

Estrategia de Pruebas

Tabla de Contenido

1.	Aplicación Bajo Pruebas.....	2
1.1.	Nombre Aplicación	2
1.2.	Versión.....	2
1.3.	Descripción.....	2
1.4.	Funcionalidades Core	2
1.5.	Diagrama de Arquitectura	4
1.6.	Diagrama de Contexto	5
1.7.	Modelo de Datos	5
1.8.	Modelo de GUI	5
2.	Contexto de la estrategia de pruebas	6
2.1.	Objetivos.....	6
2.1.1.	Objetivo 01	6
2.1.2.	Objetivo 02	6
2.1.3.	Objetivo 03	6
2.1.4.	Objetivo 04	6
2.1.5.	Objetivo 05	6
2.2.	Duración de la iteración de pruebas	7
2.3.	Presupuesto de pruebas	7
2.3.1.	Recursos Humanos	7
2.3.2.	Recursos Computacionales.....	7
2.3.3.	Recursos Económicos para la contratación de servicios/personal	7
2.4.	TNT (Técnicas, Niveles y Tipos) de pruebas	8
2.5.	Distribución de Esfuerzo	9
2.6.	Enlace video de la explicación de la estrategia de prueba:	11

1. Aplicación Bajo Pruebas

1.1. Nombre Aplicación

GHOST

1.2. Versión

Versión 3.41.1

1.3. Descripción

La aplicación Ghost es una plataforma web de creación de contenidos para que usuarios publiquen, compartan y tengan un crecimiento de su negocio. Sus principales herramientas permiten crear sitios web, publicar contenido, enviar notificaciones de contenido, suscripción de contenido y múltiples integraciones.

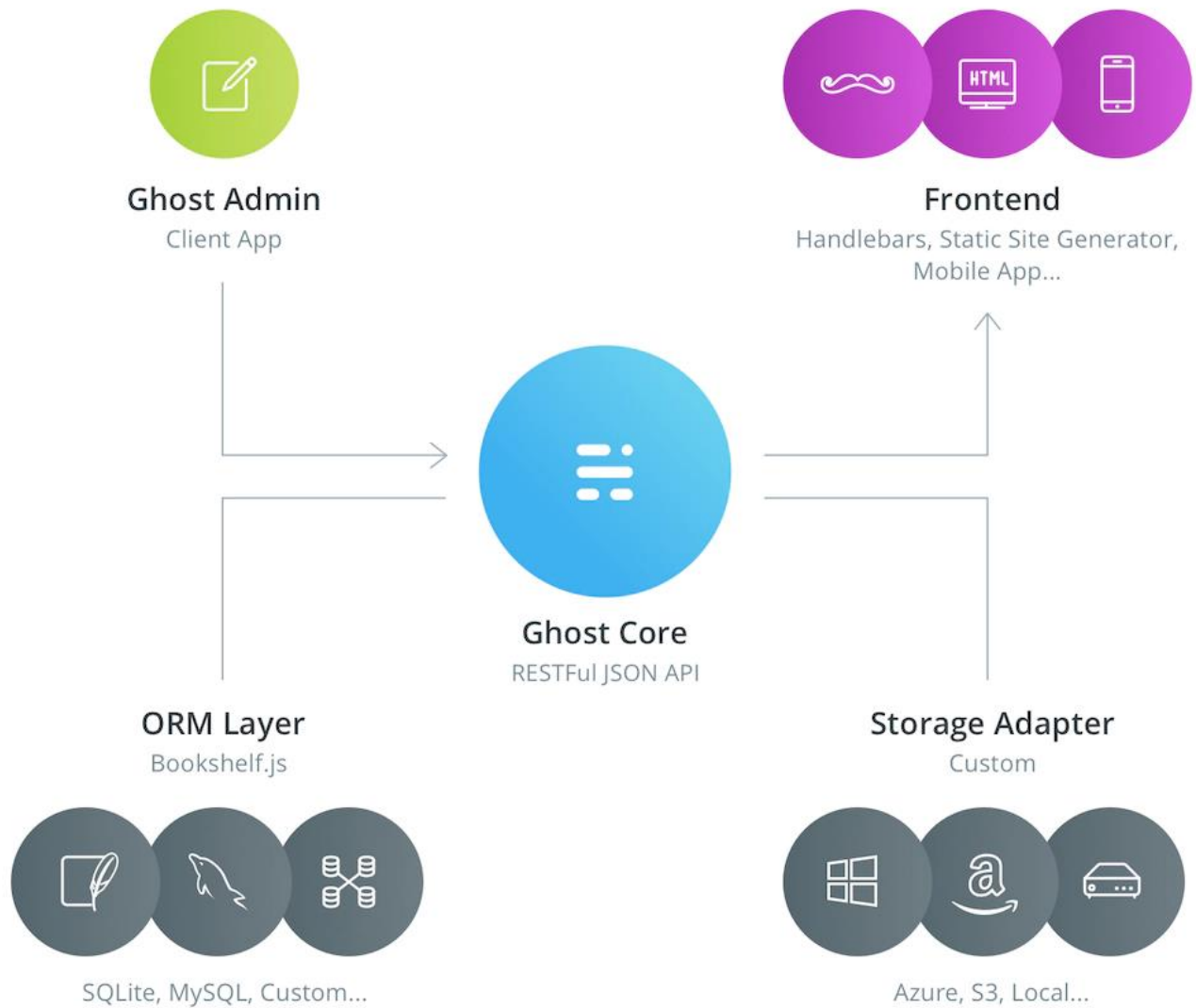
1.4. Funcionalidades Core

A continuación, se presenta grupo de funcionalidades principales de la aplicación.

Código	Nombre	Descripción
Fun-001	Registro de Usuarios e Inicio de sesión	Permite a los usuarios generar un registro para acceso a la plataforma de administración e Iniciar sesión para entrar al módulo de configuración
Fun-002	Administración de Staff users	Permite al administrador invitar staff y dar roles de autor, colaborador, administrador o editor.
Fun-003	Creación de tags para la página	Sistema de organización por etiquetas en el área administrativa y sitio web. Esta permite referenciar entre páginas, publicaciones según su categoría o segmentación de contenido
Fun-004	Creación de Paginas	Segmento para agrupación de contenido. Permite generar contenido para un mismo autor sobre diferentes segmentos agrupados según sus publicaciones de 'Página'.
Fun-005	Creación de Posts	Segmento para agrupación de contenido. Permite generar contenido para un mismo autor sobre diferentes segmentos agrupados según sus publicaciones de 'Posts'.

Fun-006	Administración y funciones de miembros	Administración de miembros de la plataforma de la página web en administración, funcionalidad debe ser activada desde la página de Laboratorio.
Fun-007	Administración Posts	Permite administrar el contenido publicado en la página, lo que está programado para ser posteo y lo que está en borrador
Fun-008	Configuración	Configuración de títulos y subtítulos para el sitio. Establecimiento de zona horaria para configuración de programación de contenido y configuración del lenguaje usado en el sitio. Puedes configurar Información de publicaciones, Metadata, cuentas sociales
Fun-009	Visualizar sitio web	Permite ver la visualización del sitio web con los diseños y contenidos generados
Fun-010	Laboratorio	Espacio para pruebas de características nuevas o experimentales. Tiene opciones como Miembros, opciones de migración y características betas
Fun-011	Perfil	Redireccionamiento hacia el módulo de Staff para configuración personal de administrador. Se encuentran las opciones para configuración del nombre completo, títulos en una página visible públicamente, correo, localización, sitio web, sitio de Facebook o Twitter, biografía, cambio de contraseña y token de acceso personal a usuarios sin necesidad de compartir usuario y contraseña.

1.5. Diagrama de Arquitectura



Tomado <<https://ghost.org/docs/architecture/>>

Ghost Core:

Ghost es un RESTful JSON API — diseñada para crear, manejar y mostrar los contenidos de las publicaciones. El API de Ghost está dividido en dos funciones principales: Contenido y Administración.

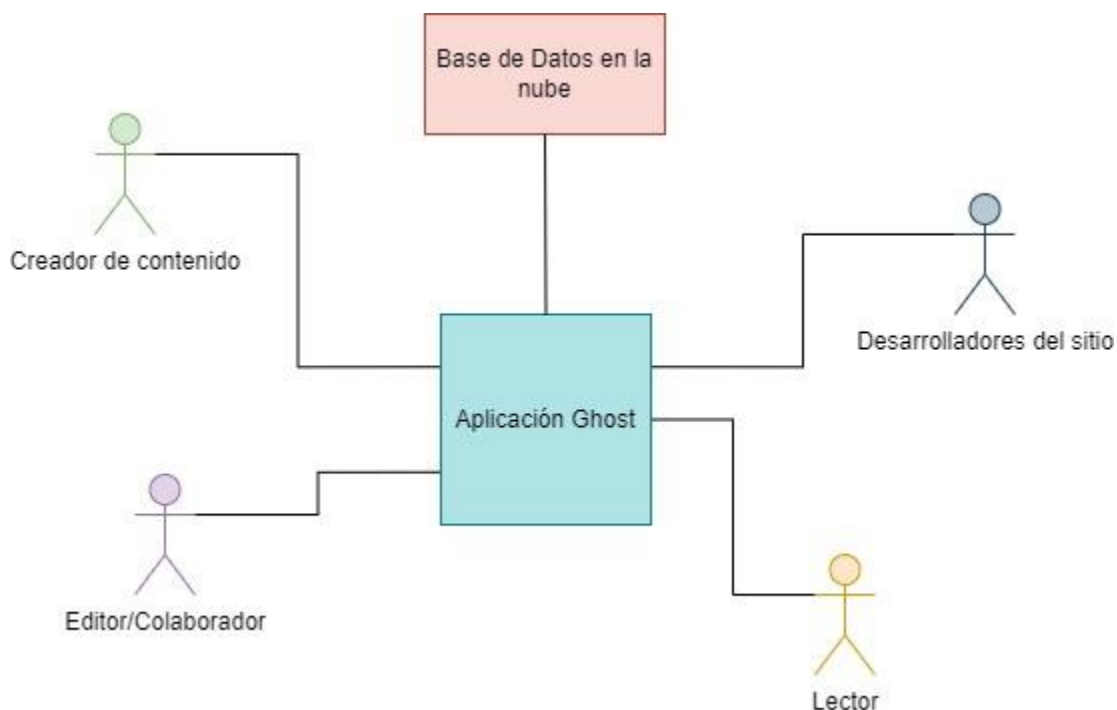
Admin Client:

Una interfaz de administración optimizada del lado del cliente para editores que necesitan una herramienta poderosa para administrar su contenido.

Front-End:

Ghost es un CMS completamente autónomo que es completamente independiente de cualquier interfaz de usuario en particular o marco de sitio estático.

1.6. Diagrama de Contexto



1.7. Modelo de Datos

En el siguiente link se puede ver el detalle:

<https://1drv.ms/u/s!AibLvPW5U5I3r-Vtw2UmgUeTZNg2vw?e=aqixsa>

1.8. Modelo de GUI

En el siguiente link se puede ver el detalle:

https://1drv.ms/u/s!AibLvPW5U5I3seNJP_FxidLizSX9JA?e=HwVGjc

2. Contexto de la estrategia de pruebas

2.1. Objetivo general

Definir la estrategia y los tipos de pruebas a utilizar en cada iteración, para garantizar una correcta implementación de las funcionalidades de la aplicación, enfocándose en la detección de defectos.

➤ **Objetivo 01**

Probar la parte funcional de la aplicación asegurándose que los flujos y funcionalidades principales tengan un correcto funcionamiento.

➤ **Objetivo 02**

Validar que el sistema funciona en la forma esperada en los siguientes browsers: Google Chrome, Edge, Firefox. A su vez validar compatibilidad y usabilidad de la aplicación en distintos tamaños de pantalla. (Large, Medium y Small).

➤ **Objetivo 03**

Verificar la escalabilidad y latencia de la aplicación ante múltiples solicitudes de consulta en la visualización de la página web creada.

➤ **Objetivo 04**

Detectar la mayor cantidad de defectos en la aplicación antes de ser publicada en producción.

➤ **Objetivo 05**

Utilizar distintos tipos de pruebas automatizadas, para obtener resultados desde varios ángulos de la aplicación.

2.2. Duración de la iteración de pruebas

Empieza 23 de mayo de 2022

Termina 17 de julio de 2022

2.3. Presupuesto de pruebas

2.3.1. Recursos Humanos

- Se cuenta 4 testers senior (8 horas/semana) (8 Semanas). Los cuales tienen experiencia en algunas de las herramientas a usar en la estrategia, pero deben aprender otras.

Personal	Disponibilidad	Experiencia
Tester 1	64h	Senior
Tester 2	64h	Senior
Tester 3	64h	Senior
Tester 4	64h	Senior

2.3.2. Recursos Computacionales

Hardware

- 4 computadores con sistemas operativo Linux (Intel Core i7 de 10gen, SSD 500GB, 8GB de RAM).

Los testers senior ya cuentan con estos recursos tecnológicos para trabajar en sus horas de dedicación a pruebas.

Recurso	Disponibilidad(Horas)
Computador 1	64 horas
Computador 2	64 horas
Computador 3	64 horas
Computador 4	64 horas

Software

Nombre	Precio	Tipo	Link
Cypress	\$0 (Open source)	E2E	https://www.cypress.io/
Puppeteer	\$0 (Open source)	Reconocimiento	https://pptr.dev/
Playwright	\$0 (Open source)	E2E	https://playwright.dev/
Kraken	\$0 (Open source)	E2E	https://kraken.ci/
Selenium	\$0 (Open source)	Record & Replay	https://www.selenium.dev/
Resemble JS	\$0 (Open source)	VRT	https://www.npmjs.com/package/resemblejs
Backstop JS	\$0 (Open source)	VRT	https://garris.github.io/BackstopJS/
Sapienz	\$0 (Open source)	Basadas en modelos	Sapienz

2.3.3. Recursos Monetarios

Se cuenta con un presupuesto total de 1000 USD para suplir las necesidades de recursos humanos y computacionales, los cuales se reparten de la siguiente manera:

- 64 horas de contratación de Ingeniero Senior = 960 USD*

TOTAL: 960 USD

* Se realizó una investigación en Upwork, plataforma para la contratación de freelancers y se escogió el siguiente precio promedio para el costo de los tester senior: Precio Tester Senior 15 USD / Hora
Referencia: <https://www.upwork.com/hire/qa-engineers/co/>

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

NIVEL	TIPO	TÉCNICA	OBJETIVO
Unitarias	Funcionales de caja negra, mixtas	Manual	Obj 01, Obj02, Obj04
Unitarias	Funcionales de caja blanca, positiva	Análisis estático del código	Obj01
Integración	Funcionales, caja negra, positivas	Record and Replay	Obj01, Obj02, Obj05
Sistema	Funcionales, caja negra, positiva y negativa	End To End	Obj01, Obj04
Sistema	Funcionales, caja negra, positiva y negativa	End To End con validación de datos	Obj01, Obj04

Sistema	Funcionales, caja negra, positiva y negativa	RIP Puppig	Obj01, Obj04
Sistema	No Funcionales, caja negra, positiva y negativa	Exploratorias	Obj 03, Obj04
Aceptación	Funcionales, caja negra, positiva y negativa	Regresión visual	Obj01, Obj02, Obj04, Obj05

Para Objetivo01

Se espera probar las funcionalidades core que están en la lista de funcionalidades. Para este objetivo se ha decidido utilizar diferentes técnicas, como son: Manuales, exploratorias, End to End, Record and Replay, Análisis estático del código y RIPuppig. Los testers deben realizar documentación de issues en la plataforma GitHub, al encontrar algún error en la ABP.

Para Objetivo02

Se busca que la aplicación tenga compatibilidad con distintos navegadores y tamaños de pantalla. Muchas de las herramientas que se van a utilizar en las técnicas ofrecen la posibilidad de realizar las pruebas en distintos navegadores y de cambiar el tamaño de la pantalla en la que se realizan las pruebas. Por lo tanto, al diseñar las pruebas se tendrá en cuenta la configuración de estos parámetros para su ejecución.

Para Objetivo03

Se realizan pruebas a nivel de aceptación, porque se está testeando requisitos de calidad del sistema, son pruebas no funcionales, de caja negra y positivas. Se escogió Monkey testing como técnica de automatización porque se considera que en función de los eventos aleatorios puede mandar muchos llamados para ver cómo se comporta a nivel de latencia el sistema. De igual manera con el monkey podríamos verificar como se distribuyen los recursos entre instancias de máquina para comprobar escalabilidad.

Para Objetivo 04

A lo largo del proyecto de pruebas, se irán documentando todos y cada uno de los defectos (issues) encontrados en la aplicación. Las herramientas principales para detectar defectos no reconocidos serán las pruebas manuales y las pruebas exploratorias. Para defectos de ingreso de datos (límites, fronteras y validación de datos), se utilizarán las pruebas E2E con generación de datos aleatorios.

Para Objetivo 05

Con la experiencia que tienen los testers acerca de distintos tipos de pruebas, se busca diversificar la exploración y búsqueda de límites de la aplicación puesta a prueba. Empezando por pruebas manuales, pasando por exploratorias, E2E, VRT y de reconocimiento. Lo anterior también incluye la utilización de herramientas actuales y reconocidas en la industria, como lo son Cypress y Selenium, además de utilizar otras en crecimiento como Kraken.

2.5. Distribución de Esfuerzo

La distribución de pruebas que se estima utilizar es la de pirámide automatización

Las pruebas serán realizadas al final del proyecto porque la aplicación ya está implementada.

La distribución de esfuerzo es la siguiente:

Distribución de esfuerzos de recursos humanos disponibles:

- Cuatro testers senior para todo el proyecto.
 - La primera semana será utilizada para aprender acerca de la aplicación a la cual se hacen pruebas, se hace reconocimiento del equipo y se inicia la realización de pruebas exploratorias (manuales) y E2E, dado que esta la manera más efectiva de poder comprender la aplicación a primera vista.
 - A partir de la segunda semana y hasta la semana cuatro, se crean y ejecutan pruebas E2E, validación de datos y de reconocimiento, además de una revisión estática del código, ya que ahora se tiene más conocimiento general de la aplicación.
 - De la semana cinco a la siete, con el despliegue de la nueva versión de Ghost, se hace refactoring de las pruebas ya creadas y también se inicia la creación y ejecución de las pruebas de regresión visual, para detectar cambios y mejoras entre versiones. Además, se utilizan pruebas de Record & Replay para detectar issues resueltos y aún fallando dentro de la nueva versión. También se realiza documentación de las pruebas VRT por medio de un reporte.
 - Finalmente, en la semana 8 se utilizan herramientas recientemente aprendidas para pruebas con modelos y así completar el cubrimiento de la aplicación. Se documentan todos los resultados listos para ser expuestos.
- Todas las semanas se agenda un espacio para hablar con el equipo acerca de avances y cualquier tipo de inquietud que haya.

- En el siguiente [link](#) se encuentra la distribución de esfuerzo por tester, por cada semana y hora laborada.

Distribución de esfuerzos de recursos tecnológicos disponibles:

- Los testers tienen disponibilidad del recurso tecnológico durante las horas hábiles de realización de las pruebas.

2.6. Enlace video de la explicación de la estrategia de prueba

[Estrategia de pruebas - Entrega Final - Grupo 23](#)

DISTRIBUCIÓN DE ESFUERZO

Tester senior					
Tarea	Horas	Semana	Tarea	Horas	Semana
Reconocimiento de aplicación a probar	5	1	Análisis preliminar del código	2	4
Preparación del ambiente de trabajo	5	1	Reunión semanal de seguimiento	1	4
Creación/inventario de pruebas manuales	2	1	Reconocimiento de nueva versión de la aplicación a probar	2	2
Documentación pruebas manuales	2	1	Relación de las pruebas E2E con pruebas con la misma versión	4	2
Documentación de casos de prueba	2	1	Reunión semanal de seguimiento	1	2
Creación pruebas E2E con Capella - Parte 1	5	1	Documentación de casos de prueba de integración	1	2
Creación y validación de pruebas E2E	5	1	Preparación de ambiente para con Docker / <i>Setup de pruebas</i>	2	4
Creación pruebas de reconocimiento en Capella (Modeling) / <i>Preparación de pruebas</i>	2	2	Relación de pruebas para incluir en la estrategia de pruebas	2	4
Validación pruebas de reconocimiento	5	2	Creación de pruebas E2E con Reversi / <i>Setup de pruebas</i>	2	4
Reunión semanal de seguimiento	1	2	Reunión semanal de seguimiento	1	4
Documentación de pruebas de aceptación	2	2	Documentación de casos de prueba de integración	1	4
Creación pruebas E2E con Capella / <i>Preparación de pruebas</i> - Parte 2	2	2	Preparación y documentación de ambiente de trabajo Selenium	2	2
Creación y validación pruebas E2E	5	2	Creación de pruebas E2E y Relación de pruebas manuales	4	2
Creación de estrategia de pruebas con Jira / <i>Modelos de prueba</i> / <i>Modelos de prueba</i>	4	2	Reunión semanal de seguimiento	1	2
Reunión semanal de seguimiento	2	2	Documentación de casos de prueba de integración	1	2
Documentación de casos de prueba de integración	2	2	Documentación de casos de prueba de integración	2	4
Reconocimiento de herramienta de prueba para E2E	5	4	Creación e implementación de pruebas Selenium	4	2
Creación de pruebas E2E con Selenium	4	4	Documentación de aplicación de resultados	2	4