

Proceso de Planeación de Pruebas

Integrantes:

- Andres Ricardo Chamorro
- Diego Armando Polanco
- Luis Manuel Rojas Correa
- Oscar Muñoz Ramírez
- Sebastián Erazo Ochoa

Contexto del Proyecto

Encora, una empresa líder en la industria de servicios digitales nativos, ingeniería de productos y experiencia digital, ha encargado a los estudiantes de quinto semestre de la Universidad Icesi la creación de un proyecto de "Online Marketplace for freelancers" como parte de la asignatura de Proyecto Integrador. Este proyecto tiene como objetivo desarrollar una plataforma en línea que facilite la interacción entre freelancers y clientes (empresas contratistas). Durante el desarrollo, los estudiantes no solo adquirirán habilidades técnicas en Django (backend), desarrollo web (frontend), y manejo de bases de datos, sino que también profundizará en el uso de metodologías ágiles como Scrum, permitiéndoles obtener una experiencia integral en tecnologías avanzadas y prácticas de desarrollo colaborativo.

Plan de Pruebas

1. Objetivo

Establecer un plan de pruebas integral que guíe el proceso de validación y verificación de la plataforma *FreeLancer Marketplace*, asegurando que se cumplan todos los requisitos de calidad, funcionalidad y seguridad definidos en las especificaciones del proyecto.

Objetivos Específicos:

- Delimitar qué áreas de la plataforma serán objeto de prueba, determinando las funcionalidades críticas y los aspectos que requieren mayor atención.
- Crear un calendario detallado que asigne tiempos específicos para cada fase del proceso de pruebas, incluyendo la planificación, ejecución y evaluación de resultados.
- Identificar a los miembros del equipo que estarán a cargo de cada fase del proceso de pruebas, asegurando una clara división de tareas y responsabilidades.
- Determinar las herramientas y metodologías más adecuadas para realizar las pruebas, incluyendo pruebas automatizadas, pruebas manuales, y herramientas de seguimiento de errores.

SOPAS Group**NextTask: Marketplace Freelancer**

- Crear una lista detallada de los escenarios de prueba, definiendo los criterios de aceptación para cada funcionalidad y asegurando una cobertura completa del sistema.
- Definir los criterios que determinarán si una prueba ha sido exitosa o si se requiere una acción correctiva, asegurando que estos criterios estén alineados con los objetivos del proyecto.
- Identificar posibles riesgos asociados al proceso de pruebas y desarrollar estrategias para mitigarlos, garantizando la continuidad del proyecto en caso de contratiempos.

2. Estrategia de Pruebas

Las pruebas se llevarán a cabo en varias etapas, con un enfoque en diferentes niveles y características de calidad del software.

1. Planificación y Preparación

- **Revisión de Requisitos:** Analizar y revisar todos los requisitos funcionales y no funcionales para comprender las necesidades del sistema.
- **Diseño de Casos de Prueba:** Crear casos de prueba detallados que cubran todos los escenarios posibles, tanto de éxito como de fallo.
- **Configuración de Entornos de Prueba:** Configurar los entornos de desarrollo, pruebas y producción, asegurando que sean representativos del entorno real de los usuarios.

2. Pruebas por Módulo**Módulo de Autenticación de Usuarios:**

- Pruebas de registro, inicio de sesión, recuperación de contraseñas, y gestión de perfiles.
- Validación de la seguridad en la autenticación y autorización, asegurando que solo usuarios autorizados puedan acceder a ciertas funcionalidades.

Módulo de Gestión de Proyectos:

- Verificación de la creación, edición, y finalización de proyectos por parte de los usuarios.
- Validación de la gestión de ofertas, aceptaciones y retroalimentación en cada proyecto.

Módulo de Pasarela de Pagos:

- Pruebas de integración con proveedores de pago, verificando la correcta ejecución de transacciones.
- Pruebas de procesamiento de pagos a los freelancers.

Módulo de Búsqueda y Filtros:

SOPAS Group**NextTask: Marketplace Freelancer**

- Validación del sistema de búsqueda, asegurando que los usuarios puedan encontrar proyectos y freelancers con base en criterios específicos.
- Pruebas de rendimiento para asegurarse de que la búsqueda y los filtros funcionan eficientemente incluso con grandes volúmenes de datos.

3. Pruebas de Integración

- Validar la integración de todos los módulos anteriores para asegurar que interactúen correctamente entre sí.
- Probar la comunicación y sincronización con sistemas externos, si los hubiera, como servicios de terceros para la gestión de pagos.

4. Pruebas de Usuario y Experiencia de Usuario (UX)

- Realizar pruebas con usuarios reales para evaluar la facilidad de uso y la navegabilidad de la plataforma.
- Recolectar retroalimentación sobre la interfaz de usuario y hacer ajustes necesarios.

5. Pruebas de Rendimiento y Escalabilidad

- Realizar pruebas de carga y estrés para evaluar cómo se comporta la plataforma bajo diferentes condiciones de tráfico y uso intensivo.
- Asegurar que la plataforma pueda manejar múltiples usuarios y transacciones simultáneamente sin comprometer el rendimiento.

6. Pruebas de Seguridad

Condición: Si se realiza correctamente la prueba de rendimiento y estabilidad del proyecto, se desea una prueba de seguridad con una buena eficacia

- Identificar posibles vulnerabilidades de seguridad en la plataforma y realizar pruebas de penetración para evaluar la resistencia del sistema a ataques externos.
- Validar el manejo seguro de datos sensibles y asegurar que se cumplan las normativas de protección de datos.

7. Recopilación de Resultados y Cierre

- Recopilar y analizar los resultados de las pruebas, generando informes detallados sobre los hallazgos.
- Presentar los resultados a las partes interesadas y tomar decisiones sobre las correcciones y mejoras necesarias.
- Cierre formal del proceso de pruebas, asegurando que todos los aspectos críticos han sido validados y corregidos según sea necesario.

3. Esquema de Trabajo

Para nuestro esquema de trabajo se definieron roles e intervalos de tiempo dedicados a coordinación y comunicación con el equipo de trabajo y las partes interesadas.

Roles y Responsabilidades:**Scrum Master (Gerente de Pruebas - Amarillo):**

Una persona. Este rol será asumido por el Gerente de Pruebas, quien será responsable de supervisar y coordinar todas las actividades de prueba dentro del equipo SCRUM. El Scrum Master (Gerente de Pruebas) se asegurará de que el plan de pruebas se ejecute correctamente, facilitando las ceremonias SCRUM (como las reuniones diarias, planificaciones de sprint, y revisiones), y reportará el progreso a los stakeholders.

Product Owner (Analista de Pruebas - Rojo):

Una persona. Este rol será desempeñado por el Analista de Pruebas, quien será el encargado de definir y priorizar los casos de prueba en el backlog del producto, asegurándose de que todas las funcionalidades críticas sean evaluadas. El Product Owner (Analista de Pruebas) también documentará los casos de prueba basados en los requerimientos del sistema y trabajará estrechamente con los stakeholders para clarificar las expectativas y prioridades del proyecto.

Equipo de Desarrollo (Ingenieros de Pruebas - Verde):

Tres personas. Los Ingenieros de Pruebas formarán parte del Equipo de Desarrollo en SCRUM. Serán responsables de ejecutar los casos de prueba durante los sprints, documentar los resultados, reportar defectos y colaborar activamente con el resto del equipo de desarrollo para resolver los problemas encontrados. Además, participarán en la planificación del sprint y en la revisión de las pruebas realizadas.

Coordinación y Comunicación:**Reuniones Diarias (stand-up):**

Objetivo: Revisar el progreso y planificar el trabajo diario.

Participantes: Todos los miembros del equipo

Duración: 15-30 minutos

Reuniones Semanales:

Objetivo: Revisar el progreso general, discutir problemas y planificar la semana siguiente.

Participantes: Todos los miembros del equipo

Duración: 1 hora

Reuniones con Stakeholders:

Objetivo: Presentar avances, discutir problemas críticos y recibir feedback.

SOPAS Group**NextTask: Marketplace Freelancer**

Participantes: Gerente de Pruebas, Analistas de Pruebas, Ingenieros de Pruebas, Stakeholders.

Duración: 1- 2 horas

Responsable de Conducción: Gerente de Pruebas.

Fase	Actividades–	Semana													
		2-3	4-5	6	7	8	9	10	11	12	13	14	15	16	
Planificación y Preparación	Revisión de requisitos, diseño de casos de prueba, figma, gestión de jira, diseño de diagrama de clases, bases de datos, diagrama de casos de usos, diagrama de secuencia, y la gestión de la organización de los informes de jira	<div><div>XX</div><div>X</div></div>	<div><div>XX</div><div>X</div></div>												
Módulo de Autenticación de Usuarios	Verificación del proceso de registro de usuarios, validación de credenciales, recuperación de contraseñas, gestión de perfiles.			<div><div>X</div></div>	<div><div>X</div></div>										
Módulo de Gestión de Proyectos	Verificación de creación y modificación de proyectos, validación del proceso de finalización de proyectos.					<div><div>X</div><div>x</div></div>	<div><div>x</div></div>	<div><div>xx</div></div>							
Módulo de pasarela de pagos	Pruebas de integración con pasarelas de pago, validación del proceso de pagos a freelancers.								<div><div>xx</div></div>	<div><div>X</div></div>					
Módulo de búsqueda y filtros	Pruebas de funcionamiento del sistema de búsqueda, evaluación del rendimiento del sistema bajo grandes volúmenes de datos										<div><div>x</div></div>	<div><div>x</div></div>			
Informe de resultados y cierre de pruebas	Recopilación de resultados, análisis, y elaboración de informes												<div><div>x</div></div>	<div><div>X</div></div>	

SOPAS Group

NextTask: Marketplace Freelancer



4. Herramientas de Apoyo

Para garantizar un proceso de pruebas eficiente y organizado, se utilizarán diversas herramientas de apoyo que facilitarán la automatización, gestión y seguimiento de las pruebas. A continuación se detallan las principales herramientas que se emplearán:

Software Utilizado:

1. Selenium

Selenium es una suite de herramientas para la automatización de pruebas de aplicaciones web. Permite simular interacciones de usuario en navegadores web y automatizar la ejecución de casos de prueba. Se utilizará para automatizar las pruebas funcionales y de regresión del software "Freelancers Marketplace", asegurando que las funcionalidades de la aplicación web se comporten como se espera.

2. Jira Software

Jira es una herramienta de gestión de proyectos y seguimiento de incidentes ampliamente utilizada en el desarrollo de software. Permite gestionar tareas, bugs y el ciclo de vida de las pruebas. Se utilizará para gestionar los casos de prueba, registrar y hacer seguimiento de los defectos encontrados, y coordinar las actividades del equipo de pruebas.

3. SonarQube

SonarQube es una plataforma para la revisión continua de la calidad del código, detectando problemas de seguridad, bugs y deuda técnica. Se integrará en el proceso de desarrollo para analizar la calidad del código fuente y asegurar que cumple con los estándares de calidad.

4. –Bootstrap

Bootstrap es un framework de diseño web que facilita la creación de interfaces responsivas y atractivas. Se empleará para el desarrollo de la interfaz de usuario del software, asegurando un diseño consistente y adaptable a diferentes dispositivos.

5. Git

Git es un sistema de control de versiones distribuido que facilita la gestión del código fuente a lo largo del ciclo de desarrollo. Será utilizado para el seguimiento de cambios en el código, permitiendo la colaboración efectiva entre los desarrolladores.

6. GitHub

GitHub es una plataforma de alojamiento de código basada en Git que facilita la colaboración y la gestión de proyectos. Se utilizará para almacenar el código fuente del proyecto, gestionar issues, pull requests, y colaborar con el equipo de desarrollo.

7. Anaconda

Anaconda es una distribución de Python que incluye una gran cantidad de paquetes y herramientas para la ciencia de datos y el desarrollo de software. Será utilizado para gestionar entornos de desarrollo y asegurar la consistencia en las dependencias utilizadas en el proyecto.

8. Python

Python es un lenguaje de programación versátil y ampliamente utilizado en el desarrollo de software y automatización de pruebas. Se empleará para la creación de scripts y herramientas personalizadas que apoyen el proceso de pruebas.

9. Visual Studio Code (VS Code)

Visual Studio Code es un editor de código fuente ligero pero potente, con soporte para una amplia variedad de lenguajes de programación y herramientas de desarrollo. Se utilizará como editor principal para el desarrollo del proyecto, aprovechando sus capacidades de depuración y extensibilidad.

10. PyCharm

PyCharm es un entorno de desarrollo integrado (IDE) para Python que proporciona herramientas avanzadas para la escritura, depuración y refactorización de código Python. Será utilizado para facilitar el desarrollo de software de alta calidad en Python.

11. Django

Django es un framework de desarrollo web en Python que permite crear aplicaciones web robustas y escalables. Se empleará para desarrollar el backend del software "Freelancers Marketplace", asegurando una arquitectura sólida y segura.

12. Herramientas de diseño web (HTML, CSS, JavaScript)

Estas herramientas son fundamentales para la creación y diseño de la interfaz de usuario del software. HTML proporcionará la estructura de las

páginas web, CSS se utilizará para definir su estilo visual, y JavaScript permitirá agregar interactividad a la interfaz de usuario.

5. Tipos de Prueba a Aplicar:

Módulo	Tipo de Prueba	Descripción
Módulo de Autenticación de Usuarios	Pruebas Funcionales	Verificación de funcionalidades como registro, inicio de sesión, recuperación de contraseñas.
Módulo de Gestión de Proyectos	Pruebas Funcionales	Validación del proceso de creación, edición, y finalización de proyectos.
Módulo de Pasarela de Pagos	Pruebas de Integración	Verificación de la integración con proveedores de pago y procesamiento de transacciones.
	Pruebas de Seguridad	Asegurar la seguridad en la transmisión de datos financieros.
Módulo de Búsqueda y Filtros	Pruebas de Rendimiento	Evaluación del tiempo de respuesta y la capacidad de manejo de grandes volúmenes de datos.

Los tipos de prueba que aplicaremos serán las siguientes:

a. **Pruebas de volumen:** Estas pruebas se llevarán a cabo para verificar la capacidad del sistema para manejar un gran número de proyectos y transacciones dentro de la plataforma Freelancer Marketplace. Se evaluará cómo el sistema gestiona altos volúmenes de datos, incluyendo la creación de proyectos, la búsqueda de freelancers, y la gestión de pagos, asegurando que el rendimiento y la funcionalidad se mantengan óptimos bajo condiciones de alta carga.

b. **Pruebas de integración:** Estas pruebas se llevarán a cabo para validar la interoperabilidad entre los diferentes módulos del sistema, como el módulo de gestión de proyectos, el módulo de pasarela de pagos, y el módulo de autenticación. Se asegurará que

SOPAS Group**NextTask: Marketplace Freelancer**

todos los módulos funcionen en conjunto de manera coherente y que los datos fluyan correctamente entre ellos.

c. **Pruebas funcionales:** Estas pruebas se centrarán en verificar que cada funcionalidad de la plataforma Freelancer Marketplace, como la creación de perfiles, la gestión de proyectos, y la ejecución de pagos, opere conforme a los requisitos especificados. Se asegurará que todas las funcionalidades esenciales del sistema estén implementadas correctamente y sin errores.

Todas estas pruebas, son realizadas por los ingenieros de pruebas.

6. Alcance Funcional

Descripción de Procesos y Subprocesos:

Autenticación de Usuarios

- Registro de Usuarios
- Inicio de Sesión y Recuperación de Contraseña

Gestión de Perfiles

- Actualización de Perfiles
- Visualización de Perfiles

Gestión de Proyectos

- Creación de Proyectos
- Búsqueda y Filtro de Proyectos
- Asignación y Gestión de Proyectos

Sistema de Búsqueda y Filtros

- Búsqueda de Freelancers
- Filtrado de Resultados

Pasarela de Pagos

- Procesamiento de Pagos
- Historial de Transacciones

Evaluación y Retroalimentación

- Sistema de Calificación
- Revisión y Comentarios

Matriz de descomposición de funcionalidades (MDF)

Proceso de negocio	Orden	Funcionalidades del sistema
--------------------	-------	-----------------------------

Autenticación de Usuarios	1	Registro de Usuarios
	2	Confirmación de correo electrónico
	3	Inicio de sesión
	4	Recuperación de contraseña
	5	Gestión de perfiles
	6	Cierre de sesión
Gestión de proyectos	1	Crear proyecto
	2	Editar proyecto
	3	Publicación de proyectos
	4	Asignación de proyectos a Freelancers
	5	Finalización de proyectos
Sistema de Búsqueda y Filtros	1	Búsqueda de Freelancers
	2	Filtros de Búsqueda
	3	Ordenamiento de Resultados
	4	Visualización de Resultados
Pasarela de Pagos	1	Procesamiento de Pagos
	2	Integración con Proveedores de Pago
	3	Gestión de Transacciones
	4	Historial de Transacciones
Evaluación y Retroalimentación	1	Sistema de Calificación
	2	Visualización de Calificaciones y Comentarios

	3	Modificación de Calificaciones y Comentarios
--	---	--

7. Esfuerzo Estimado

Valores de tiempo estimado por semana para cada rol:

Gerente de pruebas: 10 horas por semana.

Analista de pruebas: 15 horas por semana.

Ingeniero de pruebas: 20 horas por semana.

Calcular el esfuerzo total por trabajador:

Gerente de pruebas: $10/\text{semana} \times 1 \text{ gerente} = 10 \text{ horas}$

Analista de pruebas: $15 \text{ horas/semana} \times 2 \text{ analistas} = 30 \text{ horas}$

Ingeniero de pruebas: $20 \text{ horas/semana} \times 2 \text{ ingenieros} = 40 \text{ horas}$

8. Entregables del Proceso

Estrategia de Pruebas

Documento que describe el enfoque general y los principios que guiarán el proceso de pruebas. Incluye la definición de objetivos, el alcance de las pruebas, los tipos de pruebas a realizar, los criterios de éxito y fracaso, y las herramientas y técnicas a utilizar. Este documento sirve como una guía maestra para todas las actividades de prueba.

Plan de Pruebas

Documento detallado que especifica las actividades de prueba planificadas, el cronograma de ejecución, los recursos humanos y materiales necesarios, los riesgos identificados y las estrategias de mitigación. Incluye también los roles y responsabilidades de los miembros del equipo de pruebas y los criterios de entrada y salida para cada fase del proceso de pruebas.

Casos de Prueba y Datos de Prueba

Conjunto de documentos que detallan los escenarios específicos a probar, los pasos a seguir para ejecutar cada prueba, los datos necesarios para llevar a cabo las pruebas, y los resultados esperados. Los casos de prueba aseguran que todas las funcionalidades del sistema sean evaluadas exhaustivamente.

Informe de Resumen de Pruebas

Documento que presenta un resumen ejecutivo de los resultados de las pruebas realizadas. Incluye métricas clave, gráficos y tablas que ilustran el desempeño del sistema, la cobertura de pruebas, el número de defectos encontrados y corregidos, y una evaluación general de la calidad del sistema. Este informe se utiliza para informar a los stakeholders sobre el estado de las pruebas y la calidad del producto.

Informe de Cierre de Pruebas

Documento final que certifica la finalización del proceso de pruebas. Incluye un resumen de todas las actividades de prueba realizadas, los resultados obtenidos, las lecciones aprendidas y las recomendaciones para futuras fases de desarrollo. Este informe también documenta la aprobación final de las pruebas por parte del equipo de pruebas y los stakeholders.

Reporte de Incidentes

Registro detallado de todos los problemas y defectos encontrados durante las pruebas. Cada incidente incluye una descripción del problema, su severidad, el impacto en el sistema, el estado actual (abierto, en proceso, cerrado), y las acciones tomadas para resolverlo. Este documento es crucial para el seguimiento y gestión de defectos.

9. Mecanismos de seguimiento y control:

Se realiza seguimiento cualitativo y cuantitativo. Para el seguimiento cualitativo se realizan acciones de retrospectiva cada semana, durante las reuniones que se tienen planeadas y que se definieron previamente en el esquema de trabajo.

En caso de realizar un seguimiento y control cuantitativo, vamos a definir indicadores y graficar información que nos permita evaluar el desempeño del sistema.

- Cantidad de Ejecuciones de Casos de Pruebas Correctas y Fallidas.

- % de pruebas correctas: $\frac{\# \text{ de pruebas correctas}}{\text{total de pruebas}} * 100$

- % de pruebas fallidas: $\frac{\# \text{ de pruebas fallidas}}{\text{total de pruebas}} * 100$

- Esto nos permite conocer en términos generales cómo se encuentra nuestro programa.

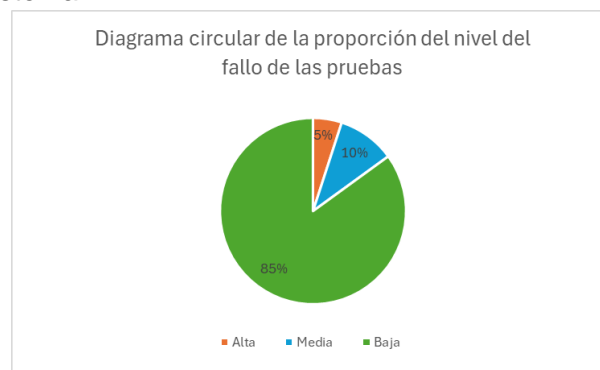
- Información sobre Defectos: Densidad de Errores, Frecuencia de Fallos.

- $\text{Densidad de errores} = \frac{\text{Número total de defectos}}{\text{Tamaño total de la base de código}} * 100$

- Nos indica que tan propagado están los errores por nuestro programa y saber el nivel de dedicación necesario para depurarlo.

- $$\text{Frecuencia de fallos} = \frac{\text{Número total de defectos encontrados}}{\text{Número total de pruebas ejecutadas}} * 100$$

- Porcentaje de completitud:
 - - $$\frac{\text{funcionalidades probadas}}{\text{total de funcionalidades}} * 100$$
 - Esto nos permite conocer la proporción del software a la que podemos garantizar que se ha probado y en caso de ser positivo, que podamos garantizar la calidad del mismo.
 - Cantidad de Errores por Importancia: Alta, Media, Baja.
 - Una gráfica que nos permita visualizar la gravedad del estado en que se encuentre el sistema.



- Este es un indicador visual que permite conocer el nivel de gravedad con el que se encuentra el sistema, lo ideal es que los niveles alto y medio sea una proporción pequeña o inexistente, ya que son aquellos fallos que pueden comprometer a nuestro sistema.
- Cantidad de errores por cada proceso de negocio:
 - Diagrama de barras que muestre el número de pruebas correctas y fallidas por cada proceso.
 - Este indicador visual nos permite conocer a que proceso de negocio debemos dedicarle más tiempo de desarrollo y corrección de errores.