

Estrategia de Pruebas – Semana 8

1. Aplicación Bajo Pruebas

1.1. Nombre Aplicación: Ghost

1.2. Versión: 5.70

1.3. Descripción:

Comercial: Ghost es una aplicación para que editores profesionales creen, compartan y hagan crecer un negocio en torno a su contenido. Viene con herramientas modernas para crear un sitio web, publicar contenido, enviar boletines y ofrecer suscripciones pagas a los miembros. Ghost tiene diferentes usuarios potenciales, entre ellos: creadores, publicadores de contenido, negocios y desarrolladores.

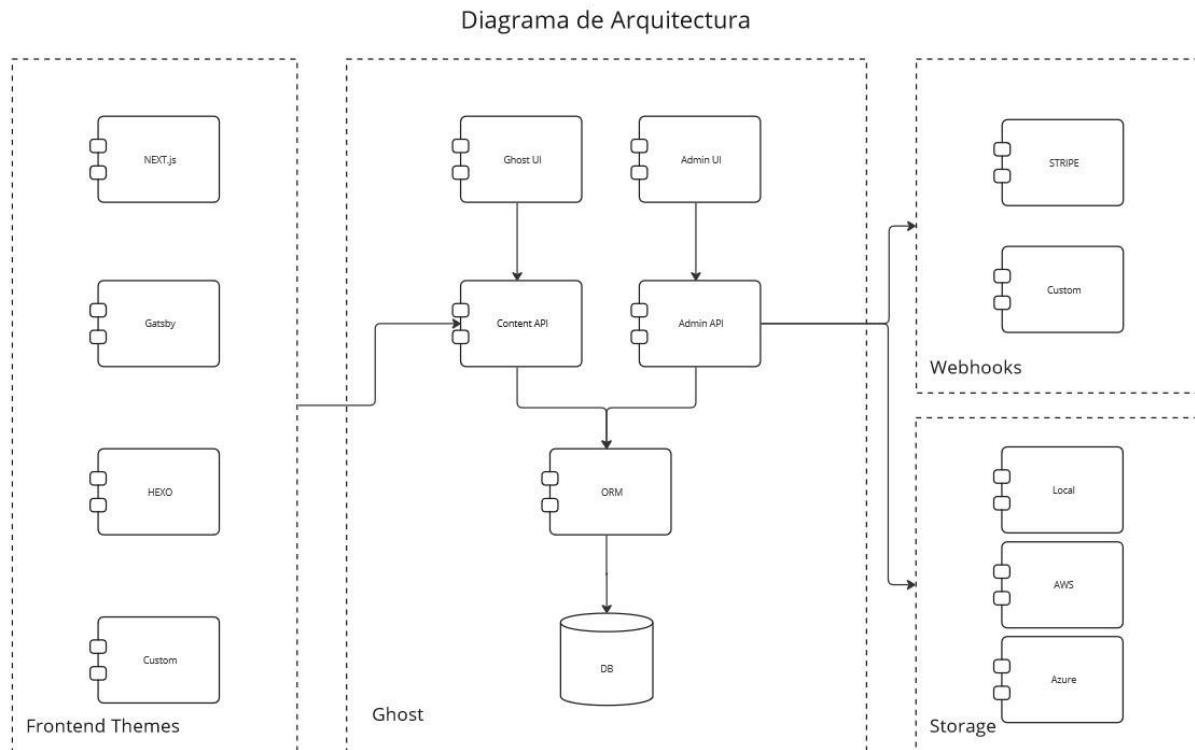
Técnica: Ghost es un gestor de contenido (Content Management System) para la creación de blogs desarrollado principalmente en JavaScript, TypeScript, HTML y CSS ¹.

1.4. Funcionalidades Core:

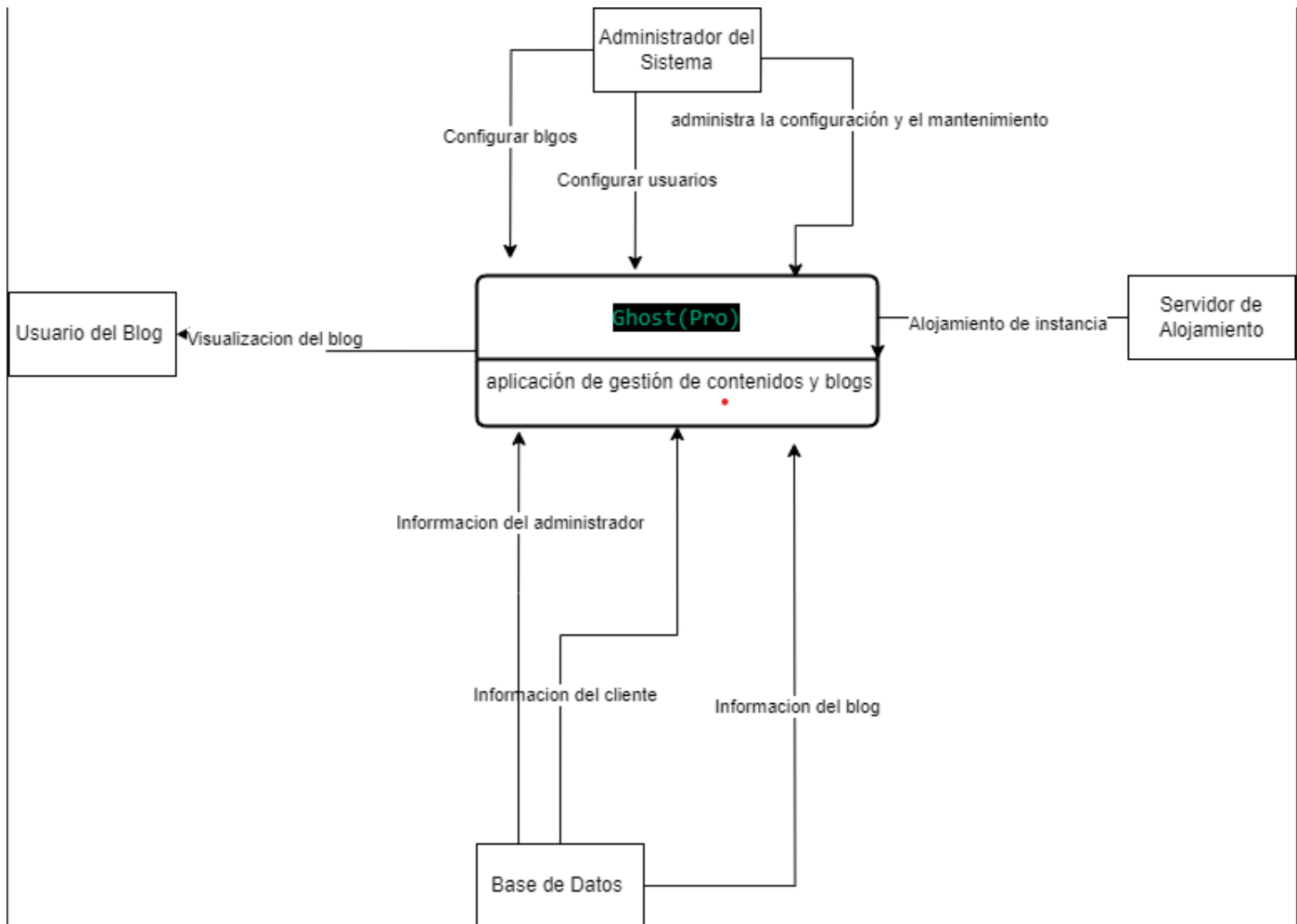
- Inicio de sesión / Autenticación
- Gestión de Páginas
- Gestión de entradas de Blog
- Gestión de Tags para organizar contenido
- Gestión de Miembros

¹ <https://github.com/TryGhost/Ghost>

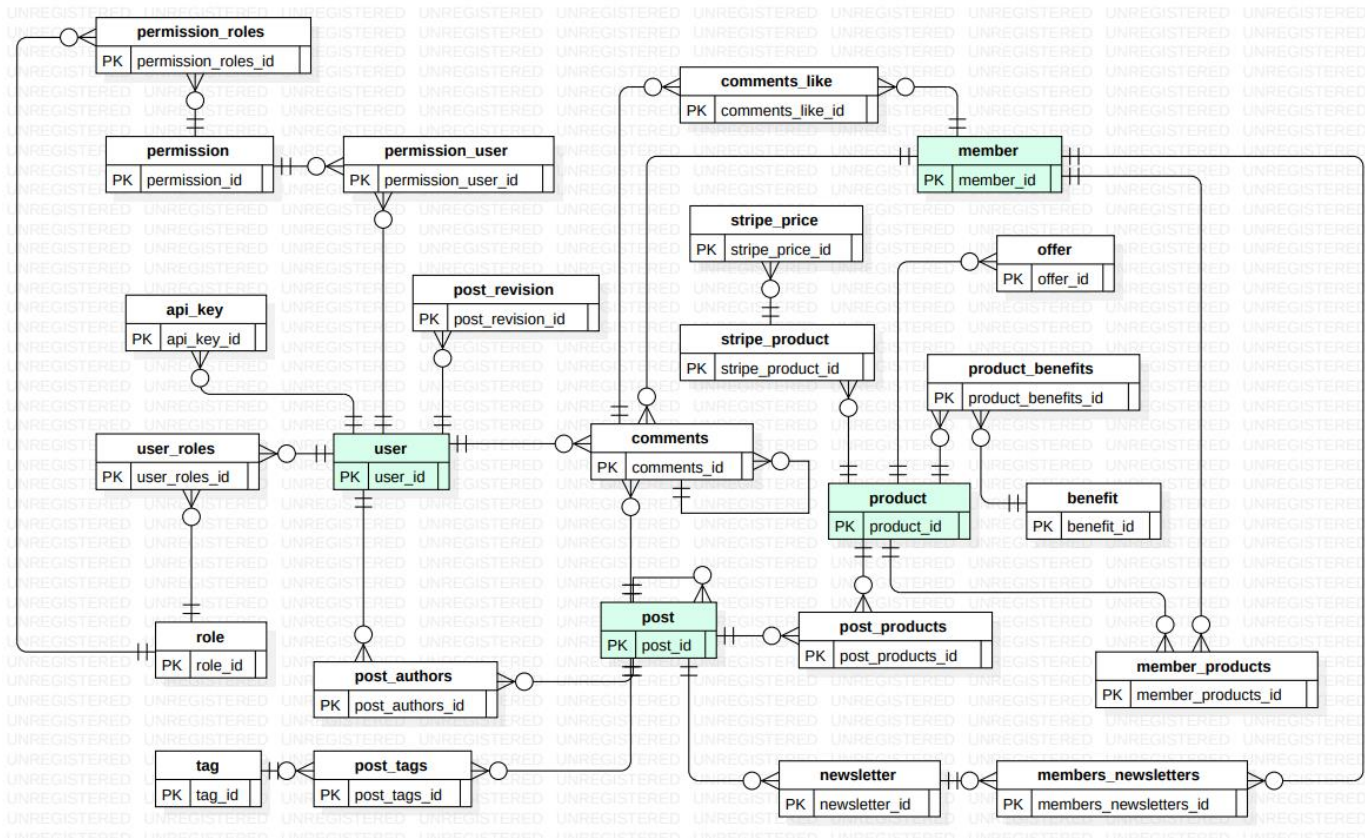
1.5. Diagrama de Arquitectura:



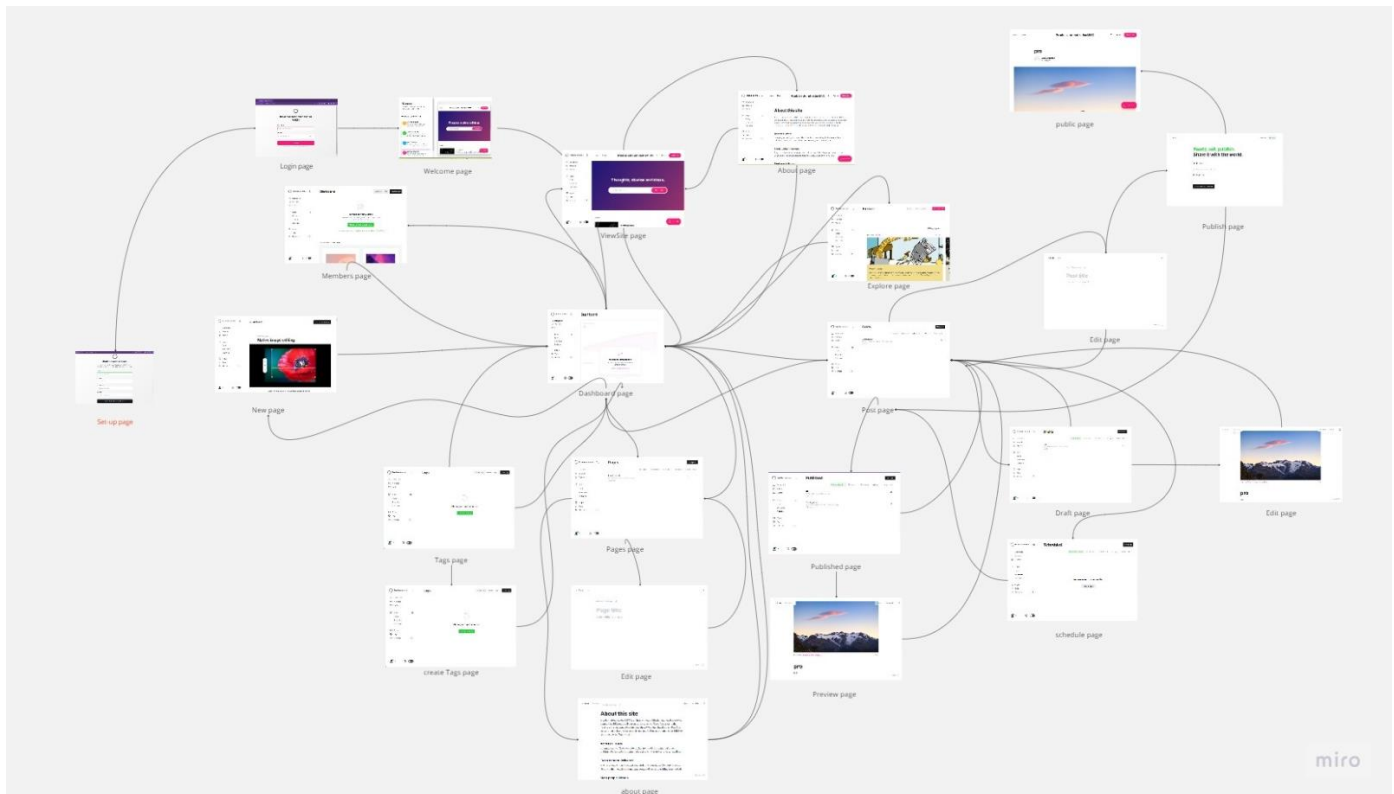
1.6. Diagrama de Contexto:



1.7. Modelo de Datos:



1.8. Modelo de GUI:



2. Contexto de la estrategia de pruebas

2.1. Objetivos:

- 2.1.1. Detectar y solucionar posibles errores de forma temprana con el fin de reducir costos por mantenimiento y solución de defectos.
- 2.1.2. Mejorar la eficiencia en la ejecución de pruebas y acelerar el proceso de desarrollo reduciendo los tiempos de entrega del software.
- 2.1.3. Asegurar un cubrimiento más eficiente de escenarios y funcionalidades provistas por la aplicación.
- 2.1.4. Aumentar la gama de casos de prueba para garantizar una mayor cobertura de funcionalidades y características ofrecidas en por el software.

2.2. Duración de la iteración de pruebas:

Se contempla iteraciones con Sprints de 1 semana, durante 8 semanas, 8 horas por semana para cada miembro del equipo de testeo.

2.3. Presupuesto de pruebas:

2.3.1. Recursos Humanos

Cuatro (4) testers senior (en este caso solo soy yo), durante 8 semanas, en una dedicación de 8 horas por persona semanalmente.

2.3.2. Recursos Computacionales

Ya que la compañía está floreciendo, ofrecemos el modelo BYOD (bring your own device) el cual permite a los empleados traer sus propios dispositivos al trabajo, si lo desean también se ofrecen laptops MacBook Pro (2018 Intel Core I7, 8 CPU, 16GB RAM) para el trabajo (carácter devolutivo), sin embargo, estas vienen con software ETS para la administración de software junto al contenido de los dispositivos y el monitoreo de los trabajadores, claramente, cubriendo los estándares legales.

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

Utilizando como referencia diferentes portales de contratación^{2 3}, se concluye que el rango salarial para la estrategia de pruebas estaría provisto de la siguiente manera:

Lower Bound: 19,687.5\$ COP la hora de trabajo (3'150,000\$ COP = mes laboral de 4 semanas, 5 dias de 8h la semana)

Upper Bound: 41,662.5\$ COP la hora de trabajo (6'666,000\$ COP = mes laboral de 4 semanas, 5 dias de 8h la semana)

² <https://co.talent.com/salary?job=senior+qa>

³ <https://co.talent.com/salary?job=senior+tester>

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

Nivel	Tipo	Técnica	Objetivo(s)
Pruebas de Sistema	Pruebas funcionales positivas de caja negra	Exploración sistemática (GUI Ripping)	2.1.3, 2.1.4
Pruebas de Sistema	Pruebas funcionales positivas de caja negra	Manual Testing	2.1.3
Pruebas de Sistema	Pruebas funcionales positivas y negativas de caja negra	Regresión Visual	2.1.3
Pruebas de Sistema	Pruebas funcionales positivas y negativas de caja negra	Escenarios de validación de datos	2.1.1, 2.1.3, 2.1.4
Pruebas de Aceptación	Pruebas funcionales positivas y negativas de caja negra	APIs de automatización (E2E)	2.1.1, 2.1.2, 2.1.3

Nótese que el esquema TNT contempla

- Pruebas manuales
- Pruebas de reconocimiento (por medio del uso de GUI Ripping)
- Pruebas de extremo a extremo
- Pruebas de regresión visual
- Escenarios de validación de datos

2.5. Distribución de Esfuerzo

Semana	Tareas a realizar
1	Primera entrega: <ul style="list-style-type: none">• Pruebas E2E sobre la interfaz (UI) de la aplicación (10 por cada tester).• Pruebas manuales sobre la interfaz (5 por tester).• Reporte de incidencias (al menos 5 por tester).
2	Segunda entrega: <ul style="list-style-type: none">• Pruebas de validación de datos (20 por tester).• Reporte de incidencias.
3	Tercera entrega: <ul style="list-style-type: none">• Pruebas de regresión visual sobre las pruebas de entregas anteriores, cada tester se encarga de sus pruebas.• Reporte de discrepancias entre versiones.
4	Cuarta entrega: <ul style="list-style-type: none">• Pruebas de exploración sistemática (uso de GUI Ripping) (30 pruebas por tester).• Pruebas manuales sobre la interfaz (5 por tester).

	<ul style="list-style-type: none"> • Reporte de incidencias.
5	<p>Quinta entrega:</p> <ul style="list-style-type: none"> • Pruebas E2E sobre la interfaz (UI) de la aplicación (10 por cada tester). • Pruebas de exploración sistemática (uso de GUI Ripping) (15 pruebas por tester). • Reporte de incidencias.
6	<p>Sexta entrega:</p> <ul style="list-style-type: none"> • Pruebas E2E sobre la interfaz (UI) de la aplicación (10 por cada tester). • Realización de 20 pruebas manuales por integrante. • Reporte de incidencias.
7	<p>Séptima entrega:</p> <ul style="list-style-type: none"> • Pruebas de validación de datos (20 por integrante). • Reporte de incidencias.
8	<p>Octava entrega:</p> <ul style="list-style-type: none"> • Pruebas de regresión visual sobre las pruebas de entregas anteriores. • Reporte de discrepancias entre versiones. • Reporte de incidencias acumulativo.