Plan de Pruebas de Software

***[‘‘MiWalker’’]***

***Fecha: [29/10/2024]***

# Tabla de contenido

[**Tabla de contenido 2**](#_heading=h.gks12brj1x43)

[**Historial de Versiones 3**](#_heading=h.rrm3dgxvpvsc)

[**Información del Proyecto 3**](#_heading=h.30j0zll)

[**Aprobaciones 3**](#_heading=h.ti4vh8mx74mn)

[**Resumen Ejecutivo 4**](#_heading=h.s2jmdceoxcc)

[**Alcance de las Pruebas 5**](#_heading=h.2i9l6l9jsuwv)

[Elementos de Pruebas 5](#_heading=h.pwaktqnde5ep)

[Nuevas Funcionalidades a Probar 6](#_heading=h.e2h1lfs3acat)

[Pruebas de Regresión 6](#_heading=h.ijt35rxgpiy0)

[Funcionalidades a No Probar 7](#_heading=h.cvhqisn4o6my)

[Enfoque de Pruebas (Estrategia) 8](#_heading=h.l5kbzq8e661)

[Criterios de Aceptación o Rechazo 9](#_heading=h.3rdcrjn)

[Criterios de Suspensión 9](#_heading=h.g7x6r1m1su96)

[Criterios de Reanudación 10](#_heading=h.6iyhwobeyqjs)

[**Entregables 11**](#_heading=h.1hq6vydcrxla)

[**Recursos 11**](#_heading=h.m8o5v068txmm)

[Requerimientos de Entornos – Hardware 11](#_heading=h.44sinio)

[Requerimientos de Entornos – Software 13](#_heading=h.ah4v4rysvcyq)

[Herramientas de Pruebas Requeridas 14](#_heading=h.2bzrs6b3mx6n)

[Personal 15](#_heading=h.iw6uetf87w06)

[Entrenamiento 17](#_heading=h.8tan7xi0b76d)

[**Planificación y Organización 19**](#_heading=h.3fdtbla8zixf)

[Procedimientos para las Pruebas 19](#_heading=h.2xcytpi)

[Matriz de Responsabilidades 21](#_heading=h.k6vofqvw3dsu)

[Cronograma 22](#_heading=h.w6l7uwh5nv66)

[Premisas 22](#_heading=h.96guxn5b24lk)

[Dependencias y Riesgos 23](#_heading=h.qsh70q)

[**Referencias 25**](#_heading=h.ezbck7t71gup)

[**Glosario 27**](#_heading=h.1omtka6mj5sy)

# Historial de Versiones

| **Fecha** | **Versión** | **Autor** | **Organización** | **Descripción** |
| --- | --- | --- | --- | --- |
| 29/10/2024 | 1.0 | Diego Bulnes | QA | Creación Plan de Pruebas |
| 05/11/2024 | 1.1 | Diego Bulnes | QA | Profundización de Pruebas |
|  |  |  |  |  |
|  |  |  |  |  |

# Información del Proyecto

| Empresa / Organización | Empresa privada multinacional en santiago |
| --- | --- |
| Proyecto | MiWalker |
| Fecha de preparación | 29/10/2024 |
| Cliente | Departamento de operaciones/ Gestión de inventario |
| Patrocinador principal | Departamento de operaciones/ Gestión de inventario |
| Gerente / Líder de Proyecto | Rodrigo Durán |
| Gerente / Líder de Pruebas de Software | Rodrigo Durán / Diego Bulnes |

# Aprobaciones

| **Nombre y Apellido** | **Cargo** | **Departamento u Organización** | **Fecha** | **Firma** |
| --- | --- | --- | --- | --- |
| Diego Bulnes | QA | Calidad | 29/10/2024 |  |
| Rodrigo Durán | Gerente Proyecto | Operaciones | 05/11/2024 |  |
|  |  |  |  |  |
|  |  |  |  |  |

# Resumen Ejecutivo

El siguiente plan de pruebas tiene como objetivo garantizar la calidad, seguridad y funcionalidad de la aplicación web que se analizará, teniendo como meta ofrecer a los usuarios una experiencia segura y fluida en el canje, sorteo y subasta de productos.

Las pruebas se centrarán en la funcionalidad de diferentes aspectos de la aplicación web tales como el inicio de sesión, la elección de productos, el canje de productos y facturación, finalizar la obtención de productos y servicios además de configuraciones específicas de usuario.

Como estrategias de prueba se realizarán pruebas funcionales exhaustivas para verificar la funcionalidad de los métodos de canje subasta y sorteo, incluyendo agregar y eliminar productos y calcular el total de puntos para el canje, pruebas de rendimiento para evaluar la velocidad de carga de la página y la capacidad de manejar una carga de usuarios concurrentes, pruebas de seguridad para identificar y mitigar vulnerabilidades potenciales en la información de los usuarios, pruebas de usabilidad para evaluar la navegación y pruebas de compatibilidad para garantizar que la aplicación web sea accesible y funcional en diferentes dispositivos.

El equipo de pruebas estará compuesto por dos probadores de calidad, un ingeniero de pruebas de seguridad y un ingeniero de rendimiento. Se utilizarán pruebas mediante experiencia de usuario con diferentes propósitos evaluativos.

Las pruebas se llevarán a cabo en tres fases: pruebas de unidad, pruebas de integración y pruebas de sistema. Se espera que el proceso de pruebas dure dos semanas, seguido de una semana adicional para documentar los hallazgos y preparar el informe final.

Las pruebas se considerarán exitosas si se cumplen los siguientes criterios:

Se han identificado y solucionado todas las vulnerabilidades de seguridad críticas.

La página web carga rápidamente y puede manejar la carga de al menos 100 usuarios concurrentes sin degradación significativa del rendimiento.

Todas las funcionalidades planteadas en las pruebas se comportan según lo previsto.

La aplicación es compatible con los navegadores principales (Chrome, Firefox, Edge, Safari) y dispositivos (móviles, tabletas, computadoras de escritorio).

# Alcance de las Pruebas

A continuación, se detallarán los elementos de pruebas que se utilizaran para el análisis de este documento.

## Elementos de Pruebas

| Módulos a Probar | Descripción |
| --- | --- |
| Iniciar sesión | Requerimiento que permite iniciar sesión en el sistema |
| Registrar usuario | Requerimiento que permite al usuario crear una cuenta en el sistema |
| Crear sorteo | Requerimiento que permite al administrador crear un sorteo en el sistema |
| Participar en sorteo | Requerimiento que permite al empleado participar en un sorteo en el sistema |
| Crear subasta | Requerimiento que permite al administrador crear una subasta en el sistema |
| Cerrar subasta | Requerimiento que permite al administrador dar cierre al plazo de la subasta en el sistema |
| Crear nueva marca | Requerimiento que permite al administrador crear un producto disponible para el canje de los empleados |
| Agregar Producto | Requerimiento que permite al usuario agregar un producto a su lista de canje |
| Ingreso de stock | Requerimiento que permite al administrador aumentar o disminuir el número de existencias de la lista de productos |
| Canje de productos | Requerimiento que permite al empleado canjear uno o varios productos en el sistema |
| Generar PDF | Requerimiento que permite al administrador generar una factura del producto obtenido por un usuario en el sistema |

## 

## Nuevas Funcionalidades a Probar

Las funcionalidades que se van a probar incluyen, entre otras:

* Página de inicio
* Listado de productos
* Visualización de categorías y productos
* Proceso de canje
* Proceso de subasta
* Proceso de sorteo
* Registro e inicio de sesión de usuarios
* Envío de mensaje asociado ana actividad
* Generar factura

## Pruebas de Regresión

A continuación se presenta un listado de pruebas de regresión, donde se busca verificar diferentes aspectos de la página web.

Pruebas de Funcionalidad General: Verificar que las funciones principales de la página web, como la búsqueda de productos, la navegación por categorías y el proceso de canje, subasta y sorteo sigan funcionando correctamente después de realizar cambios en el front-end.

Pruebas de Interfaz de Usuario: Verificar que la interfaz de usuario mantenga su coherencia visual y usabilidad. Esto incluye la verificación de la disposición de elementos, colores, tipografía y la respuesta a diferentes resoluciones de pantalla.

Pruebas de Navegación: Verificar que los enlaces y botones de navegación sigan dirigiéndose a las páginas correctas y que los menús desplegables y sistemas de navegación funcionen como se espera.

Pruebas de Compatibilidad con Navegadores: Verificar que la página web sea compatible con una variedad de navegadores web populares, como Chrome, Firefox, Safari y Edge. Esto es especialmente importante si se han realizado cambios en el front-end.

Pruebas de Rendimiento: Evaluar el rendimiento de la página web, como el tiempo de carga de las páginas, el rendimiento de búsqueda y la velocidad de respuesta, para asegurarse de que los cambios no hayan degradado el rendimiento general del sitio.

Pruebas de Integración: Verificar que el sistema de pago permite que el usuario pueda realizar las compras de forma exitosa utilizando su medio de pago preferido.

## Funcionalidades a No Probar

En este plan de pruebas se hará revisión de todos los módulos estéticos, funcionales, no funcionales para garantizar el correcto funcionamiento de la aplicación.

## 

## Enfoque de Pruebas (Estrategia)

A continuación, se detallan el tipo de pruebas que se realizarán para evaluar diversos aspectos de la aplicación web. Entre estas se encuentran las pruebas de funcionalidad, usabilidad, rendimiento y compatibilidad.

Pruebas de Funcionalidad:

* Pruebas de Unidad: Evaluar funciones y componentes frontales individuales para asegurarse de que funcionen correctamente.
* Pruebas de Integración: Verificar que los diferentes elementos frontales se integren sin problemas.
* Pruebas de Sistema: Probar el frontend en su conjunto para asegurarse de que todas las funciones se ejecuten según lo esperado.

Pruebas de Usabilidad:

* Evaluar la interfaz de usuario del frontend para garantizar que sea intuitiva y fácil de usar.
* Comprobar la navegación y la disposición de elementos en el frontend para una experiencia de usuario fluida.

Pruebas de Rendimiento:

* Realizar pruebas de carga del frontend para evaluar el rendimiento bajo carga máxima.
* Medir los tiempos de carga de páginas y la capacidad de respuesta del frontend.

Pruebas de Compatibilidad:

* Verificar la compatibilidad del frontend con una variedad de navegadores web, versiones y dispositivos.
* Asegurarse de que el frontend se vea y funcione correctamente en diferentes resoluciones de pantalla.

## Criterios de Aceptación o Rechazo

Son los criterios que serán considerados para dar por completado el Plan de Pruebas de Software, por ejemplo: Completar 100% de pruebas unitarias, cierto porcentaje de casos exitosos, cobertura de todos los componentes y líneas de código, porcentaje de defectos corregidos, entre otros.

Aceptación

* Alineación con objetivos del proyecto
* Cobertura adecuada
* Casos de prueba claros
* Recursos y cronograma razonables
* Criterios de éxito claros

Rechazo

* Falta de alineación de sus objetivos
* Cobertura insuficiente
* Estrategia de pruebas inadecuada
* Recursos y cronograma irrealistas
* Falta de claridad en los criterios de éxito

## Criterios de Suspensión

Establece claramente bajo qué condiciones se detienen un conjunto de casos de pruebas, por ejemplo en caso de existir defectos que impidan la ejecución de más casos de pruebas, cierto porcentaje de casos fallidos, o cualquier otro que se especifique.

* Cambios críticos en la aplicación
* Problemas de seguridad críticos
* Problemas de rendimiento críticos
* Recursos insuficientes
* Cambios en los objetivos

## Criterios de Reanudación

Luego de haber suspendido las pruebas, aquí se establecen bajo qué criterios se reanudarán.

* Cambios significativos en el código
* Actualizaciones de terceros
* Cambios en el entorno de implementación o nuevos servidores
* Pruebas de regresión
* Calendario de lanzamiento
* Resolución de problemas anteriores
* Aprobación de cambios en el plan
* Actualización de objetivos del proyecto

# 

# Entregables

Los siguientes documentos serán entregados al término del proyecto de manera tal que demuestren la finalización del mismo.

* Plan de Pruebas
* Casos de Pruebas
* Especificación de Diseño de Casos
* Registro de Errores
* Reportes de Incidencias
* Evidencias de Pruebas
* Reportes Emitidos por Herramientas de Pruebas
* Informe de Resultados de Pruebas
* Informe de Rendimiento y Carga
* Documentación de Aprobación de Pruebas
* Planes de Acción y Seguimiento
* Evidencia de Cumplimiento de Estándares de Calidad

# Recursos

## Requerimientos de Entornos – Hardware

Servidores de Aplicación:

* Servidores web para alojar la aplicación.
* Servidores de aplicación para ejecutar la lógica de la aplicación.
* Servidores de prueba o entornos de preproducción para replicar el entorno de producción.

Bases de Datos:

* Servidores de bases de datos (por ejemplo, MySQL, PostgreSQL, SQL Server, Oracle) para alojar la base de datos del sistema.
* Copias de bases de datos de prueba para realizar pruebas sin afectar los datos de producción.

Conectividad a la Red:

* Acceso a una red local o internet, según corresponda, para acceder a la aplicación y la base de datos.
* Conexiones a internet confiables para pruebas de rendimiento y carga.

Ambientes de Desarrollo y Pruebas:

* Entornos de desarrollo y pruebas separados de producción, si es posible, para aislar las pruebas de los usuarios finales.

Almacenamiento:

* Espacio de almacenamiento para respaldos de datos y resultados de pruebas.

Dispositivos Móviles (si es aplicable):

* Dispositivos móviles (teléfonos, tablets) para probar la aplicación en plataformas móviles.

Copias de Seguridad y Recuperación de Datos:

* Plan de copias de seguridad y recuperación en caso de pérdida de datos durante las pruebas.

Accesos y Credenciales:

* Proporcionar a los testers las credenciales de acceso necesarias para las diferentes partes del sistema.

## Requerimientos de Entornos – Software

Acceso a Sistemas y Bases de Datos:

* Cuentas y credenciales de acceso a sistemas y bases de datos de prueba.
* Acceso a entornos de prueba y desarrollo, si es necesario.

Herramientas de Pruebas de Carga:

* Herramientas de pruebas de carga y rendimiento, como Apache JMeter, LoadRunner, o herramientas específicas para simular cargas de usuarios.

Herramientas de Registro y Monitoreo:

* Herramientas de registro y monitorización para seguir el comportamiento de la aplicación durante las pruebas, como Splunk, ELK Stack, entre otros.

Navegadores Web:

* Navegadores web compatibles para pruebas de compatibilidad, como Chrome, Firefox, Safari, Edge, etc.

Software de Virtualización/Contenedores:

* Software de virtualización (por ejemplo, VMware, VirtualBox) o plataformas de contenedores (por ejemplo, Docker) si se utilizan entornos virtuales.

Herramientas de Colaboración:

* Herramientas de colaboración y comunicación, como Slack, Microsoft Teams, o herramientas de videoconferencia.

Herramientas de Desarrollo:

* Entornos de desarrollo integrados (IDE) y herramientas de desarrollo necesarias para los programadores que corregirán errores.

Bases de Datos de Prueba:

* Copias de bases de datos de prueba que se utilizarán para realizar pruebas sin afectar los datos de producción.

Herramientas de Seguridad:

* Herramientas de seguridad y escaneo de vulnerabilidades si se están realizando pruebas de seguridad.

Sistemas de Gestión de Versiones:

* Software de control de versiones (por ejemplo, Git) para el seguimiento de cambios en el código y colaboración entre equipos.

## Herramientas de Pruebas Requeridas

No se dispondrán de herramientas de pruebas, ya que el proyecto será revisado minuciosamente por el equipo de trabajo para su buen funcionamiento.

## 

## Personal

Líder de Pruebas:

Responsable de la planificación, coordinación y supervisión de todas las actividades de pruebas.

Define la estrategia de pruebas y garantiza que se siga.

Coordina la comunicación entre el equipo de pruebas y otros equipos de desarrollo.

Analista de Pruebas (Testers):

Responsables de ejecutar las pruebas de acuerdo con los casos de prueba y escenarios definidos.

Documentan y reportan los resultados de las pruebas, incluyendo cualquier defecto encontrado.

Participan en la identificación y elaboración de casos de prueba.

Analista de Pruebas de Carga y Rendimiento:

Diseña y ejecuta pruebas de carga y rendimiento para evaluar el comportamiento de la aplicación bajo cargas simuladas.

Analiza y documenta los resultados de las pruebas de rendimiento.

Identifica y soluciona cuellos de botella de rendimiento.

Especialista en Pruebas de Seguridad:

Realiza pruebas de seguridad para identificar vulnerabilidades y posibles amenazas en la aplicación.

Ayuda en la corrección de problemas de seguridad y garantiza que la aplicación cumpla con los estándares de seguridad.

Especialista en Pruebas de Usabilidad:

Evalúa la usabilidad de la aplicación, centrándose en la experiencia del usuario.

Proporciona recomendaciones para mejorar la usabilidad y la interfaz de usuario.

Coordinador de Pruebas:

Ayuda en la programación y coordinación de las actividades de pruebas.

Mantiene un registro de los casos de prueba y los resultados de las pruebas.

Facilita la comunicación dentro del equipo de pruebas.

Especialista en Gestión de Configuración:

Responsable de la gestión de configuración de los elementos de prueba, como versiones de software, casos de prueba y datos de prueba.

Garantiza que se utilice la versión correcta del software durante las pruebas.

Especialista en Documentación de Pruebas:

Encargado de documentar los procedimientos de pruebas, los casos de prueba y los resultados de pruebas.

Mantiene registros precisos de las actividades de pruebas.

Especialista en Gestión de Defectos:

Supervisa y gestiona el proceso de seguimiento y resolución de defectos.

Colabora con los equipos de desarrollo para asegurarse de que los defectos se corrijan y se vuelvan a probar.

## Entrenamiento

El entrenamiento es crucial para asegurar que el personal encargado de las pruebas esté bien preparado y sea capaz de llevar a cabo las actividades de prueba de manera efectiva.

Entrenamiento en la Aplicación o Sistema:

Familiarización con la funcionalidad y los procesos de la aplicación o sistema que se va a probar.

Comprender los objetivos del proyecto y los requisitos específicos de la aplicación.

Entrenamiento en Requisitos de Negocio y Casos de Uso:

Conocimiento profundo de los requisitos del negocio y los casos de uso para guiar el diseño de casos de prueba efectivos.

Entrenamiento en Metodologías de Pruebas:

Conocimiento de las metodologías de pruebas aplicables, como pruebas ágiles, pruebas de caja negra y caja blanca, pruebas de rendimiento, entre otras.

Entrenamiento en Escenarios de Pruebas Específicos:

Entrenamiento en la preparación y ejecución de casos de prueba específicos para escenarios críticos o de alto riesgo.

Entrenamiento en Pruebas de Carga y Rendimiento:

Comprender cómo diseñar, ejecutar y analizar pruebas de carga y rendimiento, incluyendo la simulación de múltiples usuarios y la monitorización de recursos.

Entrenamiento en Pruebas de Seguridad:

Conocimiento de las técnicas y herramientas utilizadas en pruebas de seguridad, incluyendo pruebas de penetración y escaneo de vulnerabilidades.

Entrenamiento en Pruebas de Usabilidad:

Conocimiento de cómo diseñar y llevar a cabo pruebas de usabilidad para evaluar la experiencia del usuario.

Entrenamiento en Pruebas de Integración y Aceptación del Cliente:

Comprender cómo diseñar y ejecutar pruebas de integración para garantizar que los módulos de la aplicación funcionen juntos de manera efectiva.

Entender los procesos de aceptación del cliente y cómo realizar pruebas de aceptación.

Entrenamiento en Procesos y Normas de Calidad:

Conocimiento de los estándares y normas de calidad relevantes, como ISO 9001 o CMMI, si son aplicables al proyecto.

# Planificación y Organización

## Procedimientos para las Pruebas

Pruebas Funcionales:

Pruebas de Caja Negra: Se centran en evaluar la funcionalidad de la aplicación sin acceder al código fuente. Se utilizan casos de prueba basados en requisitos y escenarios de uso.

Pruebas de Regresión:

Aseguran que las nuevas actualizaciones o cambios en el software no hayan afectado negativamente las funcionalidades existentes.

Pruebas de Integración:

Evalúan la interacción entre diferentes componentes o módulos del software para garantizar que funcionen juntos de manera coherente.

Pruebas de Usabilidad:

Se centran en la experiencia del usuario, evaluando la facilidad de uso, la navegación y la accesibilidad de la aplicación.

Pruebas de Carga y Rendimiento:

Comprueban cómo se comporta la aplicación bajo diversas cargas de usuarios o condiciones de carga. Pueden incluir pruebas de estrés y pruebas de rendimiento.

Pruebas de Seguridad:

Se centran en identificar vulnerabilidades y amenazas de seguridad en la aplicación. Incluyen pruebas de penetración y escaneo de vulnerabilidades.

Pruebas de Aceptación del Cliente:

Evaluadas por los usuarios finales o los clientes para determinar si el software cumple con sus requisitos y expectativas.

Pruebas Exploratorias:

Los testers exploran el software de manera informal para descubrir defectos sin seguir casos de prueba predefinidos.

Pruebas Ágiles:

Las pruebas se integran en el ciclo de desarrollo ágil y se ejecutan continuamente a medida que se construye el software.

Pruebas de Compatibilidad:

Aseguran que la aplicación funcione correctamente en diversos navegadores, sistemas operativos y dispositivos.

## 

## 

## Matriz de Responsabilidades



## Cronograma

Iniciando desde el 28 de octubre hasta el 14 de noviembre

Día 1: Planificación y Organización

Revisión de los requisitos del proyecto y definición de objetivos para las pruebas.

Identificación de herramientas necesarias.

Asignación de roles y responsabilidades.

Creación de un plan detallado de pruebas.

Día 2: Procedimientos para las Pruebas

Redacción de los procedimientos de prueba.

Definición de criterios de aceptación y priorización de pruebas.

Configuración del entorno de prueba.

Validación del plan con los stakeholders.

Día 3: Pruebas Funcionales

Desarrollo de casos de prueba para Pruebas de Caja Negra.

Priorización según funcionalidades críticas.

Ejecución inicial de Pruebas de Caja Negra.

Documentación de resultados.

Día 4: Pruebas de Regresión y Pruebas de Integración

Selección de escenarios para Pruebas de Regresión.

Preparación de casos de prueba.

Ejecución de pruebas para validar la integración entre módulos.

Día 5: Pruebas de Usabilidad

Planificación de pruebas con usuarios finales.

Creación de métricas para evaluar la experiencia del usuario.

Realización de pruebas con un grupo representativo de usuarios.

Semana 2

Día 6: Pruebas de Carga y Rendimiento

Configuración de herramientas para pruebas de rendimiento.

Creación de escenarios de carga y estrés.

Ejecución de pruebas y análisis de métricas.

Día 7: Pruebas de Seguridad

Identificación de áreas críticas para pruebas de seguridad.

Ejecución de pruebas de penetración.

Escaneo de vulnerabilidades con herramientas especializadas.

Día 8: Pruebas de Aceptación del Cliente

Preparación de un entorno para demostraciones.

Ejecución de pruebas con los clientes.

Recolección de feedback y ajustes necesarios.

Día 9: Pruebas Exploratorias y Ágiles

Ejecución de pruebas exploratorias para identificar defectos no cubiertos.

Integración de pruebas dentro del flujo ágil.

Día 10: Pruebas de Compatibilidad

Definición de combinaciones de dispositivos, sistemas operativos y navegadores a probar.

Ejecución de pruebas y análisis de resultados.

Revisión y Ajustes (Días 11 y 12)

Consolidación de resultados de todas las pruebas.

Análisis de defectos encontrados y priorización para resolución.

Actualización del informe de pruebas.

Entrega Final (Días 13 y 14)

Presentación de informes finales a los stakeholders.

Validación de cumplimiento de requisitos.

Cierre de actividades y retroalimentación.

## Premisas

Disponibilidad de Recursos:

Se asume que los recursos necesarios, incluyendo personal, hardware y software, estarán disponibles de acuerdo con la planificación.

Cronograma del Proyecto:

Se asume que el cronograma del proyecto general es válido y que las actividades de pruebas se llevarán a cabo según lo planeado.

Acceso a Ambientes de Pruebas:

Se asume que los ambientes de pruebas, incluyendo servidores de prueba y bases de datos, estarán disponibles y configurados de acuerdo con las necesidades del equipo de pruebas.

Colaboración del Equipo de Desarrollo:

Se asume que el equipo de desarrollo colaborará en la corrección de defectos y en la realización de pruebas de regresión de manera oportuna.

Materiales de Prueba:

Se asume que se contarán con materiales de prueba, como documentación de requisitos, casos de prueba y datos de prueba, que estarán listos y disponibles para su uso.

Uso de Metodología de Pruebas:

Se asume que se seguirá una metodología de pruebas específica, como las pruebas ágiles, pruebas de caja negra o pruebas de caja blanca, según se haya definido en el Plan de Pruebas de Software.

Uso de Herramientas de Pruebas:

Se asume que se utilizarán herramientas de pruebas específicas, como herramientas de automatización, herramientas de gestión de pruebas y herramientas de pruebas de carga, de acuerdo con lo que se ha planificado.

Colaboración de Usuarios Finales (para Pruebas de Aceptación):

Se asume que los usuarios finales o los clientes estarán disponibles y dispuestos a participar en las pruebas de aceptación cuando sea necesario.

Disponibilidad de Datos de Producción (para Pruebas de Datos Realistas):

Se asume que se dispondrá de datos de producción realistas para realizar pruebas de datos y pruebas de rendimiento realistas.

Cumplimiento de Estándares y Normas:

Se asume que el software cumple con los estándares y normas de calidad relevantes, y que las pruebas se basarán en estos estándares.

## Dependencias y Riesgos

Riesgo 1: Dependencias con Desarrollos

Descripción: Las pruebas dependen de la finalización de ciertas funcionalidades o módulos en desarrollo, lo que puede retrasar las actividades de pruebas.

Probabilidad: Moderada

Impacto: Alto

Plan de Mitigación: Establecer una comunicación regular entre el equipo de desarrollo y el equipo de pruebas para garantizar que las funcionalidades estén listas a tiempo. Utilizar un enfoque de pruebas paralelas donde sea posible, comenzando las pruebas en módulos completos antes de que todo el desarrollo esté finalizado.

Plan de Contingencia: Redefinir las fechas de pruebas y ajustar el cronograma si las dependencias con el desarrollo no se pueden resolver.

Riesgo 2: Dependencias con Otros Proyectos

Descripción: Las pruebas pueden verse afectadas si dependen de recursos o componentes proporcionados por otros proyectos.

Probabilidad: Baja

Impacto: Moderado

Plan de Mitigación: Identificar y comunicarse con los responsables de otros proyectos para garantizar la disponibilidad de los recursos necesarios. Establecer acuerdos formales de colaboración.

Plan de Contingencia: Buscar alternativas o recursos internos en caso de que las dependencias con otros proyectos no se cumplan.

Riesgo 3: Disponibilidad de Recursos

Descripción: La falta de recursos, como personal de pruebas, hardware o herramientas, puede retrasar o limitar la ejecución de las pruebas.

Probabilidad: Moderada

Impacto: Alto

Plan de Mitigación: Asignar y asegurar la disponibilidad de recursos con suficiente antelación. Establecer planes de contingencia si es necesario contratar personal adicional o adquirir recursos adicionales.

Plan de contingencia: Reasignar recursos internos o buscar proveedores externos si no se pueden obtener los recursos planificados.

Riesgo 4: Restricciones de Tiempo

Descripción: Plazos ajustados pueden limitar el tiempo disponible para realizar pruebas exhaustivas.

Probabilidad: Alta

Impacto: Alto

Plan de Mitigación: Establecer un cronograma de pruebas realista desde el principio y priorizar las pruebas más críticas. Utilizar técnicas de pruebas ágiles si es necesario para adaptarse a plazos ajustados.

Plan de Contingencia: Considerar la posibilidad de aplazar algunas pruebas no críticas o reducir la cobertura de pruebas en caso de que los plazos no puedan cumplirse.

Riesgo 5: Premisas que Resulten no Ser Ciertas

Descripción: Las suposiciones incorrectas hechas en el plan de pruebas pueden afectar la ejecución de las pruebas y la calidad del software.

Probabilidad: Moderada

Impacto: Moderado

Plan de Mitigación: Revisar y validar las premisas en el Plan de Pruebas de Software antes de iniciar las actividades de pruebas. Comunicarse regularmente con los interesados para asegurarse de que las premisas se mantengan ciertas.

Plan de Contingencia: Realizar ajustes en el plan de pruebas y en la estrategia de pruebas si se descubre que las premisas no son ciertas durante la ejecución de las pruebas.

# Referencias

Plan de Proyecto:

El plan de proyecto general proporciona un contexto para las actividades de pruebas, incluyendo plazos, recursos y objetivos del proyecto.

Especificaciones de Requerimientos:

Las especificaciones de requerimientos del software son fundamentales para diseñar casos de prueba que cubran todos los requisitos funcionales y no funcionales.

Diseño General y Detallado:

El diseño general y detallado del software proporciona información importante sobre la arquitectura y estructura del sistema, lo que puede influir en la estrategia de pruebas.

Procedimientos y Estándares de Desarrollo:

Los procedimientos y estándares de desarrollo del software pueden ayudar a comprender cómo se construyó el software y qué prácticas de desarrollo se siguieron.

Procedimientos y Estándares de Pruebas:

Los procedimientos y estándares de pruebas definen las prácticas y enfoques que se utilizan en el proceso de pruebas, lo que es relevante para la estrategia de pruebas.

Metodologías, Procedimientos y Estándares Corporativos:

Las metodologías y estándares corporativos pueden incluir prácticas y directrices específicas para

las pruebas de software, y es importante seguirlos para garantizar la consistencia y la calidad.

Documentación de Herramientas de Pruebas:

Si se utilizan herramientas de pruebas específicas, la documentación de estas herramientas puede ser una referencia valiosa para comprender cómo configurar y utilizar las herramientas de manera efectiva.

Literatura Técnica y de Pruebas:

Libros, artículos y recursos en línea relacionados con la ingeniería de pruebas de software y las mejores prácticas de pruebas pueden proporcionar información adicional.

Normas y Regulaciones de la Industria:

Si el software debe cumplir con estándares o regulaciones específicas de la industria, es importante citar y seguir estos estándares en el plan de pruebas.

Registros de Pruebas Anteriores:

Si se trata de una versión posterior de un software existente, los registros de pruebas anteriores pueden ayudar a comprender los problemas y los resultados de las pruebas previas.

# 

# Glosario

Caso de Prueba: Un conjunto de condiciones o pasos específicos que se utilizan para probar una característica o funcionalidad del software.

Pruebas de Caja Negra: Un enfoque de prueba que se centra en la funcionalidad del software sin conocer su estructura interna.

Pruebas de Caja Blanca: Un enfoque de prueba que evalúa la estructura interna del software y el código fuente.

Pruebas de Regresión: Pruebas realizadas para asegurarse de que las nuevas actualizaciones o cambios en el software no hayan afectado las funcionalidades existentes.

Pruebas de Integración: Pruebas que evalúan la interacción y el funcionamiento conjunto de los módulos o componentes del software.

Pruebas de Aceptación del Cliente: Pruebas realizadas por los usuarios finales o clientes para determinar si el software cumple con sus necesidades y expectativas.

Pruebas de Carga y Rendimiento: Pruebas que evalúan cómo se comporta el software bajo diferentes condiciones de carga y estrés.

Pruebas de Seguridad: Pruebas que identifican vulnerabilidades y riesgos de seguridad en el software.

Pruebas de Usabilidad: Pruebas para evaluar la facilidad de uso y la experiencia del usuario del software.

Automatización de Pruebas: El proceso de crear scripts o programas para ejecutar pruebas de software de manera automatizada.

Incidente o Defecto: Cualquier anomalía o problema detectado durante las pruebas que necesita ser corregido.

Ciclo de Vida de Pruebas: El conjunto de fases y actividades que abarcan el proceso de pruebas, desde la planificación hasta la finalización.

Ambiente de Pruebas: Un entorno aislado y controlado donde se realizan las pruebas de software.

Estrategia de Pruebas: Un plan global que describe el enfoque y los recursos necesarios para las pruebas.

Plan de Pruebas de Software: Un documento que describe cómo se llevarán a cabo las pruebas, incluyendo el alcance, los objetivos, los recursos y el cronograma.

Matriz de Riesgos: Una tabla que identifica y evalúa los riesgos asociados con las pruebas de software y define planes de mitigación y contingencia.

Pruebas de Humo (Smoke Tests): Pruebas iniciales para verificar que las funcionalidades críticas funcionen antes de realizar pruebas más exhaustivas.

Pruebas Exploratorias: Pruebas ad hoc realizadas por testers para descubrir defectos sin seguir casos de prueba predefinidos.

Fase de Análisis de Pruebas: La etapa en la que se recopilan y analizan los resultados de las pruebas.