

Plan de Pruebas de Software

LiquorLogix

Fecha: 04-12/2024

Tabla de contenido

Historial de Versiones	4
Información del Proyecto	4
Aprobaciones	4
Resumen Ejecutivo	5
Alcance de las Pruebas	6
Elementos de Pruebas	6
Nuevas Funcionalidades a Probar	6
Pruebas de Regresión	9
Funcionalidades a No Probar	11
Enfoque de Pruebas (Estrategia)	12
Criterios de Aceptación o Rechazo	15
Criterios de Aceptación o Rechazo	8
Criterios de Suspensión	17
Criterios de Reanudación	20
Entregables	9
Recursos	22
Requerimientos de Entornos – Hardware	10
Requerimientos de Entornos – Software	25
Herramientas de Pruebas Requeridas	26
Personal	28
Entrenamiento	28
Planificación y Organización	29

Procedimientos para las Pruebas	29
Matriz de Responsabilidades	31
Cronograma	32
Premisas	33
Dependencias y Riesgos	35
Referencias	37
Glosario	38

Historial de Versiones

Fecha	Versión	Autor	Organización	Descripción
04-12/2024	1.0	Cristopher Benavides	Servicio Nacional de Aprendizaje (SENA)	Plan de pruebas del proyecto LiquorLogix

Información del Proyecto

Empresa / Organización	SENA
Proyecto	LiquorLogix
Fecha de preparación	04-12/2024
Cliente	Los Primos
Patrocinador principal	-----
Gerente / Líder de Proyecto	Cristopher Stuart Benavides
Gerente / Líder de Pruebas de Software	Alejandro Ramirez Carvajal

Aprobaciones

Nombre y Apellido	Cargo	Departamento u Organización	Fecha	Firma
Albeiro Ramos	Instructor	Sena	Fecha revisión	

Resumen Ejecutivo

Este documento describe el Plan de Pruebas para el software de gestión de inventario desarrollado en PHP, HTML, CSS y JavaScript para la empresa *Los Primos*. El propósito del plan de pruebas es garantizar la calidad, funcionalidad y usabilidad del sistema, verificando que los módulos completos cumplan con los requisitos establecidos y que las vistas de los módulos incompletos sean coherentes con el diseño propuesto.

El presente plan es un **plan maestro**, ya que abarca la planificación general de las pruebas que se realizarán durante la validación del sistema, considerando tanto las funcionalidades terminadas como aquellas en desarrollo.

El alcance del plan de pruebas incluye los siguientes módulos:

- **Login:** Verificación de autenticación para usuarios
- **Usuarios:** Gestión de acceso
- **Categorías:** Registro, edición y consulta de categorías de productos.
- **Productos:** Manejo de inventario, incluyendo creación, modificación y eliminación de productos.

Además, se realizará una revisión de las vistas correspondientes a los módulos **Caja y Ventas**, asegurando que su diseño se alinee con las expectativas del cliente y permita una futura implementación sin complicaciones.

Restricciones:

- Recursos limitados: las pruebas serán ejecutadas manualmente por una persona.
- Presupuesto ajustado: no se contempla la integración de herramientas de pruebas automatizadas en esta etapa.

Alcance del esfuerzo de pruebas:

El esfuerzo de pruebas estará enfocado principalmente en los módulos completados, evaluando casos de prueba funcionales, de interfaz de usuario y de

experiencia del usuario. En cuanto a los módulos incompletos, se validará la estructura y navegabilidad de las vistas.

Este plan se ajustará a las restricciones y prioridades del proyecto, garantizando que los resultados obtenidos respalden la calidad y el éxito del sistema al momento de su entrega.

Alcance de las Pruebas

A continuación, se detallan los módulos y componentes que serán sometidos a pruebas en el software de gestión de inventario:

Módulos completos (funcionalidad implementada):

1. Módulo de Usuarios:

- a. Gestión de acceso
- b. Creación, edición y eliminación de usuarios.
- c. Búsqueda y listado de usuarios existentes.
- d. Validación de datos en formularios.

2. Módulo de Categorías:

- a. Registro de nuevas categorías.
- b. Edición y eliminación de categorías existentes.
- c. Listado de todas las categorías.
- d. Búsqueda de categorías específicas.

3. Módulo de Productos:

- a. Registro de productos con detalles (nombre, descripción, precios, stock, categoría).
- b. Modificación de datos de productos existentes.
- c. Listado general de productos y filtrado por categoría.
- d. Búsqueda avanzada de productos por nombre u otros criterios.

4. Módulo de Login:

- a. Autenticación de usuarios registrados.
- b. Validación de credenciales de acceso.
- c. Restricción de acceso para usuarios no autenticados.
- d. Manejo de mensajes de error y retroalimentación al usuario.

Módulos incompletos (solo vistas disponibles):

5. Módulo de Caja:

- a. Validación de diseño de la interfaz de usuario.
- b. Revisión de coherencia visual y cumplimiento de requisitos iniciales.

6. Módulo de Ventas:

- a. Evaluación del diseño de las vistas para registro y consulta de ventas.
- b. Comprobación de elementos de diseño como botones, formularios y tablas.

Componentes transversales:

- **Interfaz de Usuario (UI):**

Validación de diseño responsivo, coherencia visual, accesibilidad y facilidad de uso en todas las pantallas.

- **Base de Datos:**

Verificación de la consistencia y correcta interacción entre la aplicación y las tablas relacionadas a usuarios, categorías y productos.

- **Rutas y Navegación:**

Pruebas de acceso a las diferentes vistas y páginas asegurando que no existan enlaces rotos o inconsistencias en las rutas.

Nuevas Funcionalidades a Probar

Desde el punto de vista del usuario, se probarán las siguientes funcionalidades del sistema, enfocándose en garantizar que cumplen con los requisitos y brindan una experiencia adecuada:

Funcionalidades completas:

1. Gestión de Usuarios:

- a. Crear nuevos usuarios
- b. Editar información de usuarios existentes.
- c. Eliminar usuarios cuando ya no sean necesarios.
- d. Buscar usuarios en el sistema por nombre o ID.
- e. Control de acceso: Asegurar que solo usuarios con credenciales válidas puedan ingresar al sistema.

2. Gestión de Categorías:

- a. Registrar nuevas categorías de productos.
- b. Modificar o eliminar categorías existentes.
- c. Listar todas las categorías disponibles para facilitar la selección.
- d. Buscar categorías específicas usando palabras clave.

3. Gestión de Productos:

- a. Registrar productos con información detallada (nombre, descripción, precio de compra, precio de venta, stock, imagen, y categoría).
- b. Editar y actualizar información de productos en inventario.
- c. Eliminar productos que ya no se gestionen.
- d. Listar productos por categorías para facilitar su consulta.
- e. Buscar productos específicos por nombre o código.

4. Login de Usuarios:

- a. Permitir el inicio de sesión a usuarios registrados con credenciales válidas.
- b. Mostrar mensajes de error claros en caso de credenciales incorrectas.

Funcionalidades con vistas preliminares:

5. Gestión de Caja (Interfaz Preliminar):

- a. Navegar por las vistas relacionadas con la gestión de caja.
- b. Ver un diseño intuitivo y funcional que permita la posterior implementación de operaciones de caja.

6. Gestión de Ventas (Interfaz Preliminar):

- a. Explorar las vistas diseñadas para registrar y consultar ventas.
- b. Validar que los elementos de la interfaz sean claros y accesibles, aunque aún no se implementen las funcionalidades completas.

Funcionalidades transversales:

- **Diseño Responsivo:**
 - Validar que las pantallas se adapten correctamente a diferentes dispositivos (computadoras, tabletas, teléfonos móviles).
- **Interactividad General:**
 - Comprobar que los botones, formularios y enlaces sean funcionales y respondan adecuadamente a las acciones del usuario.

Estas pruebas garantizarán que tanto las funcionalidades completas como las vistas preliminares cumplan con las expectativas del usuario final, enfocándose en la usabilidad, claridad y accesibilidad del sistema.

Pruebas de Regresión

Las pruebas de regresión se enfocarán en garantizar que las funcionalidades no modificadas o no involucradas directamente en el desarrollo sigan funcionando correctamente tras la implementación de nuevos módulos o cambios. Estas funcionalidades se describen desde el punto de vista del usuario:

Funcionalidades a evaluar en las pruebas de regresión:

1. Autenticación y Control de Acceso:

- a. Validar que el sistema permita solo a los usuarios autorizados acceder
- b. Comprobar que las sesiones de usuario se gestionen de manera segura, sin interrupciones o errores inesperados.

2. Navegación entre Módulos:

- a. Asegurar que los menús y enlaces del sistema permitan al usuario desplazarse correctamente entre las diferentes secciones.
- b. Validar que las transiciones entre páginas no provoquen errores de carga o de diseño.

3. Interacción con el Inventario:

- a. Garantizar que las acciones relacionadas con productos, como listarlos, buscarlos o consultarlos, sigan funcionando sin errores tras los cambios en otros módulos.
- b. Confirmar que los filtros y criterios de búsqueda respondan de manera adecuada.

4. Gestión de Categorías:

- a. Verificar que las funcionalidades para agregar, editar, eliminar o consultar categorías operen correctamente después de los cambios en otros componentes del sistema.

5. Interfaz de Usuario:

- a. Validar que la apariencia general y el diseño de las vistas (como colores, botones y diseño responsivo) no hayan sido afectados por los desarrollos recientes.

6. Validación de Datos:

- a. Asegurar que las reglas de validación en formularios, como campos obligatorios y formatos específicos, continúen funcionando de manera correcta en los módulos ya implementados.

7. Seguridad del Sistema:

- a. Revisar que la aplicación no permita accesos no autorizados ni manipulación directa de datos mediante las URLs o formularios tras los cambios en otros módulos.

8. Rendimiento General:

- a. Confirmar que la velocidad de respuesta del sistema y el manejo de múltiples usuarios concurrentes sigan siendo aceptables después de las implementaciones recientes.

Estas pruebas garantizarán que la integración de los módulos de caja y ventas no impacte negativamente las funcionalidades ya implementadas, asegurando así la estabilidad del sistema completo.

Funcionalidades a No Probar

A continuación, se listan las funcionalidades que no se someterán a pruebas en esta etapa del proyecto, junto con las razones correspondientes y los riesgos asociados:

1. Módulo de Caja

- **Razón:** El módulo solo cuenta con las vistas desarrolladas; no hay lógica implementada que permita realizar pruebas funcionales.
- **Riesgos:**
 - Pueden existir problemas de integración cuando el módulo sea completado.
 - Riesgo de que las vistas no cumplan con los requisitos definidos por los usuarios al no validarse en esta etapa.

2. Módulo de Ventas

- **Razón:** Al igual que el módulo de caja, este módulo únicamente tiene desarrolladas las vistas y no se encuentra funcional en esta fase.
- **Riesgos:**
 - Posibles inconsistencias al integrarlo con otros módulos como inventario o caja.
 - Falta de alineación entre el diseño actual y los procesos comerciales reales.

3. Reportes Avanzados de Inventario

- **Razón:** Las funcionalidades avanzadas de generación de reportes (por ejemplo, estadísticas, gráficos, históricos) no forman parte del alcance definido en esta etapa del proyecto.
- **Riesgos:**

- La ausencia de pruebas en esta área puede llevar a retrasos en etapas futuras, si se decide implementar esta funcionalidad.

4. Seguridad Avanzada del Sistema

- **Razón:** El enfoque de seguridad en esta etapa se limita al control básico de acceso mediante el login. No se están probando características avanzadas como encriptación de datos, auditorías o mecanismos de prevención de ataques como inyecciones SQL.
- **Riesgos:**
 - Vulnerabilidades en el sistema que podrían ser explotadas si no se revisan adecuadamente en fases posteriores.

5. Integraciones con Terceros

- **Razón:** En esta etapa no se contempla la integración con APIs externas o sistemas de terceros, como plataformas de pago o software de contabilidad.
- **Riesgos:**
 - Fallos de integración o incompatibilidades futuras si no se planifican correctamente en el momento de su desarrollo.

Mitigación de Riesgos

Para minimizar los riesgos, se recomienda:

- Documentar las áreas excluidas para referencia en futuras fases.
- Priorizar pruebas de integración y seguridad tan pronto como se desarrollen las funcionalidades actualmente incompletas.

Enfoque de Pruebas (Estrategia)

El enfoque de pruebas para el proyecto **LiquorLogix** se centra en asegurar la funcionalidad y confiabilidad de los módulos desarrollados, garantizando que cumplan con los requisitos establecidos y que el sistema proporcione una experiencia de usuario fluida. La estrategia incluye los tipos de pruebas, configuraciones, y aspectos específicos de los módulos implementados.

1. Tipos de Pruebas a Realizar

- **Pruebas Funcionales:**

- Validar que los módulos desarrollados (Usuarios, Categorías, Productos y Login) cumplan con los requerimientos funcionales especificados.
 - Verificación de campos obligatorios y validaciones.
 - Pruebas de flujo en el sistema (creación, edición, búsqueda y eliminación de datos).

- **Pruebas de Interfaces:**

Asegurar que las interacciones entre las vistas y la lógica del sistema funcionen correctamente.

- Evaluación de formularios, botones, enlaces, menús desplegables y otros elementos interactivos.

- **Pruebas de Regresión:**

Garantizar que las nuevas implementaciones o modificaciones no afecten la funcionalidad existente de los módulos completos.

- **Pruebas de Seguridad:**

Probar los mecanismos básicos de acceso y autenticación, incluyendo:

- Login y manejo de sesiones.
- Protección contra accesos no autorizados.

- **Pruebas No Funcionales (Limitadas):**

Se realizarán pruebas básicas de rendimiento para identificar tiempos de respuesta en escenarios críticos, como la carga de listas de productos o usuarios.

2. Requerimientos Especiales de las Pruebas

- **Ambiente de Pruebas:**
 - Servidor local configurado en XAMPP (PHP, Apache, MySQL).
 - Navegadores: Google Chrome (versión reciente), Mozilla Firefox (versión reciente).
- **Datos de Pruebas:**

Subconjuntos de datos representativos que incluyen casos positivos y negativos:

- Productos con diferentes categorías y estados de inventario.
- Combinaciones de datos válidos e inválidos en los formularios.

3. Configuraciones a Probar

- **Resoluciones de Pantalla:**
 - Escritorio: 1920x1080, 1366x768.
 - Dispositivos móviles: 375x667 (iPhone), 360x640 (Android).
- **Configuraciones del Navegador:**
 - Cookies y JavaScript habilitados.
 - Pruebas de compatibilidad con extensiones comunes (adblockers, etc.).

4. Nivel de Pruebas de Regresión

- El alcance de las pruebas de regresión será limitado a los módulos completos (Usuarios, Categorías, Productos y Login).
- Se priorizarán los flujos principales:
 - Inicio de sesión y navegación.
 - Creación, edición y eliminación de usuarios, categorías y productos.

5. Consideraciones

- Las pruebas en los módulos de **Caja y Ventas** se posponen hasta que se implemente la lógica funcional correspondiente.
- Se documentarán todos los hallazgos para su resolución en etapas posteriores.

Esta estrategia busca maximizar la calidad del sistema en esta fase, con foco en los módulos terminados y los elementos críticos del sistema.

Criterios de Aceptación o Rechazo

Criterios de Aceptación o Rechazo

Los criterios de aceptación o rechazo definen los requisitos mínimos que deben cumplirse para considerar que las pruebas realizadas han sido exitosas y el sistema está listo para su implementación. A continuación se detallan los criterios aplicables al Plan de Pruebas para el proyecto **LiquorLogix**:

1. Criterios de Aceptación

- **Cobertura de Funcionalidades Principales:**
 - Se debe completar el 100% de las pruebas funcionales en los módulos completos (Usuarios, Categorías, Productos y Login). Esto incluye la validación de la creación, edición, eliminación y búsqueda de registros, así como la verificación de validaciones en los formularios.
- **Casos de Éxito:**

Al menos el **95% de los casos de prueba** deben ser exitosos (es decir, deben pasar sin fallos).

- **Validación de Flujo de Usuario:**

El sistema debe permitir a los usuarios realizar todas las operaciones básicas sin errores, como ingresar, editar, buscar y eliminar usuarios, productos y categorías.

- **Rendimiento Básico:**

El sistema debe ser capaz de manejar la carga normal de trabajo, sin tiempos de respuesta superiores a 2 segundos para las operaciones más comunes (búsqueda de productos, carga de listas de usuarios).

- **Pruebas de Seguridad Aprobadas:**

El sistema debe tener implementadas las pruebas básicas de seguridad y validación de accesos, garantizando que no se puedan eludir los mecanismos de autenticación y autorización.

- **Documentación Completa:**

Todos los hallazgos de pruebas deben estar documentados, incluyendo los casos de prueba, defectos encontrados y las acciones tomadas para resolverlos.

- **Defectos Críticos Resueltos:**

El sistema no debe tener defectos críticos no resueltos. Los defectos críticos incluyen problemas que afectan la funcionalidad principal del sistema, como la imposibilidad de iniciar sesión o realizar operaciones básicas de inventario.

2. Criterios de Rechazo

- **Cobertura Insuficiente:**

Si las pruebas no cubren al menos el **90% de las funcionalidades completas** de los módulos (Usuarios, Categorías, Productos y Login), el plan de pruebas será rechazado.

- **Defectos Críticos No Resueltos:**

Si existen defectos críticos sin corregir al final de las pruebas, el sistema no será aceptado para su implementación. Los defectos críticos incluyen:

- Fallos en el inicio de sesión o gestión de usuarios.
- Errores graves en la creación, edición o eliminación de productos o categorías.

- **Falta de Seguridad Básica:**

Si no se validan correctamente las pruebas de autenticación y control de accesos, el sistema será rechazado debido a riesgos de seguridad.

- **Desempeño Insuficiente:**

Si el tiempo de respuesta de las operaciones más comunes excede los 2 segundos bajo condiciones de carga normal, el sistema no podrá ser aceptado hasta mejorar el rendimiento.

- **Problemas de Compatibilidad No Resueltos:**

Si se identifican fallos importantes de compatibilidad entre navegadores o dispositivos y no se resuelven, el sistema será rechazado hasta que se garantice su funcionamiento adecuado en los principales entornos de usuario.

Estos criterios aseguran que las pruebas sean exhaustivas y que el sistema cumpla con los requisitos funcionales y de calidad antes de su lanzamiento.

Criterios de Suspensión

Los **Criterios de Suspensión** definen las condiciones bajo las cuales las pruebas deben ser detenidas, ya sea temporal o permanentemente, hasta que se resuelvan ciertos problemas que impidan su continuación o validación. Estos criterios aseguran que no se sigan ejecutando pruebas en un sistema que

presenta fallos críticos que afectan la validez o viabilidad de las pruebas. A continuación se detallan los criterios aplicables para el proyecto **LiquorLogix**:

1. Defectos Críticos No Resueltos

- **Condición de Suspensión:** Si se identifican defectos críticos que impiden la realización de más pruebas o que afectan funcionalidades esenciales del sistema, como:
 - El sistema no permite a los usuarios iniciar sesión o acceder a su cuenta.
 - Imposibilidad de realizar operaciones básicas de gestión de productos, categorías o usuarios.
 - Fallos en las funciones de alta, modificación o eliminación de registros.
- **Acción:** Las pruebas deberán ser suspendidas hasta que los defectos sean corregidos y verificadas las correcciones.

2. Fallos en el Sistema de Autenticación o Control de Accesos

- **Condición de Suspensión:** Si el sistema presenta fallos graves en la autenticación de usuarios, control de accesos o permisos de usuario, de manera que no se puede garantizar que los usuarios no autorizados accedan a las áreas restringidas del sistema.
- **Acción:** Las pruebas de los módulos relacionados con seguridad deben ser suspendidas hasta que se resuelvan los problemas de control de acceso.

3. Inestabilidad del Sistema

- **Condición de Suspensión:** Si el sistema se vuelve inestable durante la ejecución de las pruebas, por ejemplo:
 - Caídas frecuentes del servidor.
 - Errores constantes que impiden la ejecución de más pruebas.
- **Acción:** Las pruebas deben suspenderse hasta que se solucione la causa de la inestabilidad.

4. Porcentaje de Casos Fallidos Excesivo

- **Condición de Suspensión:** Si más del **30% de los casos de prueba** ejecutados hasta el momento fallan, especialmente en funcionalidades clave, como la gestión de inventarios, usuarios y la autenticación.
- **Acción:** Las pruebas deberán ser suspendidas y los errores críticos deberán ser resueltos antes de continuar.

5. Defectos de Desempeño Críticos

- **Condición de Suspensión:** Si el rendimiento del sistema no cumple con los criterios mínimos establecidos, por ejemplo:
 - El tiempo de respuesta de las operaciones básicas supera los 5 segundos.
 - La aplicación presenta tiempos de carga excesivos que afectan la experiencia del usuario.
- **Acción:** Las pruebas de desempeño deben suspenderse hasta que se optimicen los problemas de rendimiento.

6. Fallos en la Compatibilidad de Navegadores o Dispositivos

- **Condición de Suspensión:** Si se identifican fallos graves de compatibilidad entre navegadores o dispositivos (por ejemplo, incompatibilidad con los principales navegadores como Chrome, Firefox, Safari, o con dispositivos móviles) y no se pueden ejecutar las pruebas en los entornos más críticos.
- **Acción:** Las pruebas en los navegadores o dispositivos afectados deben suspenderse hasta que se corrijan los problemas de compatibilidad.

7. Datos de Prueba Incompletos o Incorrectos

- **Condición de Suspensión:** Si se detectan errores en los datos utilizados para las pruebas (por ejemplo, datos de prueba incorrectos o incompletos que afecten la validez de los casos de prueba).
- **Acción:** Las pruebas deben suspenderse hasta que se rectifiquen los datos y se vuelvan a cargar correctamente.

Estos **Criterios de Suspensión** garantizan que las pruebas se realicen bajo condiciones controladas y con datos y entornos válidos, evitando continuar con la ejecución en un sistema que podría generar resultados incorrectos o que aún no ha sido estabilizado adecuadamente.

Criterios de Reanudación

Los Criterios de Reanudación definen las condiciones que deben cumplirse para reanudar las pruebas después de haber sido suspendidas, asegurando que los problemas que causaron la suspensión hayan sido resueltos adecuadamente y que el sistema esté nuevamente en condiciones de ser probado. A continuación, se detallan los criterios aplicables para el proyecto LiquorLogix:

1. Corrección de Defectos Críticos

- **Condición de Reanudación:** Las pruebas se reanudarán una vez que los defectos críticos que causaron la suspensión hayan sido corregidos. Estos incluyen:
 - La restauración de las funciones esenciales del sistema, como la autenticación de usuarios y la gestión de inventarios.
 - La resolución de los errores que impedían el acceso a las áreas del sistema.
- **Acción:** Verificar que los defectos hayan sido corregidos a través de pruebas de regresión, y luego reanudar las pruebas pendientes.

2. Validación de la Estabilidad del Sistema

- **Condición de Reanudación:** Las pruebas se reanudarán cuando se confirme que el sistema es estable y no presenta caídas o errores que interrumpan el flujo de trabajo.
- **Acción:** Realizar pruebas de estabilidad básica para asegurarse de que el sistema funcione de manera continua y sin interrupciones, y luego reanudar las pruebas normales.

3. Resolución de Problemas de Desempeño

- **Condición de Reanudación:** Si las pruebas fueron suspendidas debido a problemas de rendimiento (tiempos de respuesta excesivos, lentitud en el sistema), se reanudarán una vez que se optimicen los tiempos de respuesta y se mejoren los parámetros de desempeño según los estándares establecidos.
- **Acción:** Realizar pruebas de rendimiento para asegurar que el sistema cumpla con los tiempos de respuesta definidos, y después reanudar las pruebas de funcionalidad.

4. Corregir Fallos en la Compatibilidad de Navegadores y Dispositivos

- **Condición de Reanudación:** Si la suspensión se debió a problemas de compatibilidad entre navegadores o dispositivos, las pruebas se reanudarán después de corregir las incompatibilidades y garantizar que el sistema funcione correctamente en los principales navegadores y dispositivos.
- **Acción:** Verificar la corrección de los problemas de compatibilidad a través de pruebas de validación cruzada en diferentes entornos de navegador y dispositivo, y luego continuar con las pruebas.

5. Validación de Datos de Prueba

- **Condición de Reanudación:** Si la suspensión se debió a datos incorrectos o incompletos, las pruebas se reanudarán una vez que los datos de prueba sean corregidos y verificados.
- **Acción:** Verificar que los datos estén completos y sean válidos para las pruebas, luego reanudar las pruebas en las funcionalidades afectadas.

6. Validación de Corrección de Casos de Prueba Fallidos

- **Condición de Reanudación:** Las pruebas se reanudarán cuando el porcentaje de casos de prueba fallidos sea menor o igual al 5%, asegurando que los defectos que causaron fallos en las pruebas hayan sido resueltos satisfactoriamente.

- **Acción:** Ejecución de los casos de prueba que previamente fallaron para verificar que se han solucionado los problemas, y luego reanudar la ejecución de las pruebas.

7. Aprobación de Cambios en la Funcionalidad

- **Condición de Reanudación:** Si la suspensión se debió a la implementación de nuevos cambios o funcionalidades, las pruebas se reanudarán después de que los cambios sean validados y aprobados por los responsables del proyecto (equipo de desarrollo y stakeholders).
- **Acción:** Realizar pruebas de validación de los nuevos cambios para asegurarse de que no introduzcan nuevos problemas, y luego continuar con el plan de pruebas.

Los **Criterios de Reanudación** aseguran que las pruebas se retomen solo cuando se haya abordado adecuadamente la causa de la suspensión, garantizando la integridad y efectividad de las pruebas futuras. Esto también ayuda a evitar que se reanuden las pruebas en un sistema inestable o incompleto.

Entregables

https://github.com/Cristopher1178Col/Proyecto_LiquorLogix.git

Recursos

Requerimientos de Entornos – Hardware

Para ejecutar el **Plan de Pruebas de Software** del proyecto **LiquorLogix**, es necesario disponer de los siguientes recursos de hardware y entornos para asegurar que las pruebas se realicen en condiciones óptimas. Estos incluyen servidores, bases de datos, equipos de prueba y accesos a la red. A continuación, se detallan los requerimientos específicos:

1. Servidores de Aplicación

- **Servidor de Aplicación Principal:**
 - **Especificaciones mínimas:**
 - Procesador: Intel i3 o equivalente (mínimo 2.5 GHz)
 - Memoria RAM: 8 GB
 - Almacenamiento: 200 GB de disco duro SSD o HDD
 - Sistema Operativo: Windows Server 2019 o Linux (Ubuntu 20.04 o CentOS)
 - Conectividad: Red Gigabit Ethernet o Wi-Fi 5 (mínimo 100 Mbps)
 - **Función:** Alojamiento de la aplicación web **LiquorLogix** (PHP, HTML, CSS, JS) y sus servicios backend (servidores Apache, MySQL).
- **Servidor de Bases de Datos:**
 - **Especificaciones mínimas:**
 - Procesador: Intel i3 o equivalente (mínimo 2.5 GHz)
 - Memoria RAM: 8 GB
 - Almacenamiento: 200 GB de disco duro SSD
 - Sistema Operativo: Windows Server 2019 o Linux (Ubuntu 20.04 o CentOS)
 - Software de Base de Datos: MySQL 8.0 o superior
 - Conectividad: Red Gigabit Ethernet
 - **Función:** Asegurar que el sistema de base de datos para **LiquorLogix** (MySQL) funcione de manera eficiente durante las pruebas.

2. Equipos de PC para los Testers

- **Equipos de Prueba:**
 - **Especificaciones mínimas:**
 - Procesador: Intel Core i3 o equivalente
 - Memoria RAM: 4 GB o superior

- Almacenamiento: 128 GB SSD
- Sistema Operativo: Windows 10 o superior, Ubuntu 20.04
- Navegadores: Google Chrome, Mozilla Firefox, Microsoft Edge, Safari (para pruebas de compatibilidad en diferentes entornos)
- Conectividad: Red Wi-Fi o Ethernet (mínimo 100 Mbps)
- **Función:** Los equipos de prueba se usarán para ejecutar los casos de prueba funcionales y no funcionales, interactuar con la interfaz de usuario de **LiquorLogix** y realizar pruebas de compatibilidad entre navegadores.

3. Conectividad a la Red

- **Requerimientos de Conexión de Red:**
 - **Conexión a Internet:** Acceso a Internet de alta velocidad (mínimo 50 Mbps) para obtener acceso remoto a servidores de prueba, bases de datos y otros recursos de red necesarios para las pruebas.
 - **Red Interna:** Conexión de red de alta velocidad (mínimo 1 Gbps) para permitir la comunicación eficiente entre servidores de pruebas, estaciones de trabajo y otros dispositivos involucrados en el proceso de prueba.

4. Software Requerido

- **Sistemas Operativos:**
 - Windows Server 2019 o superior, Ubuntu 20.04 o CentOS (para servidores)
 - Windows 10 o superior, macOS, Ubuntu 20.04 (para estaciones de trabajo)
- **Software de Desarrollo y Pruebas:**
 - **PHP 8.0 o superior:** Para la ejecución del código backend.
 - **MySQL 8.0 o superior:** Para el manejo de la base de datos.
 - **Apache:** Para el servidor web.
 - **Herramientas de Pruebas:**
 - OBS Studio: Para ejecución de pruebas manuales
- **Entorno de Desarrollo y Gestión de Versiones:**
 - **Git:** Para el control de versiones del código.
 - **GitHub o GitLab:** Para el almacenamiento del código fuente y la colaboración.
 - **IDE recomendados:** Visual Studio Code

5. Backup y Recuperación de Datos

- **Requerimientos de Backup:**
 - Procedimiento de respaldo diario de bases de datos y archivos importantes.
- **Requerimientos de Recuperación:**
 - Procedimiento para restaurar datos y configuraciones del entorno de prueba en caso de falla o pérdida de datos.

Resumen de los Requerimientos de Hardware:

Recurso	Especificación	Función
Servidor de Aplicación	Intel i3, 8GB RAM, 200GB SSD, Windows Server / Linux	Alojamiento de la aplicación web y servicios backend
Servidor de Base de Datos	Intel i3, 8GB RAM, 200GB SSD, MySQL 8.0, Windows Server	Alojamiento de la base de datos de la aplicación
PC de Testers	Intel i3, 4GB RAM, SSD de 128GB, Windows 10 / Ubuntu 20.04	Ejecución de pruebas funcionales y de compatibilidad
Conexión a Internet	50 Mbps mínimo	Acceso remoto a servidores y bases de datos
Dispositivos Móviles	Android 9.0 o superior, iOS 12.0 o superior	Pruebas de compatibilidad móvil (opcional)

Estos recursos son esenciales para garantizar la correcta ejecución de todas las actividades descritas en el **Plan de Pruebas de Software**, asegurando que los entornos y herramientas necesarias estén disponibles para llevar a cabo pruebas funcionales, de rendimiento y de compatibilidad.

Requerimientos de Entornos – Software

Requerimientos de Entornos – Software

Para llevar a cabo las actividades de prueba del proyecto **LiquorLogix**, es necesario contar con un conjunto de herramientas y configuraciones de software que faciliten las pruebas funcionales, de rendimiento, de integración y de compatibilidad. A continuación, se detalla el software necesario para el entorno de pruebas:

1. Software para el Entorno de Pruebas

- **Sistema Operativo de Servidores:**
 - **Windows Server 2019 o superior o Linux (Ubuntu 20.04, CentOS 8):**
Para los servidores de aplicación y base de datos.
 - **Requerimientos:** El sistema debe ser estable y compatible con las versiones de PHP, MySQL y Apache/Nginx necesarios para ejecutar el software.
- **Sistema Operativo de Estaciones de Trabajo (Testers):**
 - **Windows 10 o superior, macOS, o Ubuntu 20.04:** Debe ser compatible con las herramientas de prueba, navegadores web y editores de código.

Herramientas de Pruebas Requeridas

1. Software para el Entorno de Pruebas

- **Sistema Operativo de Servidores:**
 - **Windows Server 2019 o superior o Linux (Ubuntu 20.04, CentOS 8):**
Para los servidores de aplicación y base de datos.
 - **Requerimientos:** El sistema debe ser estable y compatible con las versiones de PHP, MySQL y Apache/Nginx necesarios para ejecutar el software.
- **Sistema Operativo de Estaciones de Trabajo (Testers):**
 - **Windows 10 o superior, macOS, o Ubuntu 20.04:** Debe ser compatible con las herramientas de prueba, navegadores web y editores de código.

2. Software de Desarrollo y Ejecución de la Aplicación

- **PHP (8.0 o superior):** Esencial para la ejecución del código backend de la aplicación **LiquorLogix**.
 - **Requerimientos:** PHP debe estar correctamente configurado en el servidor de pruebas y las estaciones de trabajo de los testers.
- **MySQL (8.0 o superior):** Base de datos relacional que almacena la información de los usuarios, productos, categorías, ventas, entre otros.
 - **Requerimientos:** Debe estar instalado y configurado en el servidor de base de datos. También se debe proporcionar acceso adecuado para la ejecución de consultas de prueba.
- **Apache:** Servidores web para alojar la aplicación en el entorno de pruebas.
 - **Requerimientos:** Apache es la opción preferida para esta aplicación debido a su compatibilidad con PHP. Debe estar configurado adecuadamente para servir la aplicación web.
- **JavaScript:** Para manejar la interacción en el frontend de la aplicación.
 - **Requerimientos:** Asegurar que las bibliotecas JavaScript necesarias estén instaladas y sean compatibles con la versión de PHP y el entorno de desarrollo.
- **Bulma o cualquier otro framework CSS:** Para la implementación del diseño responsivo de la aplicación.
 - **Requerimientos:** Bulma debe ser instalado y configurado correctamente para asegurar la coherencia en las pruebas de diseño y usabilidad.

3. Software de Gestión de Versiones y Colaboración

- **Git:** Herramienta de control de versiones para manejar el código fuente de la aplicación.
 - **Requerimientos:** Los testers deben tener acceso a un repositorio Git GitHub, donde puedan acceder a las versiones más recientes del código y realizar pruebas en el código fuente actualizado.
- **GitHub:** Plataforma para almacenar y colaborar en el código fuente.
 - **Requerimientos:** Debe estar configurada para facilitar la colaboración entre los desarrolladores y los testers, permitiendo el acceso a los repositorios donde se almacenan los casos de prueba y el código.

4. Accesos a Sistemas y Bases de Datos

- **Acceso a la Base de Datos:** Los testers deben tener acceso a la base de datos MySQL para realizar consultas directas, verificar datos y ejecutar pruebas de validación.
 - **Requerimientos:** Accesos configurados para asegurar que los testers puedan verificar la consistencia de los datos en las pruebas, además de realizar pruebas de integridad en las relaciones entre las tablas.
- **Acceso a los Entornos de Pruebas:**
 - **Acceso a las URLs de Prueba:** Los testers deben tener acceso a las versiones de prueba de la aplicación **LiquorLogix** en el servidor de pruebas.

Personal

Para la realización de las actividades de pruebas del proyecto, se contará con una sola persona encargada, que será la siguiente:

- **Jaime Carvajal:** Encargado de realizar todas las pruebas del sistema, incluyendo pruebas funcionales, pruebas de regresión, pruebas de rendimiento y validación de funcionalidades.

Jaime Carvajal se encargará de todas las actividades relacionadas con el testeo del software, desde la creación y ejecución de los casos de prueba hasta la documentación de resultados y la gestión de defectos encontrados.

Entrenamiento

Para llevar a cabo las pruebas del sistema de manera efectiva, se ha identificado la necesidad de entrenamiento en las siguientes áreas:

1. **Entrenamiento en el sistema de gestión de inventario (LiquorLogix):**
 - a. **Objetivo:** Asegurar que Jaime Carvajal comprenda a fondo cómo funciona el software, sus módulos y las interacciones entre los distintos componentes (Usuarios, Categorías, Productos, Login, Caja y Ventas).
 - b. **Contenido:**
 - i. Navegación y uso de los módulos completos (Usuarios, Categorías, Productos y Login).
 - ii. Interacción con las vistas incompletas de los módulos de Caja y Ventas.
 - iii. Identificación de flujos y procesos clave.
 - iv. Validación de funcionalidades y características del sistema.
2. **Entrenamiento en herramientas de prueba:**
 - a. **Objetivo:** Familiarizar a Jaime Carvajal con las herramientas y enfoques necesarios para realizar las pruebas de manera eficiente.
 - b. **Contenido:**
 - i. Uso de herramientas de seguimiento de errores (si aplica).
 - ii. Redacción y gestión de casos de prueba.
 - iii. Herramientas de pruebas funcionales y de regresión, así como pruebas manuales.

Duración estimada: Dependiendo de la familiaridad de Jaime con el sistema y las herramientas, se estiman aproximadamente 1 a 2 días para una capacitación inicial y seguimiento en base a las necesidades específicas.

Planificación y Organización

Procedimientos para las Pruebas

Los procedimientos para la ejecución del plan de pruebas del sistema LiquorLogix están diseñados para garantizar la calidad y funcionalidad del software. A continuación, se detallan las etapas y metodologías a seguir:

1. Preparación

- **Revisión de requisitos:** Analizar los documentos de especificación funcional y técnica para entender los módulos y sus interacciones.
- **Configuración del entorno:** Verificar que el entorno de pruebas esté completamente configurado, incluyendo hardware, software y bases de datos necesarias.
- **Revisión de casos de prueba:** Validar y ajustar los casos de prueba para asegurar que cubran todos los requisitos funcionales.

2. Ejecución de Pruebas

- **Pruebas funcionales:**
 - Ejecutar los casos de prueba diseñados para verificar que cada módulo (Usuarios, Categorías, Productos y Login) cumpla con sus especificaciones.
 - Probar las vistas de los módulos incompletos (Caja y Ventas) para garantizar su correcto despliegue.
- **Pruebas de integración:**
 - Verificar la interacción entre los diferentes módulos, como el impacto de las operaciones en Productos sobre el módulo de Categorías.
- **Pruebas de regresión:**
 - Ejecutar pruebas en funcionalidades previamente desarrolladas para confirmar que no se han visto afectadas por los cambios recientes.

3. Gestión de Defectos

- Registrar los errores detectados en un documento o herramienta de seguimiento.
- Clasificar los errores según su prioridad y severidad.
- Comunicar los defectos al equipo de desarrollo para su resolución.

4. Validación

- Confirmar que todos los errores críticos han sido corregidos.
- Re-ejecutar los casos de prueba afectados por los defectos reportados.

5. Documentación

- Registrar los resultados de cada prueba, incluyendo casos exitosos y fallidos.
- Generar un informe de pruebas que resuma los hallazgos, defectos y métricas clave.

6. Criterios de Finalización

- Completar el 100% de los casos de prueba diseñados.
- Garantizar que todos los errores críticos y de alta prioridad hayan sido corregidos.
- Confirmar que el sistema cumple con los requisitos funcionales y de calidad definidos.

Metodología

El enfoque será **manual**, dado que las pruebas serán realizadas por una única persona, Jaime Carvajal. Este método permitirá una revisión detallada y directa del comportamiento del sistema, enfocándose en identificar errores tanto funcionales como de presentación.

Matriz de Responsabilidades

Dado que las pruebas serán realizadas por un único responsable, Jaime Carvajal, se adaptará la matriz de responsabilidades a su rol principal, cubriendo tanto la ejecución como la supervisión de las pruebas.

Actividad	Responsable (R)	Aprobador (A)	Consultado (C)	Informado (I)
Revisión de requisitos funcionales	Jaime Carvajal	-	-	-
Diseño de casos de prueba	Jaime Carvajal	-	-	-
Configuración del entorno de pruebas	Jaime Carvajal	-	-	-

Ejecución de pruebas funcionales	Jaime Carvajal	-	-	-
Ejecución de pruebas de integración	Jaime Carvajal	-	-	-
Ejecución de pruebas de regresión	Jaime Carvajal	-	-	-
Registro de defectos	Jaime Carvajal	-	-	-
Seguimiento a la corrección de errores	Jaime Carvajal	-	-	-
Documentación de resultados de pruebas	Jaime Carvajal	-	-	-
Generación del informe final de pruebas	Jaime Carvajal	-	-	-

Notas:

- **Responsable (R):** Jaime Carvajal será el encargado directo de realizar cada actividad relacionada con las pruebas del software.
- **Aprobador (A):** No aplica, ya que el equipo está compuesto únicamente por una persona.
- **Consultado (C):** No aplica en esta etapa.
- **Informado (I):** No aplica debido a la estructura del equipo.

Esta matriz refleja la centralización de responsabilidades en un único tester, optimizando la gestión del proceso de pruebas en un proyecto de alcance reducido.

Cronograma

LIQUOR LOGIX

Cronograma de Actividades

Fecha Inicio Proyecto

8/10/2024 (martes)

Líder del Proyecto

CRISTOPHER STUART BENAVIDES RAMIREZ

Semana 1

Excel para la web no admite la

Semana 1

Semana 2

Semana 3

Semana 4

Semana 5

Semana 6

Semana 7

Semana 8

7 oct 2024

14 oct 2024

21 oct 2024

28 oct 2024

4 nov 2024

11 nov 2024

18 nov 2024

25 nov 2024

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

1

ACTIVIDAD

COMPETE

EVIDENCIA

RESPONSABLE

INICIO

FIN

1 FASE: ANÁLISIS

JavaScript (Angular, Vue y React)

II

22051096

3

Diagrama de distribución y/o despliegue

Luis Olajia

vi. 01/11/24

vi. 29/11/24

0

100%

21

3.9

Proponer la arquitectura de software tomando en cuenta el Modelo - Vista - Controlador (MVC) y stack de tecnologías

II

22051096

3

Diagrama de Casos de la Solución Software (Herramientas CASE - Extender UML)

Luis Olajia

-

0

0%

-

3.7

Diseñar el Diagrama de Casos de la Solución Software (Herramientas CASE - Extender UML)

II

22051096

3

Diagrama de Clases

Luis Olajia

-

0

0%

-

3.8

Desarrollar la Solución Software con conexión a la Base de Datos

II

22051096

3

Backend de la Solución Software

Christopher Benavides

vi. 04/10/24

vi. 04/10/24

0

100%

1

3.9

Implementar el Frontend a la Solución Software

II

22051096

3

Backend y Frontend de la Solución Software integrados

Christopher Benavides

vi. 04/10/24

vi. 04/10/24

0

100%

1

3.10

Desarrollar las APIs RESTful para consumir servicios

II

22051096

3

APIs RESTful de la Solución Software

Christopher Benavides

vi. 04/10/24

vi. 04/10/24

0

0%

1

4 FASE: EVALUACIÓN

4.1

Diseñar el Plan Maestro de Pruebas de la solución software

III

22051096

3

Plan Maestro de Pruebas

Christopher Benavides

vi. 04/10/24

vi. 04/10/24

0

100%

1

4.2

Diseñar los Casos de Prueba de la solución software

III

22051096

3

Casos de Prueba

Johan Sepúlveda

vi. 04/10/24

ma. 08/10/24

0

20%

1

4.3

Diseñar las pruebas unitarias de la solución software

III

22051096

3

Pruebas Unitarias

Luis Olajia

-

0

0%

-

4.4

Diseñar las pruebas de integración de la solución software

III

22051096

3

Pruebas de Integración

Luis Olajia

-

0

0%

-

4.5

Ejecutar y documentar los casos de prueba, pruebas unitarias y pruebas de integración

III

22051096

3

Documentación de las Pruebas

Jaime Canajali

-

0

0%

-

4.6

Diseñar el Manual Técnico de la solución software

III

22051096

3

Manual Técnico

Johan Sepúlveda

lu. 02/12/24

-

0

30%

-

4.7

Diseñar el Manual de Usuario de la solución software

III

22051096

3

Manual de Usuario

Johan Sepúlveda

lu. 02/12/24

ju. 05/12/24

0

100%

-

Premisas

- Disponibilidad de recursos humanos:**

El proceso de pruebas será realizado exclusivamente por Jaime Carvajal, quien tendrá disponibilidad parcial de tiempo (horario específico de trabajo asignado al proyecto).

- Limitaciones de tiempo:**

El ciclo completo de pruebas debe realizarse en un plazo de **3 semanas**, siguiendo el cronograma establecido para la finalización del proyecto.

- Disponibilidad de recursos técnicos:**

- Se utilizarán los siguientes equipos y herramientas:
 - Computador personal con conexión estable a internet.
 - Servidor local configurado con **XAMPP** para pruebas en entorno controlado.
 - Herramientas como **PHPMYAdmin** para gestionar la base de datos y **navegadores web** modernos para la ejecución de pruebas de interfaz.

- Entorno de pruebas:**

- Las pruebas se realizarán en un entorno de desarrollo local replicado del entorno de producción.

- Datos de prueba:**

- Se empleará un conjunto de datos predefinido para evaluar las funcionalidades, simulando escenarios reales de uso.
- **Metodología de pruebas:**
 - Se aplicará una metodología **manual** para ejecutar pruebas funcionales, de integración y regresión.
 - Se documentarán los resultados en hojas de cálculo para análisis y generación de reportes finales.
- **Cobertura de pruebas:**
 - El enfoque se centrará en las funcionalidades completas del sistema: módulos de **usuarios, categorías, productos y login**.
 - Los módulos incompletos de **caja y ventas** solo serán evaluados en su diseño de interfaz y navegación básica.
- **Limitaciones de herramientas:**

No se emplearán herramientas automatizadas para pruebas debido a restricciones de tiempo y presupuesto.
- **Interacción con desarrolladores:**

Jaime Carvajal se coordinará directamente con el desarrollador para reportar y corregir defectos en tiempo real.
- **Entrega de resultados:**

Se presentará un informe final con los hallazgos, incluyendo casos exitosos, defectos identificados y recomendaciones.

Dependencias y Riesgos

A continuación, se detallan las dependencias, riesgos asociados al proceso de pruebas y los planes de mitigación o contingencia propuestos:

Dependencias

1. Dependencia con el equipo de desarrollo:

- a. Las pruebas dependen de la entrega oportuna de los módulos completos e integrados para iniciar la validación.
- b. **Riesgo asociado:** Retrasos en la entrega de los módulos listos para pruebas.
- c. **Plan de mitigación:** Mantener comunicación constante con el desarrollador para establecer fechas claras de entregas parciales y finales.

2. Dependencia del entorno local:

- a. Las pruebas requieren que el entorno de desarrollo local funcione correctamente con **XAMPP** configurado.
- b. **Riesgo asociado:** Fallos en la configuración del servidor o incompatibilidades en el entorno.
- c. **Plan de mitigación:** Realizar pruebas previas para garantizar que el entorno esté operativo antes de iniciar las validaciones.

3. Dependencia de datos de prueba:

- a. Uso de datos de prueba relevantes y adecuados para simular escenarios reales.
- b. **Riesgo asociado:** Insuficiencia o inexactitud en los datos proporcionados.
- c. **Plan de mitigación:** Generar un conjunto de datos detallados que cubran todos los casos previstos antes de iniciar las pruebas.

Riesgos

1. Disponibilidad limitada de recursos humanos:

- a. Jaime Carvajal es el único responsable del proceso de pruebas.
- b. **Impacto:** Medio.
- c. **Probabilidad:** Media.

- d. **Plan de contingencia:** Priorizar casos críticos y pruebas de regresión esenciales para cumplir con los plazos.
- 2. **Restricciones de tiempo:**
 - a. Plazo corto para realizar todas las pruebas.
 - b. **Impacto:** Alto.
 - c. **Probabilidad:** Alta.
 - d. **Plan de mitigación:** Optimizar la estrategia de pruebas, enfocándose en áreas críticas del software (módulos completos) y documentando posibles pruebas pendientes.
- 3. **Cambios inesperados en los requerimientos:**
 - a. Ajustes de última hora en los módulos o funcionalidades.
 - b. **Impacto:** Alto.
 - c. **Probabilidad:** Media.
 - d. **Plan de contingencia:** Actualizar el plan de pruebas y ajustar el cronograma si es necesario.
- 4. **Fallos en la conexión a la base de datos o servicios externos:**
 - a. Incompatibilidad con el diseño de la base de datos o interrupciones durante las pruebas.
 - b. **Impacto:** Alto.
 - c. **Probabilidad:** Baja.
 - d. **Plan de mitigación:** Realizar verificaciones iniciales del esquema y conexiones antes de cada fase de pruebas.
- 5. **Errores no detectados en los módulos incompletos:**
 - a. Los módulos de **Caja y Ventas** no están completos, lo que puede afectar su integración futura.
 - b. **Impacto:** Medio.
 - c. **Probabilidad:** Alta.
 - d. **Plan de mitigación:** Realizar pruebas limitadas en las vistas y reportar posibles problemas de diseño o funcionalidad al desarrollador.

Premisas que podrían no cumplirse

- **Dependencia:** El entorno local funcionará sin problemas durante todo el proceso de pruebas.
- **Riesgo:** Problemas técnicos inesperados que interrumpan el proceso de pruebas.
- **Plan de contingencia:** Mantener configuraciones respaldadas y planificar ventanas de tiempo para resolver problemas críticos si ocurren.

Al clasificar estos riesgos y sus planes, se busca garantizar la calidad del software dentro de los límites establecidos por el cronograma y los recursos disponibles.

Referencias

A continuación, se listan los documentos y recursos utilizados como apoyo para la elaboración del plan de pruebas, así como los que pueden ser consultados para ampliar su contenido:

- **Plan de Proyecto**
 - Documento que establece los objetivos generales del desarrollo del software, cronograma, recursos asignados y alcances definidos para el proyecto.
- **Especificaciones de Requerimientos**
 - Detalle de los requisitos funcionales y no funcionales del sistema, incluyendo descripciones claras de los módulos y funcionalidades.
- **Diseño General**
 - Diagrama de arquitectura del sistema y esquema de bases de datos, con énfasis en la relación entre los módulos del software.
- **Diseño Detallado**
 - Documentación técnica que incluye diagramas de flujo, estructura de los módulos y descripción del código base, siguiendo el modelo MVC implementado.
- **Procedimientos y Estándares de Desarrollo**
 - Guías empleadas para garantizar la calidad y consistencia del código, incluyendo buenas prácticas en **PHP**, **HTML**, **CSS** y **JavaScript**.
- **Procedimientos y Estándares de Pruebas**
 - Metodologías aplicadas durante el proceso de pruebas, incluyendo los tipos de pruebas (unitarias, de integración, de regresión) y los criterios de aceptación establecidos.

Glosario

A continuación, se presentan las definiciones de términos relevantes utilizados en esta documentación y en el área de pruebas de software:

1. Plan de Pruebas

Documento que describe el enfoque, los recursos y el cronograma para las actividades de prueba de software.

2. Pruebas Funcionales

Verificación de que el sistema cumple con los requisitos funcionales establecidos en las especificaciones.

3. Pruebas No Funcionales

Evaluación de atributos de calidad del software, como rendimiento, escalabilidad y usabilidad.

4. Pruebas de Regresión

Conjunto de pruebas diseñadas para garantizar que las funcionalidades previamente desarrolladas y probadas sigan funcionando después de realizar cambios o adiciones al sistema.

5. Pruebas Unitarias

Pruebas que verifican el funcionamiento de componentes individuales del software, como funciones, métodos o clases.

6. Pruebas de Integración

Validación de la interacción entre diferentes módulos o componentes del sistema.

7. Módulo

Unidad lógica del software que agrupa funcionalidades específicas, como los módulos de usuarios, categorías, productos, etc.

8. Caso de Prueba

Descripción detallada de un escenario de prueba, incluyendo condiciones de entrada, procedimientos de prueba y resultados esperados.

9. Defecto (Bug)

Problema o error en el software que provoca resultados incorrectos o comportamiento inesperado.

10. Criterios de Aceptación

Conjunto de condiciones que deben cumplirse para que un módulo o el sistema completo sea aceptado como correcto.

11. Criterios de Suspensión

Condiciones que determinan cuándo las pruebas deben detenerse debido a problemas graves que impidan continuar con el proceso.

12. Criterios de Reanudación

Condiciones que establecen cuándo se pueden reanudar las pruebas después de una suspensión.

13. Entorno de Pruebas

Configuración de hardware, software, red y datos utilizada para llevar a cabo las pruebas.

14. Riesgo

Posibilidad de que ocurra un problema que impacte negativamente el desarrollo o pruebas del software.

15. Mitigación de Riesgos

Acciones diseñadas para reducir la probabilidad o el impacto de un riesgo identificado.

16. Plan de Contingencia

Estrategias establecidas para responder a un riesgo que no puede ser mitigado.

17. Tester

Persona responsable de ejecutar las pruebas y documentar los resultados para asegurar la calidad del software.

18. PHP

Lenguaje de programación utilizado para el desarrollo de funcionalidades del lado del servidor.

19. HTML, CSS y JavaScript

Lenguajes usados para el diseño, estructura y comportamiento interactivo de la interfaz de usuario del software.