analistas de Pruebas, Desarrolladores, Test.  Artefactos: Casos de prueba de sistema, informe de resultados de pruebas de sistema.  Ciclo de Pruebas de Aceptación:  Tareas y actividades:  Diseñar y ejecutar pruebas de aceptación para validar que el software cumple con los requisitos y expectativas del cliente.  Verificar que el software se ajusta a los escenarios y casos de uso del cliente.  Obtener el feedback del cliente y realizar ajustes o correcciones según sea necesario.  Responsables: Analistas de Pruebas, Ingenieros de Pruebas, Desarrolladores, Usuarios de Pruebas y Clientes.  Artefactos: Casos de prueba de aceptación, informe de resultados de pruebas de aceptación. |

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

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Resumen de riesgos  *Listado de riesgos relacionado al proceso de pruebas de S.W. Indicar riesgo, magnitud o impacto de este riesgo por etapa en el proceso. Magnitud: Alto, Significativo, Moderado, Inferior y Baja.* | | | | | | |
|  | **Fase del proceso de pruebas** | | | | | **Riesgo** |
| **Planificación** | **Análisis y diseño** | **Implementación y ejecución** | **Evaluación** | **Cierre** |
| **Magnitud** |  |  |  |  |  |
| **Alto** |  |  |  |  | Recursos insuficientes para pruebas |
| **Significativo** |  |  |  |  | Plazos ajustados que dificultan la planificación |
|  | **Alto** |  |  |  | Requisitos mal definidos o ambiguos |
|  | **Moderado** |  |  |  | Escenarios de prueba incompletos o mal definidos |
|  |  | **Alto** |  |  | Defectos críticos no identificados |
|  |  | **Significativo** |  |  | Problema con integración de componentes |
|  |  |  | **Moderado** |  | Criterios de aceptación mal definidos o inconsistente |
|  |  |  | **Significativo** |  | Defectos no corregidos en tiempo y forma |
|  |  |  |  | **Moderado** | Documentación de pruebas incompleta o incorrecta |
|  |  |  |  |  | **Baja** | Rechazo o falta de aprobación de los resultados |

|  |  |
| --- | --- |
| Clasificación de los defectos  *Definir la clasificación de los defectos según su nivel de severidad* | |
| Nivel de Severidad | Descripción |
| Alto | Los defectos de nivel alto tienen un impacto significativo en la funcionalidad del software y pueden causar interrupciones importantes en el sistema. Pueden afectar la capacidad del software para realizar funciones clave o generar resultados incorrectos que afecten directamente al usuario. Estos defectos deben ser corregidos de manera prioritaria. |
| Significativo | Los defectos de nivel significativo tienen un impacto importante en la funcionalidad del software, pero no llegan a ser críticos. Pueden afectar la experiencia del usuario, la eficiencia del sistema o generar resultados inexactos en situaciones específicas. Estos defectos requieren atención y corrección pronta. |
| Moderado | Los defectos de nivel moderado tienen un impacto medio en la funcionalidad del software. Pueden generar molestias o inconvenientes menores para el usuario, pero no afectan gravemente su capacidad para utilizar el sistema o lograr sus objetivos. Estos defectos pueden ser corregidos en futuras versiones o actualizaciones del software. |
| Inferior | Los defectos de nivel inferior tienen un impacto mínimo en la funcionalidad del software y son de baja relevancia. Pueden ser errores estéticos, ortográficos o pequeñas desviaciones en la presentación de la información que no afectan significativamente la experiencia del usuario o la funcionalidad principal del sistema. Estos defectos pueden abordarse en etapas posteriores o considerarse como mejoras menores. |
| Baja | Los defectos de nivel bajo tienen un impacto mínimo en la funcionalidad y la experiencia del usuario. Son errores menores o sutilezas que no tienen un impacto significativo en la operación del software. Pueden incluir problemas de formato, etiquetas incorrectas o características poco utilizadas. Estos defectos pueden abordarse en futuras versiones o simplemente aceptarse como limitaciones menores. |

|  |  |
| --- | --- |
| Definición de artefactos  *Listar y describir los artefactos que serán administrados y entregados durante este proceso de prueba.* | |
| Artefacto | Descripción |
| Plan de pruebas | Es un documento que describe la estrategia general de pruebas, los objetivos, el alcance, los recursos necesarios y la programación de las actividades de pruebas. |
| Casos de pruebas | Son documentos que contienen una descripción detallada de los escenarios y situaciones a probar, así como los pasos y los datos necesarios para ejecutar cada prueba. |
| Informe de resultados de pruebas | Es un documento que presenta los resultados y los hallazgos obtenidos durante la ejecución de las pruebas. Incluye la lista de casos de prueba ejecutados, los resultados, las observaciones relevantes y las métricas de cobertura de pruebas. |
| Documentación de pruebas | Incluye documentos que describen la estrategia de pruebas, los criterios de aceptación, los procedimientos de configuración y los ambientes de pruebas utilizados. |

|  |
| --- |
| Condiciones de aceptación para cierre del proceso de pruebas  *Condiciones que se deben cumplir para dar termino al proceso de pruebas y margen de tolerancia de aceptación de defectos.* |
| Cobertura de pruebas: Se ha alcanzado una cobertura de pruebas adecuada, lo que significa que se han diseñado y ejecutado pruebas suficientes para cubrir los requisitos funcionales y no funcionales del software.  Criterios de aceptación: Los criterios de aceptación definidos previamente se han cumplido. Esto implica que el software ha pasado todas las pruebas requeridas y cumple con los estándares y requisitos establecidos.  Tolerancia de Defectos: Se ha establecido un margen de tolerancia para la aceptación de defectos. Esto puede variar según el proyecto y las expectativas del cliente, pero generalmente se establece un límite razonable para la cantidad y la severidad de los defectos aceptables en el software probado.  Estabilidad del Software: El software se encuentra en un estado estable y funcional. Se ha realizado un seguimiento y una evaluación exhaustiva de su rendimiento, estabilidad y comportamiento en diferentes condiciones. |