

## **Plan de Pruebas para la Plataforma de Freelancers de Encora**

Jennifer Castro,  
Juan José Angarita,  
Geoffrey Pasaje,  
Juan Camilo Tobar

Facultad de Ingeniería, Universidad Icesi  
Proyecto Integrador I, 09707

Ingri Jojoa López, M.Sc.

1 de septiembre de 2024

## Objetivo de las Pruebas:

El propósito del plan de pruebas del sistema de FREELANCERS de Encora es garantizar que todas las funcionalidades del sistema se implementen correctamente y cumplan con los requisitos especificados. Este plan se centra en validar que todas las funcionalidades trabajen de manera efectiva y sin errores, proporcionando una experiencia de usuario coherente y confiable. El plan de pruebas detalla los escenarios de prueba, casos de uso, condiciones iniciales, resultados esperados y procedimientos para ejecutar las pruebas y reportar defectos.

El proceso de pruebas a realizar se basa en varias dimensiones clave:

**Validar Funcionalidades:** Evaluar las funcionalidades a forma de hacer la aplicación óptima, teniendo en cuenta la publicación de servicios, proyectos y habilidades por parte de los freelancers y la contratación, comentarios, reseñas, publicación de proyectos y métodos de pago por parte de los clientes. Además, verificar que estas características no solo cumplan con los requerimientos técnicos y funcionales, sino que también sean intuitivas y satisfactorias para los usuarios finales.

- **Asegurar la Integridad de los Datos:** Garantizar que todos los datos introducidos, modificados o eliminados en el sistema se manejen correctamente y que no se produzcan inconsistencias.
- **Evaluar la Seguridad y la Gestión de Usuarios:** Confirmar que la gestión de usuarios y roles se desempeñe adecuadamente, permitiendo agregar, editar y eliminar usuarios de manera segura, así como administrar permisos de manera eficaz.
- **Garantizar la fiabilidad del sistema:** Realizar pruebas de carga y rendimiento para asegurar que el sistema pueda manejar múltiples operaciones y usuarios simultáneamente sin degradar su rendimiento.
- **Validar la Integración con Sistemas Externos:** Comprobar la correcta integración del subsistema API con el sistema de información de la plataforma, garantizando que los datos se transfieran y registren adecuadamente.

## Estrategia de Pruebas:

Esta estrategia de pruebas para el sistema de Freelancers se enfoca en diversas áreas clave, organizadas en cuatro niveles de pruebas: pruebas unitarias, de integración, de sistema y de

aceptación. Cada nivel aborda diferentes aspectos del software, asegurando una evaluación exhaustiva de todas las funcionalidades y atributos de calidad.

### **Niveles de Pruebas:**

#### **Pruebas Unitarias:**

Estas pruebas se realizan a nivel de métodos y funciones individuales desarrolladas en el sistema. Se enfocan en componentes aislados, verificando que cada unidad de código funcione según lo previsto. Se llevan a cabo con el propósito de detectar errores de bajo nivel en los módulos y clases construidos por los desarrolladores. Los desarrolladores individuales serán los principales responsables de ejecutar estas pruebas, asegurando que cada parte del sistema funcione correctamente de manera independiente antes de proceder a su integración con otros componentes.

#### **Pruebas de Integración:**

Se centran en la interacción entre los diferentes módulos y con sistemas externos. En nuestro contexto, es crucial probar la correcta integración del subsistema API con el sistema de información. Estas pruebas garantizarán que los datos se transfieran y registren adecuadamente y que los componentes integrados funcionen juntos sin problemas. La colaboración del equipo de integración será fundamental para abordar y resolver cualquier problema que surja durante estas pruebas, asegurando una comunicación fluida y eficiente entre todos los sistemas involucrados.

#### **Pruebas de Sistema:**

La evaluación del sistema completo en su entorno operativo es el foco de estas pruebas. Se validan todas las características funcionales del programa, incluyendo el registro y autenticación de los usuarios, la publicación de proyectos, cargue de documentos e información acerca de los freelancers, comentarios, reseñas y calificaciones. Estas pruebas aseguran el cumplimiento de los requisitos técnicos y funcionales y que las características sean intuitivas y satisfactorias para los usuarios finales. Se considerarán aspectos como la usabilidad, la eficiencia y la experiencia del usuario en general, garantizando una implementación coherente y sin errores.

#### **Pruebas de Aceptación:**

Las pruebas de aceptación involucrarán la participación de los stakeholders y los usuarios finales del sistema. Estas pruebas se enfocan en validar que el sistema cumple con los requisitos especificados. El objetivo principal es confirmar la satisfacción de los stakeholders con el sistema proporcionado, asegurando que todas las funcionalidades sean coherentes con las expectativas del usuario y que el sistema funcione de manera fiable y segura en un entorno real. La retroalimentación obtenida durante estas pruebas será crucial para realizar ajustes finales antes del lanzamiento oficial del sistema.

### Matriz de pruebas de software

Relación de cada prueba con los atributos de calidad de software

	Funcionalidad	Usabilidad	Rendimiento	Seguridad	Compatibilidad	Mantenibilidad	Portabilidad
Pruebas Unitarias	x					x	
Pruebas de Integración	x				x	x	
Pruebas de Sistema	x	x	x	x	x	x	x
Pruebas de Aceptación	x	x	x	x	x		x

## **Áreas de Enfoque de enfoque extra**

### **Asegurar la Integridad de los Datos:**

Se implementarán medidas exhaustivas para velar para que todos los datos introducidos, modificados o eliminados en el sistema se manejen correctamente. Esto es esencial para evitar inconsistencias y asegurar la precisión y coherencia de la información. Para lograrlo, se incluirán pruebas de integridad de datos y revisiones de consistencia en todas las fases del proceso de pruebas. Estas pruebas se diseñarán para verificar que los datos se almacenen y recuperen de manera adecuada, asegurando que cualquier operación realizada en el sistema no comprometa la integridad de la base de datos.

### **Evaluar la Seguridad y la Gestión de Usuarios:**

**Descripción:** Confirmar que la gestión de usuarios y roles se desempeñe adecuadamente.

**Objetivo:** Permitir agregar, editar y eliminar usuarios de manera segura, así como administrar permisos eficazmente.

**Estrategia:** Se llevarán a cabo pruebas de seguridad y validación de permisos y roles durante las pruebas de sistema y de aceptación. Estas pruebas garantizarán que solo los usuarios autorizados puedan realizar ciertas acciones y acceder a información sensible, asegurando que el sistema sea robusto frente a posibles amenazas de seguridad. Además, se verificará que acciones como las reseñas (las cuales solo puede ser realizadas por personas que se hayan verificado que usaron los servicios del freelancer) se reflejen correctamente en todo el sistema, manteniendo siempre la seguridad y la integridad de los datos.

### **Garantizar la Fiabilidad del Sistema:**

**Descripción:** Realizar pruebas de carga y rendimiento para asegurar que el sistema pueda manejar múltiples operaciones y usuarios simultáneamente.

**Objetivo:** Asegurar que el rendimiento no se degrade bajo condiciones de uso intensivo.

**Estrategia:** Se realizarán pruebas de carga y estrés, así como el monitoreo del rendimiento durante las pruebas de sistema. Estas pruebas simularán escenarios de alta demanda para

evaluar cómo se comporta el sistema bajo presión, identificando posibles cuellos de botella y asegurando que el sistema mantenga un rendimiento óptimo incluso en condiciones de uso extremo. Además, se utilizarán herramientas de monitoreo para observar el desempeño en tiempo real, permitiendo ajustes proactivos para mejorar la eficiencia y la estabilidad del sistema.

### **Fases de prueba:**

**Planificación:** En esta fase inicial, se lleva a cabo un análisis exhaustivo del contexto del sistema de freelancers, identificando los objetivos de las pruebas, las funcionalidades críticas y los riesgos potenciales. Se define el esquema de trabajo, que incluye la selección de los tipos de pruebas a realizar, los recursos humanos y técnicos necesarios, y las condiciones para iniciar el proyecto de pruebas. El principal entregable de esta fase es un **Plan de Pruebas detallado**, que especificará los objetivos, alcance, estrategia, cronograma, roles y responsabilidades, así como los criterios de entrada y salida de cada fase de pruebas. Este plan será revisado y aprobado por todas las partes involucradas, garantizando la alineación y compromiso con los objetivos del proyecto.

**Diseño:** Con base en el análisis previo y el Plan de Pruebas, se elabora un cronograma detallado que guiará la ejecución de las pruebas. En esta fase, analizaremos las listas de chequeo, los requisitos funcionales y no funcionales, y los mockups de las funcionalidades del sistema. Se diseñarán los casos de prueba que cubrirán tanto los escenarios positivos como los negativos, así como las pruebas de borde y las pruebas de regresión. Los casos de prueba serán documentados de manera detallada, especificando los pasos a seguir, los datos de entrada, los resultados esperados y los criterios de aceptación. El resultado de esta fase es un **Documento de Casos de Prueba**, que servirá como referencia para la fase de ejecución.

**Ejecución:** Durante la ejecución, se llevan a cabo las pruebas planificadas en un entorno de pruebas controlado y representativo del entorno de producción. Esta fase incluye al menos dos iteraciones. En la primera se realizan pruebas funcionales para validar las nuevas características del sistema y pruebas de regresión para asegurar que las funcionalidades existentes no se vean afectadas. Mientras que la segunda se enfocará en la revisión de las correcciones realizadas tras la primera iteración, y se repetirán las pruebas de regresión en las áreas afectadas por los cambios. Los developers proporcionarán información sobre el impacto de las correcciones, lo que permitirá priorizar y seleccionar los casos de prueba

más críticos. Se documentarán todos los resultados, errores y cualquier desviación respecto a los resultados esperados.

**Seguimiento, Control y Medición:** Se realizará un monitoreo continuo del progreso y de los resultados de las pruebas mediante indicadores clave de calidad, como la tasa de fallos, el tiempo de corrección de errores y la cobertura de pruebas. Se generarán informes periódicos que detallarán el estado de las pruebas, los defectos encontrados, su gravedad y las acciones tomadas para resolverlos. Estos informes nos permitirán evaluar si se están cumpliendo los objetivos del Plan de Pruebas y facilitarán la toma de decisiones sobre ajustes en el proceso de pruebas.

**Cierre:** Al finalizar las iteraciones de prueba, se evaluará si se han alcanzado los niveles de calidad aceptables para el sistema o si es necesaria una tercera iteración de pruebas. Se elaborará un **Informe Final del Proceso de Pruebas**, que resumirá los logros en términos de calidad del producto, los defectos corregidos, los riesgos mitigados y los resultados generales obtenidos. Este informe servirá como documento de referencia para autorizar el paso del sistema a la fase de producción, asegurando que el producto esté listo para su uso por parte de los usuarios finales.

### **Esquema de Trabajo:**

Para asegurar una gestión efectiva del proyecto de freelancers, se establecerá un cronograma de reuniones que facilite una comunicación fluida y una coordinación constante entre todas las partes involucradas. Se llevarán a cabo las siguientes reuniones de forma regular:

### **Reuniones Programadas:**

#### **Reuniones diarias de Equipo:**

Frecuencia: Diaria

Participantes: Todo el equipo

**Objetivo:** Realizar un seguimiento de avance de todos los miembros del equipo y coordinar las tareas a ejecutar al día siguiente, incluyendo las relacionadas con las pruebas.

**Reuniones Semanales de Equipo:**

Frecuencia: Semanal.

Participantes: Equipo de pruebas (líder y testers).

Objetivo: Discutir el progreso del trabajo, los desafíos encontrados y cualquier cambio en los requisitos o enfoques de prueba. Mantener a todo el equipo actualizado y alineado con las metas del proyecto.

**Reuniones Quincenales de Coordinación:**

Frecuencia: Quincenal.

Participantes: Equipo de pruebas, cliente (en este caso la profesora es la persona más cercana al cliente, por lo cual buscaremos su retroalimentación), desarrolladores.

Objetivo: Revisar los avances del proyecto, discutir posibles ajustes en los planes de prueba y resolver cualquier problema o duda que surja. Alinear expectativas y prioridades entre todas las partes involucradas.

**Reuniones Mensuales de Revisión General:**

Frecuencia: Mensual.

Participantes: Todos los stakeholders relevantes.

Objetivo: Presentar un informe detallado del progreso del proyecto, evaluar el cumplimiento de los objetivos, identificar áreas de mejora y tomar decisiones estratégicas para el siguiente ciclo de desarrollo y pruebas.

Comunicación Constante:

Medios: Correos electrónicos, mensajes instantáneos.

Objetivo: Abordar cualquier problema urgente o consultar sobre aspectos específicos del proyecto de gestión de becas.

**Roles y Funciones del Equipo:**

**Juan Camilo Tobar - Coordinador de Pruebas Unitarias**



**Responsabilidades:**

- *Planificación de Pruebas Unitarias:* definir qué pruebas unitarias se necesitan y cómo se van a ejecutar, trabajando de cerca con los desarrolladores para entender bien cada componente. Su objetivo es que cada módulo del sistema funcione perfectamente por sí solo, evitando errores desde el inicio, desarrollando estrategias de prueba que incluyan la selección de herramientas adecuadas y la definición de los casos de prueba unitarios necesarios.
- *Ejecución de Pruebas Unitarias:* Realiza las pruebas asegurándose de que cada parte individual del software cumple con los requisitos y no tiene fallos. Si algo no funciona como se espera, trabaja con los desarrolladores para encontrar una solución rápida.
- *Automatización de Pruebas:* Ayuda a implementar herramientas de automatización para agilizar el proceso de pruebas y detectar errores de manera más eficiente, lo cual ahorra tiempo y mejora la calidad del software.
- *Documentación de Resultados:* Documenta los resultados de cada prueba, registrando qué funcionó, qué no, y por qué. Esta información es clave para mejorar el software y evitar que los mismos errores se repitan.

**Geoffrey Pasaje - Responsable de Pruebas de Integración:****Responsabilidades:**

- *Planificación de Pruebas de Integración:* Coordinará cómo se van a probar los distintos módulos del sistema en conjunto, asegurándose de que todo funcione bien cuando los componentes se combinan. Su trabajo incluye planificar las pruebas y definir qué escenarios deben cubrirse.
- *Ejecución de Pruebas de Integración:* Se enfoca en probar cómo interactúan los módulos entre sí y con otros sistemas externos. Si detecta problemas de integración, trabaja para resolverlos y asegurar que no haya fallos en la interacción entre los componentes.
- *Gestión de Datos de Prueba:* Prepara los datos que se necesitan para las pruebas, asegurando que sean representativos y seguros. Esto incluye crear escenarios que reflejen situaciones reales de uso del sistema.
- *Comunicación de Resultados:* Documenta los resultados y los comunica al equipo, asegurando que todos estén al tanto de los problemas encontrados y de las acciones correctivas tomadas.

### **Jennifer Castro - Líder de Pruebas de Sistema:**

#### **Responsabilidades:**

- *Coordinación de Pruebas de Sistema:* Organizará y liderará las pruebas que se realizan sobre el sistema completo, asegurándose de que todo funcione bien en conjunto y que se cumplan los requisitos del proyecto. También definirá los objetivos de cada prueba, supervisando su ejecución.
- *Ejecución de Pruebas de Sistema:* Supervisar y participar activamente en las pruebas del sistema completo, verificando que el software funcione correctamente bajo diferentes condiciones de uso. Ha de asegurar que el sistema responda adecuadamente y cumpla con las expectativas del usuario.
- *Revisión de Resultados:* Analiza los resultados de las pruebas, identifica problemas y coordina con el equipo para solucionarlos, manteniendo a todos informados sobre el estado de las pruebas y qué tan cerca está el sistema de cumplir con los estándares de calidad.

### **Juan José Angarita - Coordinador de Pruebas de Aceptación:**

#### **Responsabilidades:**

- *Planificación de Pruebas de Aceptación:* Se encargará de definir el enfoque y los criterios de aceptación en coordinación con la profesora, quien actuará como la principal evaluadora del sistema. Planificará las pruebas para asegurar que el sistema cumple con los requisitos definidos y con las expectativas académicas del proyecto.
- *Ejecución de Pruebas de Aceptación:* Coordinar y llevar a cabo las pruebas de aceptación basándose en las directrices y retroalimentaciones de la profesora. Su objetivo es validar que el sistema está listo para su implementación final y cumple con los estándares exigidos.
- *Documentación de Resultados y Retroalimentación:* Documentar cuidadosamente los resultados de las pruebas de aceptación, comunicando los hallazgos de manera clara y detallada. Asegura que las observaciones de la profesora se tomen en cuenta para hacer los ajustes necesarios antes de que el sistema sea aprobado.
- *Gestión de la Retroalimentación:* Recoge y analiza las observaciones proporcionadas durante las pruebas, asegurándose de implementar los cambios requeridos. Esta gestión es clave para lograr un producto final que cumpla con las expectativas de los stakeholders y usuarios finales.

**Analistas de Pruebas:**

Número de Personas Necesarias: 4 (uno para cada tipo de prueba principal: unitaria, integración, sistema, y aceptación).

Responsabilidades: Realizar las pruebas asignadas, documentar los resultados, y reportar cualquier defecto encontrado.

**Usuarios y Stakeholders:**

Número de Personas Necesarias: Al menos 2-3 usuarios finales y 2-3 stakeholders clave.

Responsabilidades: Participar en las pruebas de aceptación, proporcionar retroalimentación sobre la funcionalidad y usabilidad del sistema, y validar que el sistema cumple con las expectativas y requisitos del negocio.

**Evaluación de Seguridad Adicional:**

El equipo realizará una evaluación exhaustiva de la seguridad del sistema utilizando herramientas internas y metodologías de pruebas de penetración (pentesting). Los miembros del equipo de desarrollo y pruebas, capacitados en prácticas de seguridad, identificarán posibles vulnerabilidades y puntos débiles en la infraestructura y en las aplicaciones. Esta evaluación contribuirá a reforzar la protección y la confidencialidad de los datos, asegurando que el sistema sea seguro antes de su implementación.

**Herramientas de Apoyo:**

En el contexto del proyecto, utilizaremos diversas herramientas y artefactos de ingeniería de software para facilitar y optimizar el proceso de pruebas. Estas herramientas abarcarán diferentes aspectos del proceso, desde la gestión de pruebas hasta la automatización, el seguimiento de errores y la integración continua. A continuación se detallan los artefactos de ingeniería y el software que se utilizará, cabe recalcar que esto es solo una proyección, pues se define sólo aquellos artefactos e herramientas pensados a priori de la realización del proyecto:

**Artefactos de Ingeniería**

- **Casos de Prueba (Test Cases):** Documentos detallados que describen las condiciones, entradas, acciones, y resultados esperados para una funcionalidad específica del sistema BECASIS. Los casos de prueba asegurarán que todas las características del sistema sean verificadas minuciosamente.
- **Matriz de Trazabilidad (Traceability Matrix):** Un documento que mapea los requerimientos del sistema a los casos de prueba correspondientes, asegurando que cada requisito ha sido cubierto por al menos un caso de prueba.
- **Planes de Prueba (Test Plans):** Documentos que definen el alcance, enfoque, recursos y calendario de las actividades de prueba. El plan de prueba incluye los objetivos de las pruebas, criterios de entrada y salida, y responsabilidades del equipo.
- **Scripts de Prueba Automatizados:** Scripts que automatizan la ejecución de pruebas repetitivas y regresivas. Estos scripts se crearán usando herramientas de automatización para asegurar que las pruebas se ejecuten de manera consistente y eficiente.

## Software de Apoyo

**Pytest:** Pytest es un marco de pruebas para Python que facilita la creación de pruebas unitarias y de integración para validar el correcto funcionamiento de los componentes del backend en Django. Es muy versátil y permite probar tanto la lógica de negocio como la interacción con la base de datos. La usaremos para probar las funcionalidades relacionadas con la gestión de usuarios (registro, autenticación), el cargue de documentos, la creación y actualización de perfiles de freelancers, la lógica de la pasarela de pagos, y la validación de la experiencia y proyectos listados.

**Jest:** Es un framework de pruebas para JavaScript, diseñado principalmente para probar aplicaciones React. Es ideal para realizar pruebas unitarias y de integración en el frontend, asegurando que los componentes de la interfaz se comporten correctamente. Lo usaremos para validar la correcta visualización y funcionamiento de las interfaces de usuario, como el sistema de reseñas y calificaciones de los freelancers, la gestión de proyectos (creación y visualización), y la integración del sistema de notificaciones en tiempo real.

**Postman:** Es una herramienta para probar y documentar APIs, permitiendo enviar solicitudes HTTP y verificar respuestas. Es útil para asegurar que las interacciones entre el frontend y backend cumplen con las especificaciones. La usaremos para probar la correcta ejecución de las API relacionadas con la gestión de proyectos (crear, listar, actualizar), el sistema de mensajería entre freelancers y clientes, y la integración con la pasarela de pagos, garantizando que todas las comunicaciones sean seguras y funcionales.

**Selenium:** Es una herramienta para la automatización de pruebas de interfaz de usuario (UI), que permite simular la interacción del usuario con la plataforma y validar la funcionalidad desde el punto de vista del usuario. Se utilizará para probar la experiencia del usuario en el sistema de mensajería, la interacción con el panel de administración, la creación de proyectos y ofertas, y el flujo de registro y autenticación de usuarios, asegurando que todas las acciones se realicen sin problemas.

**Jira:** Es la herramienta que estamos usando como base de nuestro proyecto para la gestión de este y seguimiento de defectos. Además de organizar el trabajo del equipo, facilita el control de calidad mediante la documentación y seguimiento de las pruebas y problemas encontrados. Jira se utilizará para planificar y coordinar las pruebas en todas las áreas del sistema, documentar cualquier fallo detectado en las pruebas de integración de los módulos de usuarios, proyectos, pagos, y mantener la comunicación continua sobre el estado de corrección de errores.

**Locust:** Es una herramienta de pruebas de rendimiento que simula el comportamiento de múltiples usuarios concurrentes para evaluar cómo responde el sistema bajo diferentes cargas de uso. Se aplicará para probar la capacidad de la plataforma para manejar el acceso simultáneo a las funcionalidades críticas, como la carga de proyectos, la pasarela de pagos, y el sistema de notificaciones, asegurando que el sistema mantenga un rendimiento óptimo incluso bajo alta demanda.

**AllPairs:** Es una herramienta de pruebas de combinación de pares que ayuda a reducir el número de casos de prueba necesarios, asegurando una cobertura adecuada de combinaciones de entradas sin redundancia. Utilizaremos AllPairs principalmente para pruebas de funcionalidades con múltiples combinaciones de entradas, como la configuración de opciones en los perfiles de freelancers, la personalización de notificaciones, y los filtros de búsqueda y listado de proyectos.

### **Tipos de pruebas a aplicar:**

#### *Pruebas de Humo:*

Descripción: Se utilizarán para verificar que las funciones críticas del sistema de freelancers estén operativas después de cada nueva compilación o integración. Estas pruebas rápidas se centrarán en la validación de las funciones esenciales como el registro y autenticación de usuarios, la carga de proyectos, y la integración con la pasarela de pagos,

asegurando que no haya errores evidentes que impidan el funcionamiento básico del sistema.

**Importancia:** Son esenciales para garantizar que la plataforma esté en un estado funcional antes de proceder a pruebas más detalladas. Validarán que no existan defectos graves que puedan interrumpir la experiencia del usuario.

**Estimación:** Tiempo corto, generalmente se ejecutan en pocas horas.

#### *Pruebas de Partición de Equivalencias y Análisis de Valores Frontera:*

**Descripción:** Estas pruebas se enfocarán en validar los inputs numéricos, de rango y límites en los formularios del sistema, como las ofertas en los proyectos, precios, y calificaciones de usuarios. Se verificarán las entradas válidas e inválidas y los valores límite para asegurarse de que el sistema maneje correctamente todas las posibles entradas del usuario. Estas nos permitirán verificar que los montos de las ofertas hechas por los freelancers estén dentro de los límites establecidos y que el sistema maneje correctamente valores mínimos y máximos permitidos.

**Importancia:** Son fundamentales para evitar errores en los campos de entrada del sistema, garantizando que las funciones como la oferta de proyectos y la gestión de pagos procesen correctamente todos los valores posibles, especialmente en situaciones de borde.

**Estimación:** Tiempo medio, dependiendo de la complejidad de los campos a validar.

#### *Pruebas de Transiciones de Estado:*

**Descripción:** Se evaluarán las transiciones entre diferentes estados del sistema, como el cambio de estado de un proyecto (por ejemplo, de "Pendiente" a "En progreso"), la actualización del perfil de un freelancer, o el cambio de estado de una transacción de pago. Estas pruebas asegurarán que las transiciones entre estados ocurran sin errores y de manera lógica.

**Importancia:** Son cruciales para verificar que la plataforma gestione adecuadamente los cambios en los estados de los usuarios, proyectos y pagos, evitando inconsistencias que podrían afectar la experiencia del usuario o la integridad de los datos.

**Estimación:** Tiempo medio, ya que implican el diseño y ejecución de múltiples casos de prueba para cada posible transición de estado.

### *Pruebas de Caja Negra:*

Descripción: Estas pruebas se centrarán en evaluar la funcionalidad del sistema sin considerar la estructura interna del código. Se basarán en los requisitos funcionales de la plataforma, como la gestión de usuarios, la creación de proyectos y la interacción en la pasarela de pagos, para verificar que el sistema se comporte correctamente bajo diversas condiciones de entrada.

Importancia: Estas pruebas son fundamentales para asegurar que la plataforma cumple con los requisitos definidos por los stakeholders y que las funcionalidades clave proporcionan una experiencia de usuario intuitiva y libre de errores.

Estimación: Tiempo medio, requiriendo la definición y ejecución de casos de prueba basados en los requerimientos del sistema.

### *Pruebas de Carga y Estrés:*

Descripción: Se llevarán a cabo para evaluar el rendimiento de la plataforma bajo condiciones de alta carga, simulando un gran número de usuarios accediendo simultáneamente a funciones críticas como el registro, la búsqueda y la oferta de proyectos. Las pruebas de estrés determinarán los límites del sistema bajo situaciones extremas para identificar puntos de fallo.

Importancia: Son cruciales para garantizar que la plataforma pueda manejar una alta cantidad de usuarios sin degradación significativa en el rendimiento, asegurando la estabilidad y velocidad de respuesta en escenarios de alto tráfico.

Estimación: Tiempo alto, debido a la complejidad de configurar y simular escenarios de alta carga.

### *Pruebas de Regresión:*

Descripción: Se ejecutarán para verificar que las funcionalidades o correcciones no afecten negativamente a las funcionalidades ya existentes del sistema. Estas pruebas asegurarán que el sistema sigue funcionando correctamente tras los cambios, enfocándose en las áreas

afectadas por las actualizaciones recientes, como la modificación de flujos de pagos o ajustes en la interfaz de usuario.

**Importancia:** Son esenciales para mantener la estabilidad del sistema y asegurar que los cambios realizados no introduzcan defectos en las áreas previamente probadas y aprobadas.

**Estimación:** Tiempo medio, ya que implica la revisión de casos de prueba ya existentes y la ejecución de estos tras cada cambio significativo en el sistema.

#### *Pruebas de Rendimiento:*

**Descripción:** Se enfocarán en medir la rapidez, capacidad de respuesta y estabilidad del sistema durante su uso normal y bajo condiciones de alta demanda. Estas pruebas incluirán la evaluación de tiempos de carga, procesamiento de pagos, y rendimiento en la gestión de múltiples transacciones concurrentes.

**Importancia:** Garantizan que la plataforma funcione de manera eficiente bajo diferentes niveles de carga, proporcionando una experiencia fluida y sin interrupciones para los usuarios, especialmente en momentos de alta actividad.

**Estimación:** Tiempo alto, dado que implica la configuración de entornos de prueba específicos y el uso de herramientas de simulación para medir el desempeño en escenarios reales.

#### *Pruebas de Compatibilidad:*

**Descripción:** Se llevarán a cabo para garantizar que la plataforma funcione correctamente en diferentes dispositivos, sistemas operativos y navegadores. Se probará la visualización y usabilidad en dispositivos móviles, tabletas y computadoras, así como la compatibilidad con navegadores como Chrome, Firefox, Safari y Edge.

**Importancia:** Aseguran una experiencia uniforme para todos los usuarios, sin importar el dispositivo o plataforma que utilicen, lo cual es vital para mantener la accesibilidad y funcionalidad del sistema en un entorno diverso.

**Estimación:** Tiempo medio, debido a la necesidad de probar múltiples combinaciones de dispositivos y entornos de usuario.



## Entregables del proceso:

### *Planeación*

**Matriz de descomposición funcional:** Un desglose de todas las funciones y subfunciones del sistema, organizado de manera jerárquica para asegurar que todos los aspectos funcionales estén cubiertos en las pruebas.

**Plan de pruebas:** Un documento que define el enfoque, recursos, calendario y actividades para las pruebas del proyecto. Incluye objetivos de prueba, criterios de entrada y salida, y los entregables esperados.

### *Diseño*

Matriz de casos de prueba diseñados por cada funcionalidad descrita en las historias de usuarios. Cada caso de prueba debe incluir pasos específicos, datos de prueba y los resultados esperados.

**Escenarios de Negocio para Pruebas End to End:** Descripciones detalladas de flujos de trabajo completos que simulan escenarios de negocio reales. Estas pruebas aseguran que el sistema funcione correctamente de principio a fin, integrando múltiples funcionalidades y sistemas externos.

Documentos adicionales como diagramas de flujo, tablas de decisión y listas de control que ayudan a validar las funcionalidades y garantizar que todos los aspectos del sistema sean probados exhaustivamente.

### *Ejecución*

**Soportes de ejecución:** Evidencias de la ejecución de pruebas almacenadas en un repositorio centralizado (Screenshots, logs de ejecución o cualquier tipo de evidencia que deje constancia de las pruebas realizadas).

Reportes o una documentación adecuada de bugs y la forma en que se manejaron, la herramienta que estamos usando para hacer este control será Jira. Se deben tener en cuenta detalles como la descripción del defecto, pasos para reproducirlo, severidad y estado actual.

### *Seguimiento*

**Informes de seguimiento de actividades:** Reportes regulares que detallan el progreso de las actividades de prueba, incluyendo métricas clave como la cantidad de casos de prueba ejecutados, defectos encontrados y resueltos, y cualquier bloqueo o riesgo identificado.

**Informe de facturación:** Detalle de los costos incurridos durante el proceso de pruebas, incluyendo una comparación con el presupuesto original y explicaciones de cualquier desviación.

### *Cierre*

**Informe consolidado del servicio:** Un informe final que resume todas las actividades de prueba, resultados obtenidos, métricas de calidad alcanzadas, y una evaluación general del cumplimiento de los objetivos del proyecto.

## **Mecanismos de seguimiento y control:**

### *Planificación y Seguimiento del Proyecto:*

Realizar reuniones semanales con el equipo de desarrollo y pruebas para revisar el progreso, identificar problemas o dificultades y planificar las acciones de la semana siguiente.

También definir objetivos clave del proyecto y realizar revisiones formales en cada uno de ellos para asegurar que se están cumpliendo y mantener unos plazos razonables y adecuados para el proyecto.

### *Respecto a pruebas*

- **Revisiones Diarias:** Realizar reuniones diarias de seguimiento para revisar el estado de las pruebas y los defectos.
- **Informes Semanales:** Generar informes semanales que incluyen los indicadores de pruebas y calidad, y compartirlos con todos los interesados.

### *Indicadores:*

**Indicadores de pruebas:** Medir y monitorear indicadores clave como la cantidad de casos de prueba ejecutados, la cantidad de casos de prueba correctos y fallidos, la densidad de errores, y la frecuencia de fallos.

**Indicadores de calidad del producto:** Evaluar la cantidad de defectos según su importancia (Alta, Media, Baja) y la cantidad de requerimientos elaborados y aprobados.

**Frecuencia de evaluación:** Los indicadores deben evaluarse en cada ciclo de pruebas y en los hitos clave del proyecto.

**Progreso:** Evaluar el progreso del proyecto y el estado de la calidad del software.

**Resolución problemas tempranos:** Detectar y resolver efectivamente problemas temprano en el ciclo de desarrollo.

**Nivel de optimización de procesos:** Utilizar los datos para identificar y clasificar áreas de mejora o fortalecen el proceso de desarrollo y pruebas.

**Retrospectivas de pruebas:** Después de cada ciclo de pruebas, realizar retrospectivas para discutir qué funcionó bien, qué no funcionó y cómo se pueden mejorar los procesos.

**Requerimientos elaborados:** Total de requerimientos con sus funcionalidades asociadas terminadas.

**Requerimientos a desarrollar o parcialmente desarrollados:** Total de requerimientos que están en proceso de elaboración o debido a otras funcionalidades se han completado de forma parcial.

**Correctitud:** Casos de prueba con resultado exitoso/Cantidad de casos de prueba ejecutados.(se espera  $\geq 90\%$ )

**Densidad de Defectos:** Cantidad de Defectos/Cantidad de casos ejecutados. (se espera  $\leq 10\%$ )

**Confiabilidad:**  $(1 - \text{Densidad NC}) \times 100$ , (se espera  $\geq 90\%$ )

**Cobertura:** Cantidad de casos ejecutados/Cantidad de casos de prueba planeados ejecutar. (se espera  $100\%$ )

**Alta Severidad:** Defectos críticos que afectan directamente la funcionalidad principal del sistema y requieren una resolución inmediata.

**Media Severidad:** Defectos importantes que afectan la funcionalidad del sistema, pero tienen soluciones temporales o workarounds.

**Baja Severidad:** Defectos menores que no afectan significativamente la funcionalidad del sistema y pueden ser corregidos en futuras versiones.

### **Alcance Funcional:**

Definiremos las características y funcionalidades del sistema que serán evaluadas durante el proceso de pruebas, garantizando que el sistema cumpla con los requisitos y expectativas de los usuarios y stakeholders. Para la plataforma web de servicios freelance, el alcance funcional incluye las siguientes áreas:

#### *Gestión de Usuarios:*

**Registro y Autenticación:** Verificación de los flujos de registro, inicio de sesión y recuperación de contraseñas. Asegurar que los usuarios puedan crear cuentas, autenticarse correctamente y gestionar sus credenciales de manera segura.

**Creación y Gestión de Perfiles:** Evaluación de la capacidad de los usuarios para crear, actualizar y visualizar sus perfiles. Incluye la validación de datos como biografía, experiencia profesional, habilidades y foto de perfil.

**Sistema de Mensajería:** Pruebas de la funcionalidad de mensajería interna para asegurar la correcta comunicación entre freelancers y clientes, incluyendo el envío, recepción, y notificación de mensajes.

#### *Gestión de Proyectos:*

**Listado y Creación de Proyectos:** Validación de la creación, visualización y edición de proyectos por parte de los clientes. Se asegurará que los proyectos puedan ser listados correctamente, incluyendo filtros y búsquedas.

Oferta de Servicios: Evaluación de la capacidad de los freelancers para hacer ofertas en los proyectos disponibles, asegurando que los flujos de oferta, negociación y aceptación funcionen sin problemas.

Seguimiento y Gestión de Proyectos: Pruebas del sistema de seguimiento de tareas y progreso del proyecto, incluyendo la capacidad de actualizar estados, añadir comentarios y cargar documentos relevantes.

#### *Sistema de Calificación y Reseñas:*

Calificación de Freelancers y Clientes: Verificación del sistema de calificación después de la finalización de los proyectos, asegurando que ambas partes puedan dejar valoraciones y comentarios.

Visualización de Reseñas: Pruebas para confirmar que las reseñas y calificaciones sean correctamente almacenadas y visibles en los perfiles de los usuarios, contribuyendo a la credibilidad del sistema.

#### *Pasarela de Pagos:*

Procesamiento de Pagos: Pruebas de integración con la pasarela de pagos, asegurando que los pagos se procesen correctamente, incluyendo depósitos, retiros y reembolsos.

Gestión de Transacciones: Validación de los registros de transacciones y la actualización del saldo de usuarios, garantizando la precisión y seguridad de los movimientos financieros.

#### *Sistema de Notificaciones:*

Notificaciones en Tiempo Real: Evaluación del sistema de notificaciones para asegurarse de que los usuarios reciben alertas relevantes sobre ofertas, mensajes, actualizaciones de proyectos y transacciones.

Configuración de Notificaciones: Pruebas de las opciones de configuración de notificaciones para que los usuarios puedan personalizar qué alertas desean recibir y en qué formato.

*Interfaz de Usuario (UI) y Experiencia de Usuario (UX):*

Consistencia y Navegación: Evaluación de la interfaz de usuario para garantizar una navegación intuitiva, una disposición clara de los elementos y una experiencia fluida a través de las diferentes secciones de la plataforma.

Accesibilidad: Pruebas para asegurar que la plataforma sea accesible para todos los usuarios, incluyendo la adaptación a dispositivos móviles y la compatibilidad con diferentes navegadores.

**Matriz de mapeo de funcionalidades:**

REQ	Proceso de negocio	Orden	Funcionalidades del sistema
1	Registro de empresa como freelancer	1	Solicitar información de registro para la empresa (datos de la empresa y representante legal).
		2	Validar los datos ingresados (nombre de la empresa, representante legal, NIT, etc.).
		3	Creación de cuenta de empresa freelance.
		4	Envío de correo de confirmación al representante legal.
	Registro de empresa como cliente	1	Solicitar información de registro para la empresa que necesita freelancers (datos de la empresa y representante legal).

		2	Validar los datos ingresados (nombre de la empresa, representante legal, NIT, etc.).
		3	Creación de cuenta de cliente para la empresa.
		4	Envío de correo de confirmación al representante legal.
	Registro de persona natural como freelancer	1	Solicitar información personal para el registro de un freelancer individual.
		2	Validar los datos ingresados (nombre completo, correo electrónico, cédula, etc).
		3	Creación de cuenta de freelancer individual.
		4	Envío de correo de confirmación al freelancer.
	Registro de persona natural como cliente.	1	Solicitar información personal para el registro de un cliente individual.
		2	Validar los datos ingresados (nombre completo, correo electrónico, cédula, etc.).
		3	Creación de cuenta de cliente individual.

	Autenticación de Usuarios	4	Envío de correo de confirmación al cliente.
		1	Solicitar credenciales (correo electrónico y contraseña).
		2	Validar credenciales contra la base de datos.
		3	Iniciar sesión del usuario si las credenciales son correctas.
		4	Mostrar mensajes de error en caso de datos incorrectos.
	Recuperación de Contraseña	1	Solicitar el correo electrónico registrado.
		2	Verificar la existencia del correo en la base de datos.
		3	Enviar enlace de restablecimiento de contraseña al correo proporcionado.
		4	Permitir la actualización de la contraseña tras la verificación del enlace.
2	Creación de Perfil de Freelancer	1	Solicitar información básica del freelancer (nombre, habilidades, experiencia, educación, etc).
		2	Validar la información ingresada.



		3	Guardar el perfil detallado del freelancer.
	Añadir imágenes	1	Cargar una imagen de perfil.
		2	Validar el formato y tamaño de la imagen de perfil.
		3	Cargar una imagen de portada.
		4	Validar el formato y tamaño de la imagen de portada.
		5	Mostrar las imágenes en el perfil del freelancer.
	Añadir Habilidades	1	Seleccionar la opción para añadir habilidades al perfil.
		2	Seleccionar habilidades desde una lista predefinida.
		3	Añadir habilidades personalizadas.
		4	Guardar y visualizar las habilidades en el perfil.
	Ingreso de Experiencia Laboral	1	Seleccionar la opción para añadir experiencia laboral.
		2	Ingresar los detalles de la experiencia laboral (empresa, puesto, fechas, referencias, descripción).

		3	Guardar y visualizar la experiencia en el perfil.
	Ingreso de Educación	1	Seleccionar la opción para añadir educación.
		2	Ingresar la información educativa (título, institución, país, fechas).
		3	Subir los certificados educativos.
		4	Guardar y visualizar la educación en el perfil.
	Ingreso de Portafolio	1	Seleccionar la opción para añadir proyectos al portafolio.
		2	Ingresar los detalles del proyecto (título, habilidades usadas, descripción, archivos de apoyo).
		3	Adjuntar archivos de apoyo para los proyectos.
		4	Guardar y visualizar el portafolio en el perfil.
	Sistema de calificación	1	Seleccionar la opción para calificar y comentar sobre un freelancer.
		2	Calificar con estrellas y dejar un comentario.

		3	Visualizar la calificación y responder a los comentarios si es necesario.
		4	Visualizar el promedio de las calificaciones.
	Ingreso de Enlaces Personales	1	Seleccionar la opción para añadir enlaces personales.
		2	Ingresar los enlaces a LinkedIn, GitHub y página web personal.
		3	Guardar y visualizar los enlaces en el perfil.
	3	Ingreso de Detalles del Proyecto	1
2			Ingresar el título, descripción, tipo de proyecto, habilidades requeridas, y ubicación del proyecto.
3			Guardar la información ingresada.
Selección del Tipo de Proyecto		1	Seleccionar si el proyecto es de "Rango de Presupuesto" o "Precio Fijo".
		2	Ingresar los valores correspondientes según el tipo seleccionado (rango de presupuesto o precio fijo).

		3	Validar la información ingresada.
	Publicación de proyecto	1	Seleccionar la opción de publicar proyecto.
		2	Validar que la información requerida esté correcta.
		3	Publicar proyecto y notificar a posibles interesados.
	Búsqueda de Proyectos por Tipo	1	Seleccionar el filtro de búsqueda por tipo de proyecto ("Rango de Presupuesto" o "Precio Fijo").
		2	Visualizar los proyectos según el tipo seleccionado.
	Búsqueda de Proyectos por Habilidades Requeridas	1	Seleccionar una o varias habilidades en los filtros de búsqueda.
		2	Visualizar los proyectos que requieran las habilidades seleccionadas.
	Búsqueda de Proyectos por Ubicación del Cliente	1	Seleccionar el país del cliente en los filtros de búsqueda.
		2	Visualizar los proyectos que correspondan a la ubicación seleccionada

	Ordenamiento de Proyectos por Reciente, Antiguo, Precio y Número de Ofertas	1	Seleccionar la opción para ordenar por "Más Reciente", "Más Antiguo", "Menor Precio", "Mayor Precio", "Menos Ofertas" o "Mayor Número de Ofertas".
		2	Visualizar los proyectos ordenados según la opción seleccionada.
4	Iniciar Chat desde el Perfil	1	Seleccionar el perfil de un freelancer o cliente desde la lista o búsqueda de perfiles.
		2	Clicar en el botón o enlace para iniciar un chat (por ejemplo, "Iniciar Chat" o un icono de mensaje).
		3	Verificar que se abre una nueva ventana o pestaña de chat en tiempo real con la persona seleccionada.
	Enviar Mensajes en Tiempo Real	1	Escribir un mensaje en el campo de texto disponible en la ventana de chat.
		2	Clicar en el botón de enviar (por ejemplo, "Enviar") o presionar Enter para enviar el mensaje.
		3	Verificar que el mensaje se muestra en la ventana de chat de ambos participantes al instante.
	Notificaciones de Nuevos Mensajes	1	Configurar el sistema para enviar una notificación al usuario cuando reciba un nuevo mensaje en un chat activo.

		2	Verificar que la notificación incluya un resumen del mensaje o un indicador de la llegada de un nuevo mensaje.
5	Visualización de la Línea de Tiempo del Proyecto	1	Acceder a la sección de proyectos.
		2	Elegir el proyecto específico para visualizar.
		3	Mostrar Línea de Tiempo con fechas de inicio y finalización.
		4	Seleccionar un hito o tarea en la línea de tiempo.
		5	Visualizar detalles asociados a ese hito, incluyendo estatus, fechas, subtareas y comentarios.
	Creación y Propuesta de Roadmap por el Freelancer	1	Ir a la sección de gestión de proyectos activos
		2	Seleccionar la opción para crear un roadmap.
		3	Ingresar nombre, descripción y fechas de inicio y finalización para cada hito.
		4	Guardar y Enviar Roadmap.
		5	Notificar Recepción de Roadmap.

	Revisión y Retroalimentación del Cliente	1	Acceder a la sección de revisión del roadmap.
		2	Ver el roadmap con hitos y tareas detalladas.
		3	Agregar comentarios o sugerencias de retroalimentación.
		4	Aprobar o rechazar el roadmap.
	Ajustes al Roadmap por el Freelancer	1	Recibir notificación de retroalimentación del cliente.
		2	Revisar los comentarios recibidos.
		3	Hacer los ajustes necesarios en el roadmap.
		4	Guardar y enviar el roadmap revisado al cliente para revisión final.
	Gestión de Hitos y Tareas en el Roadmap	1	Seleccionar un hito en el roadmap.
		2	Cargar el entregable correspondiente antes de la fecha estipulada.
		3	Actualizar el estatus del hito (Ej. En Progreso, Completado, Retrasado).

		4	Seleccionar una tarea dentro de un hito.
		5	Modificar los detalles de la tarea.
		6	Reflejar los cambios en el roadmap y la línea de tiempo del proyecto.
6	Procesamiento de Pagos	1	Seleccionar la Opción de Realizar un Pago.
		2	El sistema debe redirigir al usuario a una pasarela de pagos segura.
		3	El sistema debe actualizar el saldo o deuda del usuario en la plataforma tras la transacción.
	Lista de Pagos Pendientes (Eventos)	1	Ir a la sección "Recibos y Pagos" en la plataforma.
		2	Ver la lista de pagos pendientes (eventos).
		3	Cada evento debe mostrar: descripción, usuario al que se le hace el pago, valor del concepto, fecha de vencimiento y fecha de facturación.



	Pago de Salario a Freelancer	1	Elegir el evento correspondiente al salario del freelancer.
		2	Ver los detalles del freelancer (nombre, ID, número de cuenta).
		3	El sistema debe redirigir al usuario a la plataforma de pagos.
		4	El sistema debe actualizar el evento como "pagado" en la plataforma.
7	Envío de Notificaciones de Pagos, Mensajes y Publicación o Avance de Proyectos	1	El usuario realiza un pago exitoso en la pasarela de pagos.
		2	La pasarela de pagos confirma la transacción y actualiza la plataforma.
		3	El sistema muestra una notificación en la pantalla del usuario informando del pago exitoso.
	Direccionamiento por Medio de las Notificaciones)	1	Notificación push o correo con un enlace a un evento importante.
		2	El usuario hace clic en la notificación o enlace.

	Personalización de las Notificaciones y horas silenciosas	3	Redirigir al usuario a la sección específica (chat, proyecto, pagos) de la plataforma.
		1	Acceder a la sección de configuración de notificaciones en la cuenta del usuario.
		2	Modificar los parámetros de notificaciones push.
		3	Cambiar los horarios donde no se recibirán notificaciones push.
		4	Validar los cambios seleccionados.
8	Postulación a Proyecto con Pago por Horas	1	Seleccionar un proyecto con pago por horas.
		2	Ingresar su tarifa por hora, el número de horas límite a trabajar semanalmente y una descripción de la oferta.
		3	El freelancer envía la oferta al cliente.
		4	El sistema confirma el envío de la oferta y la guarda en la base de datos asociada al proyecto.

	Postulación a Proyecto con Presupuesto Fijo	1	Seleccionar un proyecto con presupuesto fijo.
		2	El freelancer ingresa el valor del presupuesto, la fecha de entrega del proyecto y una descripción de la oferta.
		3	Validar los datos ingresados.
		4	Confirmar el envío de la oferta y la guarda en la base de datos asociada al proyecto.
	Aceptación o Rechazo de Oferta	1	El cliente revisa una oferta para un proyecto específico.
		2	El cliente decide aceptar o rechazar la oferta.
		3	Validar esta respuesta.
		4	Si el cliente rechaza la oferta, tiene la opción de enviar un mensaje de respuesta al freelancer.

#### Bibliografía:

<https://es.parasoft.com/learning-center/application-security-testing-guide/>

<https://jmpovedar.wordpress.com/wp-content/uploads/2014/03/ejemplo-plan-de-pruebas.pdf>