

# Estrategia de Pruebas para GHOST

Fecha:	30 de noviembre de 2024
Autores:	Calletana López Juan Tapia Raúl Ramos Sergio Andrés Gelves
Empresa:	The Software Design Company (TSDC)

## 1. Contenido

1	Aplicación bajo prueba .....	2
1.1	Nombre de la Aplicación: .....	2
1.2	Versión de la aplicación: .....	2
1.3	Descripción: .....	2
1.4	Listado de funcionalidades .....	2
1.5	Diagrama de la arquitectura .....	5
1.6	Diagrama de Contexto .....	6
1.7	Modelo de datos .....	6
1.8	Modelo de GUI .....	7
2	CONTEXTO DE LA ESTRATEGIA DE PRUEBAS .....	8
2.1	Objetivos: .....	8
2.2	Duración de la iteración de pruebas: .....	8
2.3	Presupuesto de pruebas: .....	9
2.3.1	Recursos Humanos: .....	9
2.3.2	Recursos Computacionales: .....	9
2.4	TNT (Técnicas, Niveles y Tipos) de Pruebas: .....	10
2.5	Distribución de Esfuerzo: .....	11
2.5.1	Primera Etapa: .....	12
2.5.2	Segunda Etapa: .....	12

## **1 APLICACIÓN BAJO PRUEBA**

### **1.1 Nombre de la Aplicación:**

Ghost

### **1.2 Versión de la aplicación:**

5.96.0

### **1.3 Descripción:**

Ghost es un sistema de gestión de contenidos (CMS) que se usa principalmente para crear blogs y sitios web enfocados en publicaciones. Es conocido por ser fácil de usar y por permitir que las personas publiquen y gestionen contenido de manera sencilla

### **1.4 Listado de funcionalidades**

Inicio

1. Iniciar sesión: inicia sesión en el módulo administrativo y muestra el dashboard de la aplicación.

Configuración General.

2. Configurar título y descripción: permite cambiar el título y la descripción del sistema

Staff:

3. Agregar miembro: permite invitar otras personas para participar en el sistema
4. Editar miembro: permite editar los datos de un colaborador

### Personalizar el diseño:

5. Personalizar marca: Permite cambiar descripción, color de botones y de fondo de la pagina
6. Personalizar ancho del sitio: permite cambiar el tipo de letra y ubicación del logotipo
7. Personalizar Home Page: permite cambiar la estructura de la página principal
8. Personalizar Post: permite activar opciones de personalización en la vista de los posts
9. Personalizar Navegación: Permite configurar los vínculos de navegación del menú principal
10. Modificar perfil: permite cambiar el correo y la contraseña entre otros datos del usuario que inicio sesión.

### Post

11. Agregar post: permite crear una nueva publicación en estado borrador.
12. Editar post: permite modificar los datos de un post existente
13. Eliminar post: permite eliminar un post publicado
14. Configurar post: permite modificar la configuración de la publicación del post
15. Publicar post: permite publicar el post en la pagina
16. Despublicar post: permite ocultar un post publicado, el cual no se vera en la pagina

### Paginas

17. Nueva página: permite agregar una nueva página en modo borrador.
18. Editar página: permite modificar la información de una página existente
19. Configurar página; permite modificar los datos como la url, fecha de publicación, tags, tipo de acceso, entre otros.
20. Eliminar página: permite eliminar una página existente
21. Publicar página: permite publicar una página en el sistema

22. Despublicar página: permite ocultar una página y volver al estado en borrador

#### Elementos web

23. Agregar imagen: permite agregar una imagen a una pagina

24. Agregar video: permite agregar un video a una pagina

25. Agregar producto: permite agregar una imagen con un título, botón y estrellas para calificación del producto

26. Agregar galería: permite agregar varias imágenes organizadas, máximo 9

#### Tags

27. Agregar tag: permite agregar tag para agrupar los posts publicados

28. Editar tag: modifica un tag existente

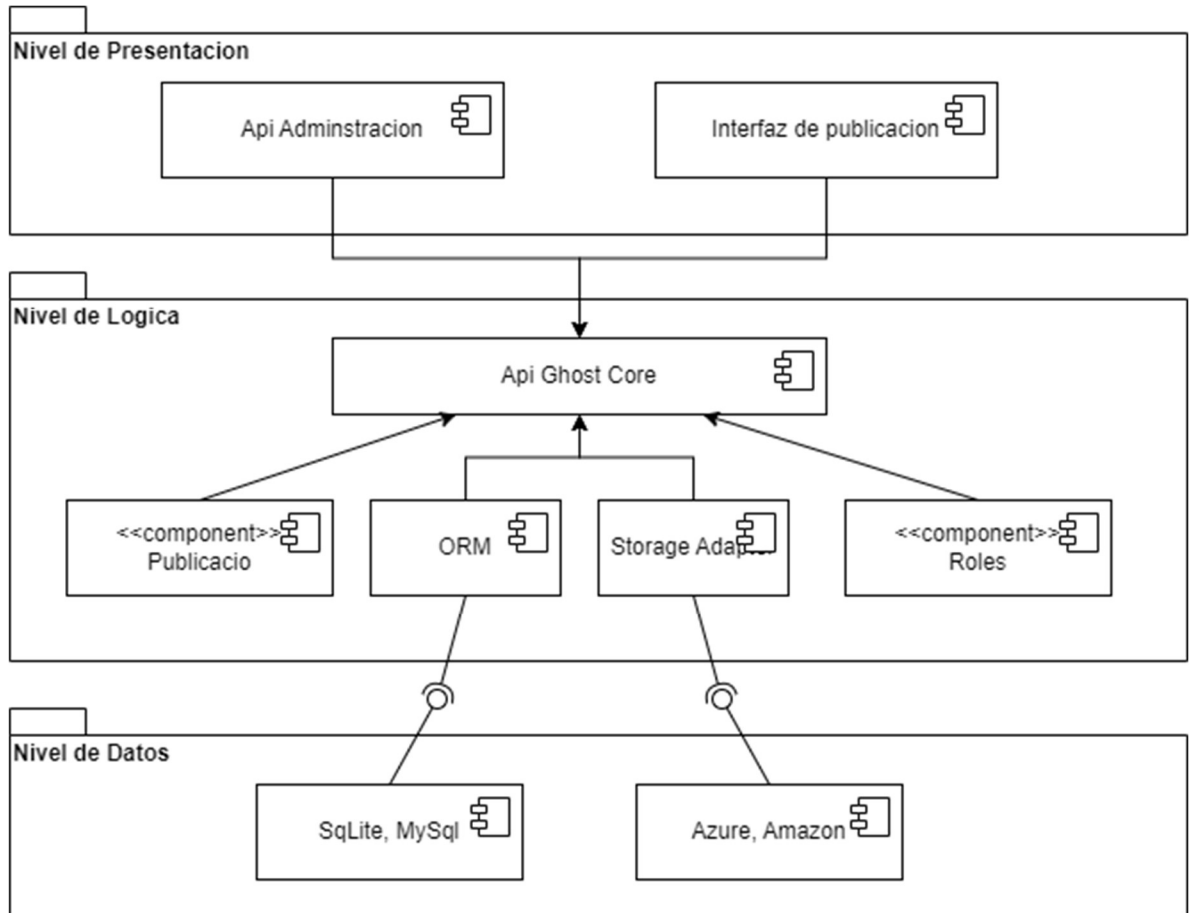
#### Suscriptores

29. Agregar suscriptor: permite agregar un suscriptor por medio del correo, este es una persona que tiene acceso a contenido exclusivo

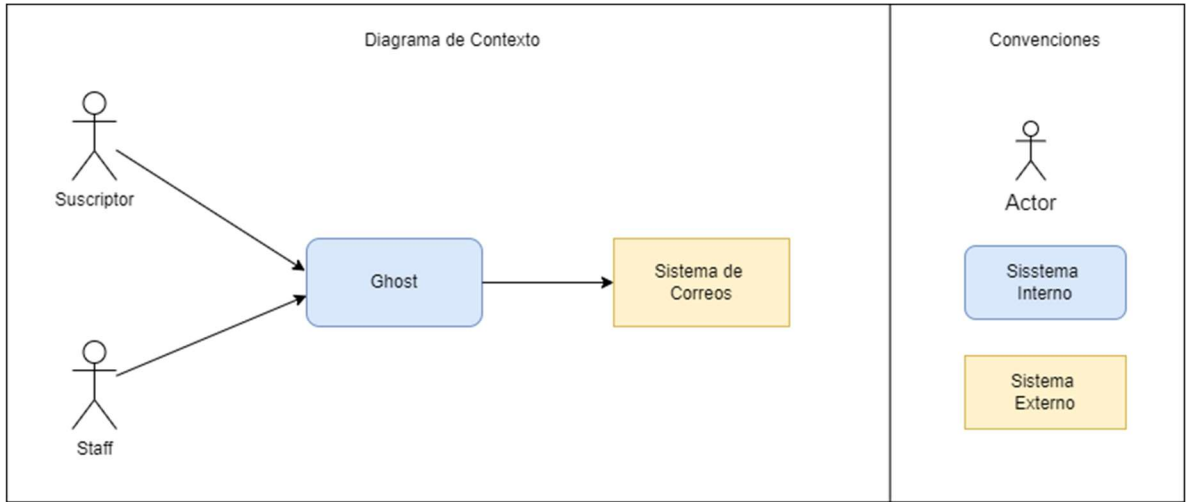
30. Editar suscriptor: Editar un suscriptor existente.

## 1.5 Diagrama de la arquitectura.

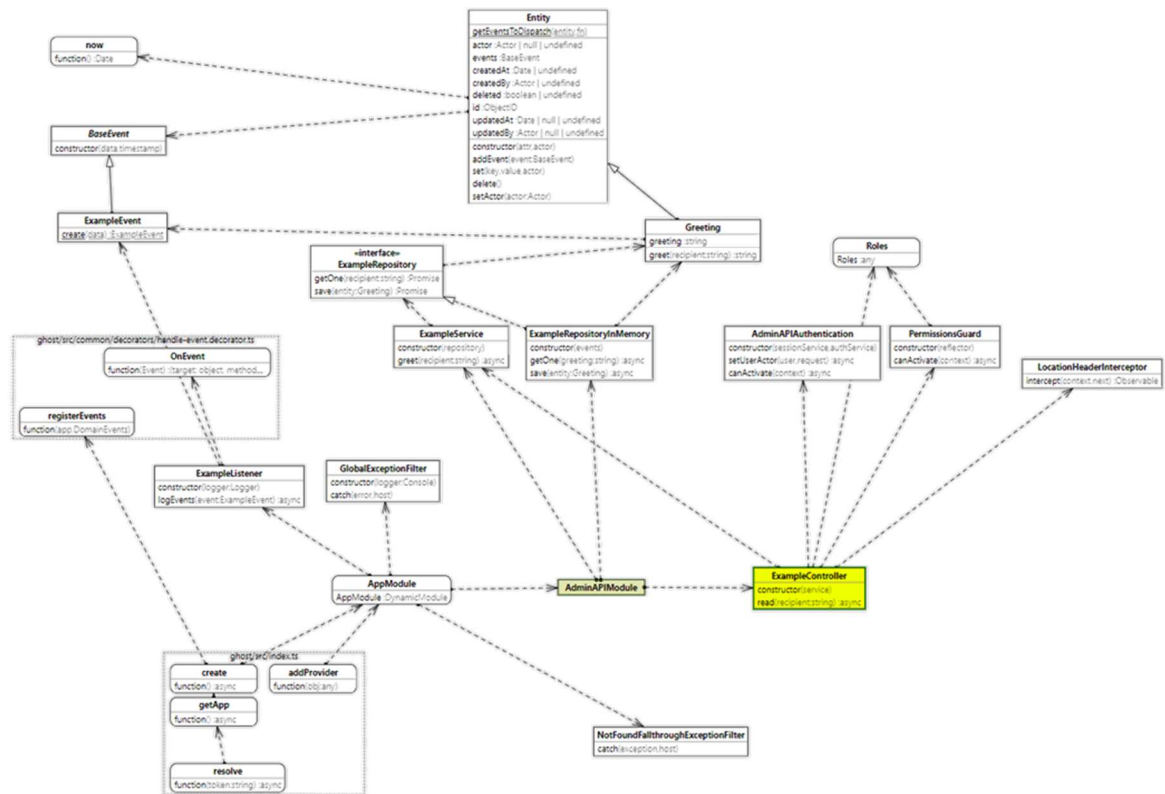
### Diagrama de Componentes



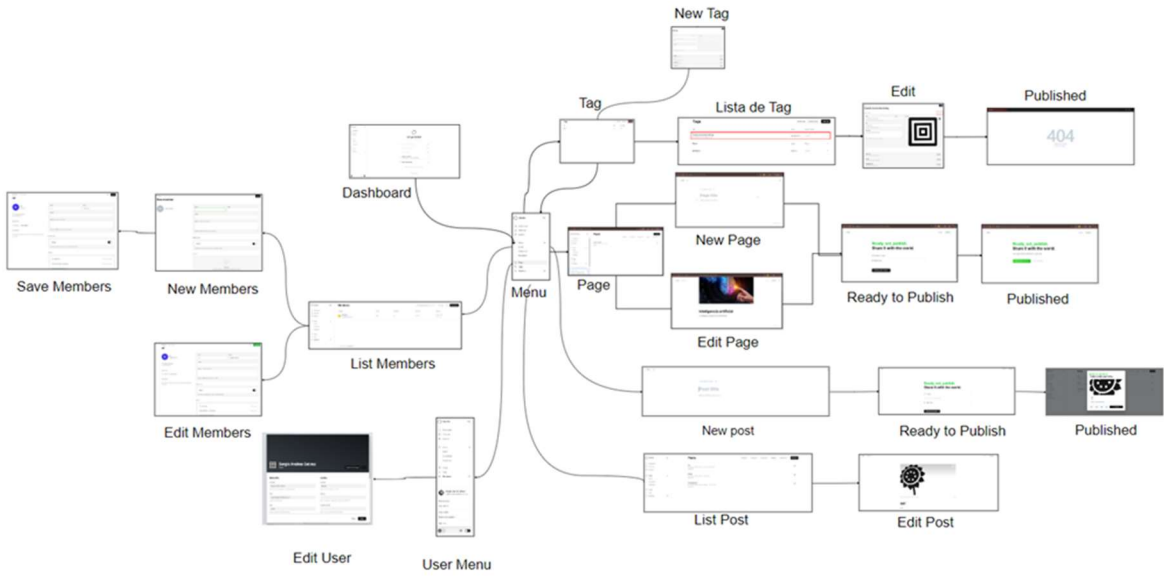
## 1.6 Diagrama de Contexto



## 1.7 Modelo de datos



1.8 Modelo de GUI



## **2 CONTEXTO DE LA ESTRATEGIA DE PRUEBAS**

### **2.1 Objetivos:**

Se pretende validar aspectos funcionales del sistema GHOST mediante pruebas tanto de caja negra como de caja blanca, priorizando pruebas automatizadas para alcanzar la mayor cobertura posible. Los objetivos principales son:

1. Cubrir adecuadamente áreas de prueba manual donde la interacción del usuario pueda generar errores.
2. Realizar pruebas exploratorias para encontrar casos raros que se salgan del resto de pruebas que se plantean.
3. Verificar funcionalidades críticas del sistema, enfocándose en aquellas de alto impacto para el usuario usando pruebas de extremo a extremo.
4. Realizar pruebas de regresión visual para garantizar la estabilidad de funcionalidades en diferentes navegadores usando ResembleJs.
5. Automatizar pruebas de validación de datos de procesos clave, como la creación y publicación de contenido que permitan probar el buen funcionamiento de los diferentes formularios.

### **2.2 Duración de la iteración de pruebas:**

**Fechas:** Inicia el 25 de noviembre al 17 de enero (8 semanas). Horas totales: 256

**Ingenieros senior (4):** 32 horas semanales.

**Distribución de tiempo:** Las 256 horas se refieren al tiempo de trabajo de los ingenieros más no al tiempo de ejecución en máquina y están distribuidas de la siguiente manera:



<b>Tipo de Prueba</b>	<b>Dedicación en el proyecto</b>
Pruebas manuales	12 horas
Pruebas Exploratorias	12 horas
Pruebas e2e	102 horas
Pruebas VRT	40 horas
Validación de datos	90 horas

Para mayor detalle de la distribución de las horas en las diferentes semanas ver el numeral **2.5 Distribución de Esfuerzo**:

### **2.3 Presupuesto de pruebas:**

#### **2.3.1 Recursos Humanos:**

**Ingeniero automatizador Senior:** 4 ingenieros con 8 horas disponibles para la creación de scripts y configuración de herramientas de reconocimiento.

#### **2.3.2 Recursos Computacionales:**

400 horas en máquinas propias para la ejecución de pruebas automatizadas y de reconocimiento.

## 2.4 TNT (Técnicas, Niveles y Tipos) de Pruebas:

Las pruebas se estructuran de acuerdo con la pirámide de automatización y clasifican en pruebas de caja negra y caja blanca. Las pruebas de caja negra validan el sistema desde la perspectiva del usuario, sin necesidad de conocimiento interno, mientras que las de caja blanca verifican el funcionamiento del código. En esta estrategia se priorizan los siguientes niveles y tipos:

Nivel	Tipo	Técnica	Objetivo
Sistema	Funcional	Pruebas Manuales	Objetivo 1
Sistema	Funcional Caja Negra	GUI Ripping (Ripper)	Objetivo 2
Sistema	Funcional	APIs De Automatización (cypress)	Objetivo 3
Sistema	Funcional	Pruebas Automatizadas de Regresión Visual (ResembleJs)	Objetivo 4
Sistema	Funcional	APIs de Automatización (a priori, pool, ramdon) (Cypress)	Objetivo 5

**Pruebas manuales:** Sirven para detectar errores o comportamientos inesperados en casos específicos, explorar áreas nuevas o críticas del sistema, y verificar aspectos subjetivos como la experiencia de usuario.

**Pruebas de extremo a extremo:** Estas pruebas evalúan el sistema completo desde el inicio hasta el final, simulando escenarios reales de uso. Verifican la integración y comunicación entre todos los componentes del sistema, asegurando que las funcionalidades principales funcionen correctamente en un entorno realista.

**Pruebas de regresión visual:** Comparan visualmente interfaces o páginas web entre diferentes versiones del sistema para detectar cambios inesperados o no deseados en el diseño, estilos o contenido visual. Son útiles en proyectos con interfaces dinámicas o frecuentes actualizaciones de diseño.

**Pruebas de validación de datos:** Verifican la consistencia, precisión y completitud de los datos dentro de un sistema. Aseguran que los datos ingresados, transformados o transferidos entre sistemas se comporten de acuerdo con los requisitos especificados.

**Pruebas de sistema:** Validar el sistema completo desde el punto de vista del usuario final.

**Prueba Funcional:** Verifica que el sistema funcione según lo esperado.

**Prueba de caja blanca:** Evalúa la lógica interna del código, verificando su estructura y funcionamiento sin considerar las entradas y salidas.

**Prueba de caja negra:** Se centra en las entradas y salidas del sistema, validando que funcione según los requisitos sin necesidad de conocer su implementación interna.

## **2.5 Distribución de Esfuerzo:**

La siguiente sección presenta la distribución del esfuerzo en dos etapas clave del proyecto, considerando las actividades realizadas por los equipos de desarrollo y pruebas. En ambas tablas se observa cómo se desglosan las horas y los recursos asignados a diferentes tipos de tareas, incluyendo desarrollo, pruebas y validación.

Horas dedicadas a Pruebas					
	Manuales	Exploratorias	e2e	VRT	Validación de datos
Semana1	4	4	20	2	2
Semana2	4	4	18	2	4
Semana3	4	4	16	4	4
Semana4	0	0	14	4	14
Semana5	0	0	12	6	14
Semana6	0	0	10	6	16
Semana7	0	0	8	8	16
Semana8	0	0	4	8	20

Cantidad de pruebas estimadas					
	Manuales	Exploratorias	e2e	VRT	Validación de datos
Semana1	8	1 script	20	2	16
Semana2	8	1 script	18	2	32
Semana3	8	1 script	16	4	32
Semana4	0	1 script	14	4	112
Semana5	0	1 script	12	6	112
Semana6	0	1 script	10	6	128
Semana7	0	1 script	8	8	128
Semana8	0	1 script	4	8	160

A continuación, se detalla cada etapa en función de los datos presentados:

### 2.5.1 Primera Etapa:

En esta etapa, se distribuyen los esfuerzos principalmente en actividades de desarrollo, con una asignación significativa hacia las pruebas y validación.

Las horas están concentradas en pruebas funcionales (20 horas), validación de datos y pruebas visuales, mientras que las pruebas exploratorias y de regresión ocupan un menor porcentaje del total.

### 2.5.2 Segunda Etapa:

Se observa una redistribución del esfuerzo, con un aumento notable en las pruebas y validaciones automatizadas.

Las pruebas de regresión visual tienen un incremento considerable, alcanzando un total de 160 horas, lo que indica un enfoque más automatizado y robusto en esta etapa del proyecto.

Las pruebas manuales y exploratorias mantienen una presencia, pero representan una menor proporción en comparación con el uso de herramientas y pruebas automatizadas.

Estas tablas reflejan cómo la estrategia evoluciona a lo largo del proyecto para adaptarse a las necesidades cambiantes y a la optimización de los recursos disponibles.

# AnexoS

- Link de video presentación: <https://youtu.be/LiLmOtVEPr4>