

Estrategia de Pruebas

1. Aplicación Bajo Pruebas

1.1. Nombre Aplicación: Ghost

1.2. Versión: 5.96.1

1.3. Descripción:

Ghost es una plataforma de blogs de código de abierto diseñada para crear, diseñar y distribuir contenido de forma sencilla. La plataforma proporciona una infraestructura minimalista enfocada en la velocidad y facilidad de uso lo cual la convierte en un buen candidato para blogueros individuales o pequeños grupos editoriales.

1.4. Funcionalidades Core:

- **Creación de Publicaciones:** Permite a los usuarios crear, editar y publicar entradas de blog con contenido personalizado, imágenes, y etiquetas.
- **Gestión de suscripciones:** Permite a los administradores manejar las suscripciones de los usuarios que desean recibir contenido exclusivo o notificaciones.
- **Editor de Contenido:** Proporciona un editor fácil de usar que permite agregar texto, imágenes, videos y otros elementos a las publicaciones del blog.
- **Temas Personalizables:** Los usuarios pueden instalar y personalizar temas para adaptar el diseño del blog a su estilo o marca.
- **Gestión de Usuarios:** Permite a los administradores gestionar roles y permisos de diferentes usuarios que colaboran en el contenido (editores, escritores, administradores).
- **Búsqueda de Contenidos:** Incluye una barra de búsqueda para que los usuarios puedan encontrar fácilmente publicaciones específicas en el blog, ayudando a mejorar la navegación y experiencia del lector.
- **Crear Cuenta de Administrador:** Funcionalidad que permite crear una cuenta de administrador al iniciar Ghost por primera vez, donde se establece el nombre, correo electrónico y contraseña para el control total del sitio.
- **Inicio de Sesión:** Permite a los administradores y colaboradores acceder al panel de control de Ghost con su correo electrónico y contraseña para gestionar el contenido y las configuraciones del sitio.
- **Gestionar Parámetros de configuración:** Proporciona toda la información útil y algunas configuraciones del sitio, que pueden ser establecidas por el usuario.
- **Importar y Exportar:** Facilita la exportación de contenido como respaldo y la importación de datos en caso de restauración o migración de información.

1.5. Diagrama de Arquitectura:

(Archivo: Diagrama de arquitectura.png)

1.6. Diagrama de Contexto:

(Archivo: Diagrama de contexto.png)

1.7. Modelo de Datos:

(Archivo: Modelo Dominio Ghost.png)

1.8. Modelo de GUI:

(Archivo: Modelo GUI.png)

2. Contexto de la estrategia de pruebas

2.1. Objetivos:

- Fortalecer los conocimientos del equipo de trabajo interno en la aplicación de herramientas de automatización de pruebas y su configuración.
- Aplicar las técnicas de pruebas para evaluar el correcto funcionamiento de la aplicación en las funcionalidades identificadas de manera independiente.
- Comprobar el funcionamiento de la herramienta en el proceso de creación de publicaciones, la gestión de suscripciones y de usuarios y su integración.
- Realizar diferentes tipos de prueba como lo son pruebas manuales, pruebas de reconocimiento, pruebas E2E, VRT y escenarios de validación de datos con el objetivo de cubrir la mayor cantidad de funcionalidades del sistema.
- Planificar sesiones de pruebas exploratorias para descubrir defectos o áreas de mejora que posiblemente no estén cubiertos con otros casos de prueba.
- Realizar pruebas de rendimiento con diferentes niveles de carga para asegurar un sistema robusto que pueda escalar sin interrupciones.

2.2. Duración de la iteración de pruebas:

Período de Ejecución: 8 semanas con dedicación de 8 horas por semana

Fecha de Inicio: 2 de diciembre de 2024.

Fecha de Fin: 24 de enero de 2025.

Distribución Semanal de Actividades:

SEMANA	FECHAS	FASE	N° PERSONAS
1	Del 02/12/2024 al 06/12/2024	Fase 1 – Configuración inicial de herramientas, casos y entornos de pruebas. Se utiliza RIPuppet como herramienta de exploración.	4
2	Del 09/12/2024 al 13/12/2024	Fase 2 – Ejecutar pruebas de reconocimiento, pruebas manuales y validación de datos en funcionalidades clave.	4
3	Del 16/12/2024 al 20/12/2024	Fase 3 – Ejecutar pruebas E2E en flujos críticos y revisar resultados de pruebas previas.	4

Plantilla elaborada por

THE SW DESIGN LAB

4	Del 23/12/2024 al 27/12/2024	Fase 4 – Ejecutar pruebas E2E en flujos críticos y revisar resultados de pruebas previas.	4
5	Del 30/12/2024 al 03/01/2025	Fase 5 - Realizar pruebas de regresión visual automatizadas y documentar defectos visuales.	4
6	Del 06/01/2025 al 10/01/2025	Fase 6 – Realizar validación de datos y ajuste de casos de prueba.	4
7	Del 13/01/2025 al 17/01/2025	Fase 7 - Ejecutar pruebas E2E en funcionalidades no críticas y revisar resultados de pruebas previa.	4
8	Del 20/01/2025 al 24/01/2025	Fase 8 – Preparar informe final con hallazgos, recomendaciones y cerrar pruebas.	4

2.3. Presupuesto de pruebas:

2.3.1. Recursos Humanos

Se requiere un equipo de 4 ingenieros de sistemas con experiencia en programación y pruebas manuales, quienes serán responsables de diseñar, configurar y ejecutar diversos escenarios de prueba. Además, deberán documentar y presentar informes detallados sobre los hallazgos obtenidos durante el proceso de pruebas.

2.3.2. Recursos Computacionales

Para la ejecución de la presente estrategia de pruebas, es necesario que cada ingeniero cuente con su PC personal, en donde desarrollará las diferentes actividades. Para esto, a continuación se presenta la referencia de los PC con los que deben contar los testers.

- **MacBook Pro 13" (2020) con chip Apple M1:**

Procesador: Chip Apple M1

Memoria RAM: 8 GB de memoria unificada.

Almacenamiento: SSD de 256 GB.

Pantalla de 13,3 pulgadas con resolución de 2560 x 1600 píxeles.

- **Lenovo ThinkPad X1 Carbon Gen 9**

Procesador: Intel Core i5-1135G7 (Quad-core, 2.4 GHz base, hasta 4.2 GHz en turbo boost).

Memoria RAM: 8 GB de memoria LPDDR4x (configurable a 16 GB o 32 GB).

Almacenamiento: SSD de 256 GB (configurable a 512 GB, 1 TB o 2 TB).

Pantalla: 14" (1920 x 1080 píxeles)

Adicionalmente, cada equipo debe tener instaladas las siguientes aplicaciones:

- Node v20.18.0
- Cypress v10.8.2
- Google Chrome v131.0.6778.86
- Ghost-CLI v1.26.1
- Ghost v5.101.2
- Herramienta de grabación de video

- BackstopJS

2.3.3. Recursos Económicos para la contratación de servicios/personal:

No se contempla presupuesto para la contratación de personal ni servicios de terceros, por lo que el enfoque estará en maximizar el uso de los recursos disponibles.

2.4. TNT (Técnicas, Niveles y Tipos) de pruebas:

A continuación, se presenta un cuadro detallado que especifica las **Técnicas, Niveles y Tipos de Pruebas (TNT)** seleccionados para esta fase de pruebas. Cada técnica ha sido escogida en función de los objetivos previamente establecidos permitiendo asegurar que cada una contribuye a los resultados esperados y maximiza el uso eficiente de los recursos disponibles.

NIVEL	TIPO	TÉCNICA	OBJETIVO
Unidad	<ul style="list-style-type: none"> • Funcional • Caja negra • Enfoque positivo 	Pruebas manuales	Llevar a cabo prueba manuales sobre las funcionalidades críticas de la aplicación.
Unidad	<ul style="list-style-type: none"> • Funcional • Caja negra • Enfoque positivo 	Pruebas Automatizadas Guiadas por Script. Se utilizará Cypress	Llevar a cabo pruebas detalladas de unidad para verificar las diferentes funciones principales de Ghost.
Sistema	<ul style="list-style-type: none"> • Exploratorias. • Enfoque negativo 	Exploradores Sistemáticos mediante RIPuppet	Planificar sesiones de pruebas exploratorias para descubrir defectos o áreas de mejora que posiblemente no estén cubiertos con otros casos de prueba
Unidad	<ul style="list-style-type: none"> • Funcional • Enfoque negativo 	Pruebas de regresión visual	Garantizar la integridad visual de la aplicación mediante pruebas de regresión visual, detectando y corrigiendo cambios no deseados en la interfaz de usuario.

2.5. Distribución de Esfuerzo

Para cada una de las fases con las que cuenta la presente estrategia, se presenta la siguiente distribución de esfuerzo, que tiene como objetivo abarcar las pruebas de la mayor cantidad de funcionalidades, dentro del tiempo de dedicación está incluido la autocapacitación para el uso de herramientas y creación de casos de prueba, según apliquen para cada fase.

Distribución de Esfuerzo: Semana 1

Fase	Horas día 1	Horas día 2	Horas día 3	Horas día 4	Horas día 5	Total de horas semana 1
1 – Configuración inicial de herramientas, casos y entornos de pruebas. Se utiliza RIPuppet como herramienta de exploración.	-	12	-	12	8	32
Distribución de Esfuerzo: Semana 2						
Fase	Horas día 1	Horas día 2	Horas día 3	Horas día 4	Horas día 5	Total de horas semana 1
2 – Ejecutar pruebas de reconocimiento, pruebas manuales y validación de datos en funcionalidades clave	-	12	-	12	8	32
Distribución de Esfuerzo: Semana 3						
Fase	Horas día 1	Horas día 2	Horas día 3	Horas día 4	Horas día 5	Total de horas semana 1
3 – Ejecutar pruebas E2E en flujos críticos y revisar resultados de pruebas previas.	6	6	6	6	8	32
Distribución de Esfuerzo: Semana 4						
Fase	Horas día 1	Horas día 2	Horas día 3	Horas día 4	Horas día 5	Total de horas semana 1
4 – Ejecutar pruebas E2E en flujos críticos y revisar resultados de pruebas previas.	6	6	6	6	8	32
Distribución de Esfuerzo: Semana 5						
Fase	Horas día 1	Horas día 2	Horas día 3	Horas día 4	Horas día 5	Total de horas semana 1
5 - Realizar pruebas de regresión visual automatizadas y documentar defectos visuales.	6	6	6	6	8	32
Distribución de Esfuerzo: Semana 6						
Fase	Horas día 1	Horas día 2	Horas día 3	Horas día 4	Horas día 5	Total de horas semana 1
6 – Realizar validación de datos y ajuste de casos de prueba.	-	12	-	12	8	32
Distribución de Esfuerzo: Semana 7						
Fase	Horas día 1	Horas día 2	Horas día 3	Horas día 4	Horas día 5	Total de horas semana 1

7 - Ejecutar pruebas E2E en funcionalidades no críticas y revisar resultados de pruebas previa.	-	12	-	12	8	32
Distribución de Esfuerzo: Semana 8						
Fase	Horas día 1	Horas día 2	Horas día 3	Horas día 4	Horas día 5	Total de horas semana 1
8 – Preparar informe final con hallazgos, recomendaciones y cerrar pruebas.	-	-	-	15	17	32

Nota: El número de casos de prueba será incremental de la semana 1 a la semana 7. Esto con el objetivo de cubrir la mayor cantidad de funcionalidades.

El objetivo de estas fases es garantizar la calidad del software mediante un proceso estructurado de pruebas que abarca desde la definición de estrategias y herramientas hasta la validación final del sistema. Este enfoque incluye la planificación, preparación de casos y entornos de prueba, ejecución de pruebas manuales y automatizadas en funcionalidades clave y no críticas, pruebas de regresión visual y validación de datos. A lo largo del proceso, se busca asegurar que todas las funcionalidades del sistema sean evaluadas adecuadamente, que los errores se identifiquen y documenten y que el producto final cumpla con los requisitos y estándares establecidos.