**Plan de Pruebas**

**Spintech**

**10-10-2024**

**Versión 1.0**

# Histórico de Revisiones

| **Fecha** | **Versión** | **Descripción** | **Autor** |
| --- | --- | --- | --- |
| 15/10/2024 | 1.0 | Creación del documento de plan de pruebas. | Bárbara Barraza |
| 22/10/2024 | 1.1 | Actualización de casos de prueba y ajuste de cronograma | Bárbara Barraza |
| 11/01/2024 | 1.2 | Revisión y validación de pruebas de regresión | Bárbara Barraza |
| 11/10/2024 | 1.3 | Revisión final y ajuste de métricas de calidad. | Bárbara Barraza |

**Contenido**

[Histórico de Revisiones 2](#_heading=h.30j0zll)

[1 Introducción 4](#_heading=h.1fob9te)

[1.1 Resumen 4](#_heading=h.3znysh7)

[1.2 Descripción breve del proyecto 5](#_heading=h.2et92p0)

[1.3 Ambiente de producción 6](#_heading=h.tyjcwt)

[2 Alcance 8](#_heading=h.3dy6vkm)

[2.1 Módulos de Software que van a ser probados 8](#_heading=h.1t3h5sf)

[2.2 Documentación Referenciada 11](#_heading=h.4d34og8)

[2.3 Riesgos 14](#_heading=h.2s8eyo1)

[3 Descripción de las pruebas 16](#_heading=h.17dp8vu)

[3.1 Enfoque de las pruebas: 16](#_heading=h.3rdcrjn)

[3.2 Tipos de pruebas a realizar: 19](#_heading=h.26in1rg)

[3.3 Estrategia y técnicas de pruebas a aplicar 22](#_heading=h.lnxbz9)

[3.4 Requerimientos del test 25](#_heading=h.35nkun2)

[3.5 Clasificación de los defectos 28](#_heading=h.1ksv4uv)

[3.6 Roles y responsabilidades 29](#_heading=h.44sinio)

[3.7 Calendarización de las actividades de pruebas 32](#_heading=h.z337ya)

# 1 Introducción

## 1.1 Resumen

Este Plan de Pruebas documentado para el proyecto "Spintech" tiene los siguientes objetivos:

* **Identificar qué va a ser probado:** Se enfoca en las funcionalidades clave del sistema, incluyendo módulos de registro de usuarios, gestión de cursos, soporte de accesibilidad, generación de estadísticas y configuraciones administrativas. También se valida el cumplimiento de los requisitos no funcionales, como seguridad, usabilidad y desempeño.
* **Resumir el enfoque general usado para las pruebas:** El enfoque utilizado es el de pruebas de caja negra, con técnicas basadas en escenarios y casos de uso. Esto asegura que cada funcionalidad cumpla con los requisitos definidos sin necesidad de evaluar el código subyacente. Las pruebas incluyen pruebas de funcionalidad, integración, sistema y aceptación.
* **Resumir los objetivos y las metodologías utilizadas:** El objetivo principal de las pruebas es garantizar que el sistema cumpla con los requisitos especificados, tanto funcional como no funcional, minimizando los defectos críticos antes de su despliegue. Se emplearon metodologías como:
  + Diseño de casos de prueba basados ​​en requisitos.
  + Validación y verificación a través de escenarios reales.
  + Documentación de resultados y retroalimentación para los desarrolladores.

## 1.2 Descripción breve del proyecto

El proyecto **"Spintech"** consiste en el desarrollo de una plataforma en línea diseñada para proporcionar una experiencia educativa inclusiva, accesible y personalizada. El sistema está dirigido tanto a estudiantes con discapacidades como a instructores y administradores, con el objetivo de promover una educación equitativa.

La plataforma incluye los siguientes módulos y características principales:

* **Registro de usuarios e inicio de sesión** : Permite la creación de cuentas para estudiantes, instructores y administradores, asegurando roles y permisos adecuados.
* **Gestión de cursos y recursos** : Los instructores pueden crear, editar y gestionar cursos y recursos asociados (videos, documentos, entre otros).
* **Accesibilidad** : Ofrece soporte para lenguaje de señas, herramientas de navegación accesibles y diseño optimizado para dispositivos móviles.
* **Personalización del aprendizaje** : Los usuarios pueden filtrar cursos por categoría y visualizar su progreso, adaptándose a sus necesidades específicas.
* **Estadísticas y seguimiento** : Los administradores tienen acceso a métricas y estadísticas clave para evaluar el rendimiento del sistema y de los usuarios.
* **Seguridad y cumplimiento de estándares** : La plataforma prioriza la seguridad de los datos, la privacidad de los usuarios y la conformidad con los estándares web.

## 1.3 Ambiente de producción

El ambiente de producción del proyecto **"Spintech"** ha sido diseñado para garantizar la estabilidad, escalabilidad y accesibilidad de la plataforma. Este ambiente incluye los siguientes aspectos clave:

#### Infraestructura

* **Servidor web**: Apache o Nginx, configurado para optimizar la entrega de contenidos y garantizar un tiempo de respuesta bajo.
* **Base de datos**: MySQL, con mecanismos de respaldo automatizados para garantizar la integridad y recuperación de datos.
* **Lenguaje de programación**: PHP (versión 7.4 o superior) en el backend, con soporte de frameworks como Laravel para la gestión eficiente de funcionalidades.
* **Frontend**: HTML5, CSS y JavaScript, con el uso de frameworks modernos como Bootstrap para garantizar un diseño responsivo.
* **Servicios en la nube** : AWS o Google Cloud, configurados para el alojamiento seguro y escalable de la plataforma.

#### Accesibilidad

* Configuración para el soporte de herramientas de asistencia como lectores de pantalla y funcionalidades de navegación por teclado.
* Implementación de soporte multilingüe para garantizar el acceso a usuarios de diversas regiones.

#### Seguridad

* Certificados SSL para garantizar una comunicación cifrada mediante HTTPS.
* Implementación de políticas de acceso basadas en roles (usuarios, administradores, profesores).
* Almacenamiento seguro de contraseñas mediante técnicas de hash.

#### Plataformas soportadas

* **Navegadores:** Compatibilidad con las versiones más recientes de Google Chrome, Mozilla Firefox, Microsoft Edge y Safari.
* **Dispositivos:** Optimización para uso en computadoras de escritorio, portátiles y tabletas.

#### Configuraciones específicas

* **Monitorización y registros**: Herramientas configuradas para supervisar el rendimiento del sistema y registrar eventos críticos.
* **Capacidades de escalado**: Configuración para soportar un número creciente de usuarios, ajustando automáticamente los recursos del servidor según la demanda.

# 2 Alcance

## 2.1 Módulos de Software que van a ser probados

#### 2.1 Módulos de software que van a ser probados

El siguiente Plan de Pruebas abarca los módulos principales de la **"Spintech"** , con el propósito de garantizar su funcionalidad, usabilidad y accesibilidad:

##### 1. Registro y Autenticación de Usuarios

* **Funcionalidad** : Permitir el registro de nuevos usuarios y su autenticación mediante credenciales únicas.
* **Pruebas principales** :
  + Validación de campos obligatorios (correo electrónico, contraseña, nombre).
  + Verificación de contraseñas seguras.
  + Recuperación de contraseña mediante correo electrónico.
  + Restricción de acceso a usuarios no autenticados.

##### 2. Gestión de cursos

* **Funcionalidad** : Creación, edición, eliminación y asignación de cursos por parte de los profesores.
* **Pruebas principales** :
  + Creación de cursos con detalles específicos como nombre, descripción y recursos asociados.
  + Validación de permisos de edición según el rol del usuario.
  + Verificación de la visualización correcta de cursos disponibles para los estudiantes.

##### 3. Visualización de Contenido

* **Funcionalidad** : Permitir a los usuarios acceder al contenido del curso, incluyendo vídeos, módulos y materiales adicionales.
* **Pruebas principales** :
  + Reproducción de vídeos y recursos multimedia.
  + Carga rápida de módulos y contenidos.
  + Soporte para subtítulos y lengua de señas en contenido multimedia.

##### 4. Estadísticas de uso

* **Funcionalidad** : Brindar al administrador una vista general sobre el progreso de los estudiantes, la cantidad de cursos activos y métricas de uso.
* **Pruebas principales** :
  + Cálculo correcto de estadísticas como el progreso del curso.
  + Filtrado y visualización de estadísticas por usuario o curso.
  + Protección contra accesos no autorizados.

##### 5. Configuración de usuario

* **Funcionalidad** : Edición de perfil, cambios de contraseña y ajustes de preferencias personales.
* **Pruebas principales** :
  + Actualización de datos personales como correo electrónico, teléfono y dirección.
  + Cambios de contraseña con validación de contraseñas actuales y nuevas.
  + Comprobación de restricciones para usuarios no autenticados.

##### 6. Filtros y Búsqueda

* **Funcionalidad** : Filtrar cursos según categorías y palabras clave.
* **Pruebas principales** :
  + Búsqueda precisa de cursos según el criterio introducido.
  + Validación de resultados cuando no se encuentran coincidencias.
  + Verificación de la funcionalidad en dispositivos móviles.

##### 7. Seguridad

* **Funcionalidad** : Garantizar la protección de datos de los usuarios y del sistema.
* **Pruebas principales** :
  + Verificación de conexiones HTTPS y certificados SSL.
  + Pruebas de integridad en el almacenamiento de contraseñas.
  + Validación de roles y permisos de acceso.

##### 8. Compatibilidad y Accesibilidad

* **Funcionalidad** : Asegurar el correcto funcionamiento de la plataforma en diferentes dispositivos y navegadores, con soporte para usuarios con discapacidades.
* **Pruebas principales** :
  + Validación de compatibilidad en navegadores populares.
  + Pruebas de navegación mediante teclado y lectores de pantalla.
  + Verificación de diseño responsivo en dispositivos móviles.

## 2.2 Documentación Referenciada

Para el desarrollo de este Plan de Pruebas se han tomado como referencia los siguientes documentos y materiales relacionados con el proyecto **"Spintech"**:

#### 1. Requerimientos Funcionales y No Funcionales

* Documento que describe detalladamente los requerimientos funcionales y no funcionales del sistema.
* **Versión:** 1.2
* **Fecha:** 23-09-2024
* **Propósito:** Servir como base para la elaboración de los casos de prueba asociados a las funcionalidades del sistema.

#### 2. Matriz de Gestión y Control de Cambios

* Documento que registra y gestiona los cambios solicitados durante el desarrollo del proyecto.
* **Versión:** 1.0
* **Fecha:** 22-10-2024
* **Propósito:** Identificar los cambios que puedan impactar las pruebas y garantizar que los casos de prueba reflejen las modificaciones aprobadas.

#### 3. Manual de Usuario

* Documento que explica el uso del sistema desde la perspectiva del usuario final.
* **Versión:** 1.1
* **Fecha:** 11-11-2024
* **Propósito:** Servir como guía para validar la funcionalidad y la usabilidad del sistema durante las pruebas.

#### 4. Casos de Uso

* Diagrama y descripción de los casos de uso que representan las interacciones entre los usuarios y el sistema.
* **Versión:** 2.0
* **Fecha:** 13-11-2024
* **Propósito:** Proporcionar un punto de partida para la creación de casos de prueba basados en los flujos funcionales del sistema.

#### 5. Minuta de Control de la Programación

* Documento que detalla los avances, desviaciones y ajustes realizados durante el desarrollo del proyecto.
* **Versión:** 1.0
* **Fecha:** 22-10-2024
* **Propósito:** Garantizar que el alcance de las pruebas considere las funcionalidades implementadas en las iteraciones actuales.

#### 6. Matriz de Casos de Prueba

* Registro de todos los casos de prueba diseñados para el proyecto.
* **Versión:** 1.0
* **Fecha:** 12-11-2024
* **Propósito:** Centralizar los casos de prueba asociados a los requerimientos funcionales y no funcionales.

#### 7. Documentación Técnica

* Detalle técnico del diseño de la base de datos, la arquitectura del sistema y la implementación del backend.
* **Versión:** 1.3
* **Fecha:** 13-11-2024
* **Propósito:** Proveer información técnica para la validación de pruebas específicas, como las relacionadas con la seguridad, el rendimiento y la integridad de los datos.

## 2.3 Riesgos

En esta sección se identifican los principales riesgos asociados con las pruebas del proyecto **"Spintech"**, así como las estrategias de mitigación propuestas:

#### Riesgo 1: Incompletitud de Casos de Prueba

* **Descripción:** Existe el riesgo de que los casos de prueba no cubran todos los escenarios posibles debido a malentendidos en los requerimientos.
* **Impacto:** Omisión de errores críticos en funcionalidades clave.
* **Estrategia de Mitigación:**
  + Revisar los requerimientos funcionales y no funcionales con los interesados antes de crear los casos de prueba.
  + Realizar revisiones cruzadas entre los equipos de QA y desarrollo para asegurar la cobertura adecuada.

#### Riesgo 2: Dependencia de Entregas de Desarrollo

* **Descripción:** Los retrasos en las entregas de funcionalidades por parte del equipo de desarrollo pueden afectar el inicio oportuno de las pruebas.
* **Impacto:** Retraso en la validación de funcionalidades críticas, afectando los cronogramas.
* **Estrategia de Mitigación:**
  + Establecer puntos de control semanales con el equipo de desarrollo para monitorear el progreso.
  + Implementar pruebas parciales o en entornos simulados si las funcionalidades completas aún no están disponibles.

#### Riesgo 3: Recursos Limitados para las Pruebas

* **Descripción:** El equipo de pruebas podría no contar con los recursos humanos o tecnológicos suficientes para ejecutar todas las pruebas en el tiempo asignado.
* **Impacto:** Reducción de la cobertura de pruebas, aumentando el riesgo de errores no detectados.
* **Estrategia de Mitigación:**
  + Priorizar las pruebas de funcionalidades críticas basándose en el impacto y la frecuencia de uso.
  + Automatizar pruebas repetitivas y de regresión para optimizar el tiempo del equipo.

#### Riesgo 4: Cambios No Planificados en los Requerimientos

* **Descripción:** Cambios tardíos en los requerimientos pueden impactar el diseño de los casos de prueba y generar inconsistencias.
* **Impacto:** Re-trabajo significativo y posibles omisiones de pruebas.
* **Estrategia de Mitigación:**
  + Mantener un control riguroso de cambios mediante la Matriz de Gestión de Control de Cambios.
  + Comunicar los cambios al equipo de pruebas de forma inmediata para ajustar los casos de prueba.

#### Riesgo 5: Incompatibilidades en el Entorno de Pruebas

* **Descripción:** El sistema podría no comportarse correctamente en todos los navegadores o dispositivos debido a diferencias en configuraciones.
* **Impacto:** Problemas de accesibilidad y funcionalidad en el entorno del usuario final.
* **Estrategia de Mitigación:**
  + Ejecutar pruebas de compatibilidad en múltiples navegadores y dispositivos clave.
  + Establecer un entorno de pruebas que refleje el entorno de producción lo más fielmente posible.

#### Riesgo 6: Problemas de Seguridad y Privacidad

* **Descripción:** El sistema podría estar expuesto a vulnerabilidades de seguridad que comprometen los datos del usuario.
* **Impacto:** Violación de la privacidad de los usuarios y pérdida de confianza en la plataforma.
* **Estrategia de mitigación:**
  + Realizar auditorías de seguridad periódicas.
  + Probar escenarios de ataques simulados para evaluar la fortaleza de las medidas de seguridad implementadas.

# 3 Descripción de las pruebas

## 3.1 Enfoque de las pruebas:

El enfoque principal de las pruebas para el proyecto **"Spintech"** se centra en garantizar que el sistema cumpla con los requerimientos funcionales y no funcionales definidos, detectando y corrigiendo defectos antes de la implementación en producción. A continuación, se describe el enfoque:

**1. Cobertura de Funcionalidades:**

* Se asegurará que todas las funcionalidades clave sean evaluadas, desde el registro de usuarios hasta las estadísticas para el administrador, incluyendo módulos de accesibilidad como soporte para lengua de señas y multilingüe.

**2. Pruebas Orientadas a Defectos:**

* Las pruebas estarán dirigidas a identificar defectos en cada etapa del desarrollo, priorizando aquellas áreas con mayor criticidad, como:
  + Seguridad de los datos de usuario.
  + Visualización de estadísticas para el administrador.
  + Compatibilidad y accesibilidad en diferentes navegadores y dispositivos.

**3. Validación Contra Requerimientos:**

* Se validará que cada componente desarrollado cumpla con los requerimientos establecidos, tanto funcionales como no funcionales. Esto incluye:
  + Correcto funcionamiento de filtros por categoría en la búsqueda de cursos.
  + Adecuada visualización del progreso del curso y generación de estadísticas.

**4. Pruebas Integrales:**

* Se realizarán pruebas de integración para verificar la interacción entre los diferentes módulos del sistema, como la relación entre el backend de certificación y el frontend de visualización.

**5. Automatización de Pruebas:**

* Las pruebas repetitivas, como las de regresión, serán automatizadas para optimizar tiempo y recursos, mientras que las pruebas manuales se centrarán en escenarios críticos y casos únicos.

**6. Pruebas de Calidad No Funcional:**

* Se evaluarán atributos de calidad como:
  + **Rendimiento:** Tiempo de respuesta del sistema bajo diferentes cargas.
  + **Seguridad:** Protección de datos confidenciales mediante auditorías y pruebas de penetración.
  + **Usabilidad:** Experiencia del usuario en términos de accesibilidad y facilidad de uso.

**7. Ciclo de Corrección de Defectos:**

* Los defectos identificados serán reportados, categorizados y priorizados para su corrección. Posteriormente, se realizarán pruebas de regresión para garantizar que los cambios no afecten otras áreas del sistema.

**8. Estrategia de Comunicación:**

* Los resultados de las pruebas se comunicarán regularmente al equipo de desarrollo y a los interesados clave, asegurando que las decisiones sobre correcciones y priorizaciones sean informadas.

## 3.2 Tipos de pruebas a realizar:

Para el proyecto **"Spintech"**, se desarrollarán los siguientes tipos de pruebas, priorizando pruebas de confirmación y regresión para garantizar la calidad del sistema:

#### 1. Pruebas de Confirmación

* **Descripción:** Las pruebas de confirmación se enfocan en verificar que los defectos identificados en ciclos previos han sido corregidos correctamente sin introducir nuevos problemas.
* **Enfoque:**
  + Validar que las funcionalidades corregidas cumplen con los requerimientos originales.
  + Realizar pruebas puntuales en las áreas donde se realizaron cambios.
* **Áreas Críticas:**
  + Corrección de problemas de accesibilidad (soporte para lengua de señas, multilingüe).
  + Resolución de defectos en estadísticas del administrador.
  + Confirmar que las modificaciones no afectan la integridad de datos.

#### 2. Pruebas de Regresión

* **Descripción:** Las pruebas de regresión aseguran que las nuevas funciones o correcciones no hayan introducido defectos en otras áreas del sistema.
* **Enfoque:**
  + Ejecutar pruebas completas o parciales en módulos previamente funcionales.
  + Automatizar casos de pruebas repetitivos para optimizar el tiempo.
* **Áreas Críticas:**
  + Interacción entre módulos (backend de estadísticas y frontend de visualización).
  + Comportamiento de filtros por categoría después de agregar cursos.
  + Integridad de datos en el registro y edición de usuarios.

#### 3. Pruebas Funcionales

* **Descripción:** Verifican que cada funcionalidad cumpla con los requerimientos funcionales definidos.
* **Enfoque:**
  + Probar escenarios de usuario final, como la inscripción a cursos, visualización de progresos y generación de certificados.
  + Asegurar el funcionamiento de los módulos clave, como registro, inicio de sesión y edición de perfil.
* **Áreas Críticas:**
  + Flujo de registro e inicio de sesión.
  + Visualización de contenidos (videos y estadísticas).

#### 4. Pruebas No Funcionales

* **Descripción:** Evalúan atributos de calidad como rendimiento, seguridad, compatibilidad y accesibilidad.
* **Enfoque:**
  + Evaluar el tiempo de carga de páginas y la estabilidad bajo carga de usuarios.
  + Realizar pruebas de accesibilidad para garantizar que el sistema sea utilizable para personas con discapacidades.
  + Realizar auditorías de seguridad para proteger datos confidenciales.
* **Áreas Críticas:**
  + Tiempo de respuesta al cargar estadísticas.
  + Seguridad en el manejo de contraseñas.

#### 5. Pruebas de Integración

* **Descripción:** Aseguran que los diferentes módulos del sistema interactúen correctamente.
* **Enfoque:**
  + Verificar la interacción entre backend, base de datos y frontend.
  + Validar la generación de estadísticas basada en datos almacenados.
* **Áreas Críticas:**
  + Comunicación entre módulos de gestión de usuarios, cursos y estadísticas.

#### 6. Pruebas de Usabilidad

* **Descripción:** Evalúan la facilidad de uso y la experiencia del usuario final.
* **Enfoque:**
  + Probar la claridad de las interfaces y la accesibilidad de las funcionalidades.
  + Asegurar que el diseño sea intuitivo y cumpla con estándares de usabilidad.
* **Áreas Críticas:**
  + Diseño del tablero de estadísticas para administradores.
  + Navegación en dispositivos móviles.

## 3.3 Estrategia y técnicas de pruebas a aplicar

Para garantizar la calidad del sistema **"Spintech"** , se empleará una estrategia integral basada en técnicas específicas y actividades orientadas a validar tanto la funcionalidad como los atributos no funcionales del sistema.

#### 1. Hardware que Ejecuta el Sistema

* **Entorno principal de pruebas:**
  + **Servidor:**
    - Procesador: Intel Xeon o equivalente.
    - Memoria RAM: Mínimo 16 GB.
    - Almacenamiento: SSD con al menos 500 GB.
    - Sistema Operativo: Ubuntu Server 20.04 LTS.
  + **Cliente (usuario):**
    - PC de escritorio/portátil: Procesador Intel i5, 8 GB de RAM, navegador actualizado (Chrome, Firefox).
    - Dispositivos móviles: Android 9 o superior, iOS 14 o superior.
    - Resoluciones: 1920x1080, 1366x768 y pantallas móviles.
* **Infraestructura de Red:**
  + Conexión de red mínima de 10 Mbps.
  + Certificados SSL/TLS para garantizar conexiones seguras.

#### 2. Conexiones, Redes y Flujos de Datos

* **Objetivo:** Garantizar que el flujo de datos entre los módulos sea seguro, confiable y eficiente.
* **Acciones:**
  + Verifique que las conexiones entre frontend, backend y base de datos sean estables.
  + Monitorear el tiempo de respuesta de las consultas a la base de datos.
  + Asegurar el cifrado de datos en tránsito mediante HTTPS.
* **Técnicas:**
  + Inspección de red con herramientas como Postman y Wireshark.
  + Simulación de tráfico de red para evaluar la escalabilidad.

#### 3. Pruebas de Caja Negra

* **Objetivo:** Validar que cada módulo cumpla con sus requisitos funcionales sin analizar su implementación interna.
* **Aplicación:**
  + Pruebas en los módulos de registro, inicio de sesión, estadísticas y exploración de cursos.
  + Validación de los filtros por categoría y personalización de contenido.
* **Técnicas:**
  + Generación de casos de prueba basados ​​en los requisitos funcionales y no funcionales.
  + Validación de entradas y salidas esperadas.

#### 4. Actividades desarrolladas por el Equipo de Pruebas

* **Planificación:**
  + Elaboración de casos de prueba detallados por requerimiento.
  + Priorización de pruebas funcionales y de regresión.
* **Ejecución:**
  + Implementación de ciclos de pruebas manuales y automatizadas.
  + Validación de datos en los módulos críticos (estadísticas, seguridad, accesibilidad).
* **Informe:**
  + Documentación de resultados, defectos encontrados y métricas de calidad.
  + Retroalimentación continua para el equipo de desarrollo.

#### 5. Ciclos de Prueba a Ejecutar

1. **Pruebas Unitarias:**
   * Pruebas individuales de los módulos para asegurar que funcionen según lo esperado.
2. **Pruebas de integración:**
   * Validación de la interacción entre módulos (backend, frontend, base de datos).
3. **Pruebas del sistema:**
   * Evaluación completa del sistema para asegurar que todos los componentes funcionen juntos.
4. **Pruebas de aceptación:**
   * Validación con usuarios finales para garantizar que el sistema cumpla con las expectativas.
5. **Pruebas de regresión:**
   * Revisión de funcionalidades previas después de realizar correcciones o agregar nuevos módulos.

## 3.4 Requerimientos del test

#### 1. Ambiente de pruebas

Para garantizar la validez de las pruebas, se requiere un entorno que sea lo más similar posible al ambiente de producción. Este ambiente deberá incluir los siguientes elementos:

* **Servidor:**
  + Sistema operativo: Ubuntu Server 20.04 LTS.
  + Infraestructura: Procesador Intel Xeon o equivalente, 16 GB de RAM, almacenamiento SSD.
  + Base de datos: MySQL 8.0 o superior.
  + Backend: PHP 8.1 configurado con las extensiones necesarias para el sistema.
  + Seguridad: Certificados SSL/TLS válidos para conexiones HTTPS.
* **Cliente:**
  + Discos de prueba:
    - PC de escritorio: Windows 10/11, navegador Chrome/Firefox actualizado.
    - Móviles: Android 9 o superior, iOS 14 o superior.
  + Resoluciones de pantalla:
    - Escritorio: 1920x1080, 1366x768.
    - Móviles: 360x640, 414 x 896.
  + Redes de prueba: Simulación de conexión a internet con ancho de banda mínimo de 5 Mbps.

#### 2. Documentos requeridos

Para realizar pruebas efectivas, se necesitan documentos detallados que respalden los objetivos de las pruebas y las funcionalidades esperadas:

* **Requisitos Funcionales y No Funcionales:**
  + Documento detallado de los requerimientos priorizados.
  + Trazabilidad entre los casos de prueba y los requerimientos definidos.
* **Diseños y modelos:**
  + Diagramas de flujo de datos para entender las interacciones entre los módulos.
  + Prototipos de interfaz web para validar la experiencia del usuario.
  + Modelos de base de datos para validar estructuras y relaciones.
* **Manual de usuario:**
  + Instrucciones claras para los usuarios finales sobre el uso de los módulos.
  + Escenarios de prueba básicos detallados para referencia.
* **Especificaciones técnicas:**
  + Detalles del diseño del backend, APIs, endpoints y lógica de negocio.
  + Configuración del entorno de pruebas.

#### 3. Código

El equipo de pruebas debe tener acceso a:

* **Código fuente:**
  + Repositorio principal del sistema (por ejemplo, GitHub, GitLab).
  + Acceso a ramas estables para pruebas, incluyendo "main" y "release".
* **Scripts de configuración:**
  + Scripts para la configuración inicial del entorno, como creación de tablas en la base de datos o carga de datos iniciales.
* **Automatización de pruebas:**
  + Scripts de pruebas automatizadas para regresión y pruebas funcionales clave.
  + Herramientas como Selenium para pruebas web y JUnit para pruebas unitarias.

## 3.5 Clasificación de los defectos

| **Nivel de Severidad** | **Descripción** | **Ejemplo** |
| --- | --- | --- |
| Crítico | Falla en el sistema que impide continuar con el procesamiento. Este tipo de error bloquea la operación de la aplicación, y su resolución es de máxima prioridad para asegurar la disponibilidad del sistema. | **Ejemplo:** Se ha encontrado un error crítico que impide el inicio de sesión de los usuarios, bloqueando el acceso al sistema. |
| Grave | Impide la continuación del proceso de una función específica. Aunque no bloquea todo el sistema, afecta una funcionalidad clave necesaria para completar tareas importantes. | **Ejemplo:** El módulo de carga de recursos no funciona correctamente, impidiendo que se suban archivos de contenido para los cursos. |
| Medio | Restricciones en ciertas funciones; Sin embargo, el sistema puede continuar operando. Estos defectos impactan funcionalidades no críticas o secundarias. | **Ejemplo:** Algunos gráficos en el módulo de estadísticas no se muestran correctamente, pero las demás funciones del sistema están disponibles. |
| Leve | Cambios menores que no afectan el uso ni la funcionalidad del sistema. Su corrección es importante para la usabilidad y apariencia del sistema, pero no es urgente. | **Ejemplo:** Errores de ortografía en los mensajes de notificación o etiqueta |

## 3.6 Roles y responsabilidades

| **Rol** | **Responsabilidades** |
| --- | --- |
| Líder de Proyecto | * Definir los objetivos, requisitos y alcance del proyecto de testing en coordinación con el cliente. * Establecer los criterios de aceptación para el proyecto de pruebas y los criterios de finalización, junto con el cliente. * Acordar procedimientos en caso de aparición de defectos que impidan continuar las pruebas. * Planificar el proyecto, incluyendo la estrategia de testing y las técnicas a utilizar. * Supervisar la creación del entorno de pruebas con los técnicos y especialistas necesarios. * Identificar las funcionalidades con mayores riesgos y definir prioridades de testing. * Coordinar la ejecución de pruebas y el seguimiento de incidentes en colaboración con el cliente y los diseñadores. * Validar y aprobar los resultados de pruebas y los informes finales. |
| Diseñador de Pruebas | * Definir criterios para la ejecución de pruebas. * Diseñar casos de prueba detallados, cubriendo todas las funcionalidades y escenarios. * Colaborar con el cliente para generar los datos de prueba necesarios. * Revisar los resultados de las pruebas, analizando la información recopilada y ajustando casos si es necesario. * Elaborar informes de prueba, incluyendo observaciones y hallazgos técnicos. |
| Tester | * Ejecutar los casos de prueba, documentando los resultados de manera detallada. * Registrar incidentes y defectos, proporcionando detalles precisos para facilitar la solución. * Reportar la realización de pruebas y comunicar el estado del testing. * Seguir la trazabilidad de defectos hasta su resolución y cierre. |
| Responsable por el Cliente | * Definir junto con el líder de proyecto los criterios de aceptación de pruebas y los defectos críticos. * Participar en la validación de resultados y en la revisión de incidentes de alto impacto. * Proveer acceso a documentación, manuales y otros recursos necesarios. * Validar el informe final del proyecto de pruebas y los entregables asociados. |
| Contraparte técnica del cliente | * Colaborar en la definición de estrategias de generación de datos de prueba y validación de casos. * Apoyar en la creación del entorno de pruebas, garantizando la disponibilidad de recursos necesarios. * Verificar y validar resultados de pruebas, participando en la revisión de incidentes y corrección de errores. * Estar disponible para consultas y revisión de los resultados, asegurando la correcta implementación de los requisitos técnicos. |

## 3.7 Calendarización de las actividades de pruebas

| **Actividad** | **Semanas** |
| --- | --- |
|
| Análisis de requerimiento | Semana del 15 al 17 de octubre |
| Diseño de los casos de prueba | Semana del 18 al 20 de octubre |
| Reunión de avance y control | Semana del 21 de octubre |
| Adquirir recursos | Semana del 21 de octubre |
| Preparación ambiente de pruebas | Semana del 22 al 23 de octubre |
| **Ejecutar 1er ciclo de pruebas** | **Semana del 24 al 26 de octubre** |
| Corregir defectos | Semana del 27 al 29 de octubre |
| Reunión de avance y control | Semana del 30 de octubre |
| **Ejecutar 2do ciclo de pruebas** | **Semana del 1 al 4 de noviembre** |
| Corregir defectos | Semana del 5 al 7 de noviembre |
| Reunión de avance y control | Semana del 8 de noviembre |
| Pruebas de aceptación con usuario | Semana del 9 al 10 de noviembre |
| Preparación de Métricas y Evaluación | Semana del 11 al 13 de noviembre |
| Cierre proceso de pruebas | Semana del 14 al 15 de noviembre |

#### 

#### Leyenda de colores

* **Rojo** : Actividades de preparación y corrección.
* **Verde** : Reuniones de control y avance.
* **Negro** : Ciclos de pruebas.