

# Estrategia de Pruebas – Presupuesto 1

## 1. Aplicación Bajo Pruebas

**1.1. Nombre Aplicación:** Ghost

**1.2. Versión:** 3.41.1

**1.3. Descripción:**

Ghost es una plataforma todo en uno, pensada y diseñada para que los creadores de medios y empresas administren, publiquen, compartan y hagan crecer un negocio en torno a su contenido.

Ghost cuenta con múltiples herramientas modernas para crear un sitio web, enviar boletines y ofrecer suscripciones pagas o gratuitas a sus miembros.

En términos tecnológicos Ghost es una API de contenido RESTful que ofrece contenido abierto, por ende, cualquier cliente puede acceder a él en modo de solo lectura para representarlo en un sitio web, una aplicación u otro medio incrustado.

El control de acceso se gestiona a través de una clave API, e incluso los filtros más complejos se simplifican con el potente SDK que esta aplicación brinda. La API de contenido está diseñada para ser totalmente almacenable en caché, lo que significa que puede obtener datos con la frecuencia que desee sin limitaciones.

**1.4. Funcionalidades Core:**

- Post: Crear, Editar, Visualizar, Publicar, Despublicar, Draft, Scheduler
- Pages: Crear, Editar, Visualizar, Publicar Despublicar
- Tag: Crear, Editar, Eliminar
- Members: Crear, Editar, Eliminar, Importar
- DashBoard
- Login: Sign In, Sign out, Recordar contraseña
- Personalizar sitio:

Members: Membership, Email Newsletter

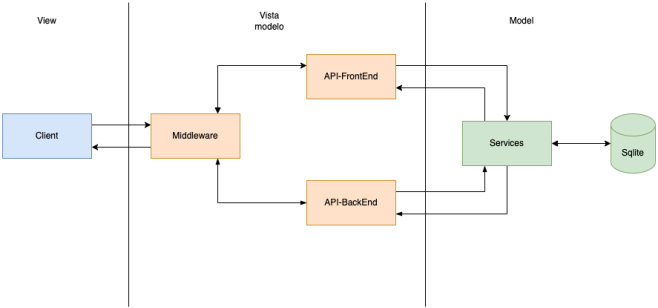
WebSite: General, Design, Navigation, Staff

Advanced: Integration, Code Injection, Labs

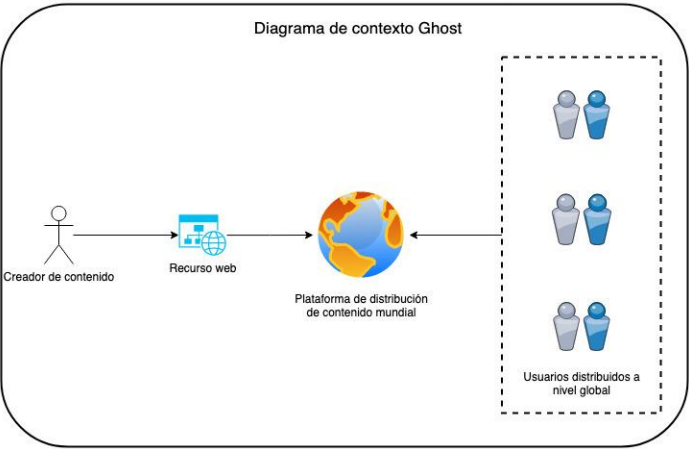
Plantilla elaborada por

**THE SW DESIGN LAB**

1.5. Diagrama de Arquitectura:



1.6. Diagrama de Contexto:



1.7. Modelo de Datos:

[Modelo de Datos](#) (Este enlace debe ser accedido con usuarios Uniandes)

1.8. Modelo de GUI:

[Modelo de GUI](#) (Este enlace debe ser accedido con usuarios Uniandes)

## 2. Contexto de la estrategia de pruebas

Debido a la transformación digital y a la actualidad tecnológica del mercado **The Software Design Lab** viene incursionando en un proceso de transformación hace ya varios años, en el cual se ha adquirido capacitaciones a nivel de nuevas tecnologías, metodologías ágiles y algunas herramientas de despliegue e integración continua en la nube, lo anterior ha generado el aumento de las funcionalidades expuestas en la aplicación Ghost. Lamentablemente sin un uso adecuado de pruebas a diferentes niveles. Actualmente el sistema cuenta con un bajo porcentaje de pruebas unitarias y cobertura de código, sin embargo, la empresa busca optimizar tiempos asociados al personal dedicado exclusivamente a pruebas manuales de aceptación que se repiten en cada reléase. Por lo anterior se han destinado tres ingenieros automatizadores junior con el objetivo de validar, actualizar y mejorar el inventario de pruebas.

Respecto a la entrega pasada se realizaron las siguientes modificaciones:

- Se ajusto la estrategia de pruebas con base en la retroalimentación recibida por parte del tutor, respecto a integrar el proceso de CI/CD y mejorar la distribución de esfuerzo por funcionalidades.
- Se decide mejorar las pruebas de sistema mediante reconocimiento automático con un monkey con el fin de identificar errores no comunes en toda la aplicación, para su posterior reporte y registro en issue tracker, lo anterior con base en las siguientes ventajas:
  - Se puede explorar el sistema aleatoriamente, en busca de fallas a través de eventos planificados.
  - Facilidad de implementación.
  - Existe trazabilidad de las evidencias, lo cual permite reconstruir los pasos para posteriormente solucionar la falla identificada.
  - La ejecución de estas pruebas es de bajo costo.
  - La ejecución es rápida.
  - Permite la identificación de casos no comunes.
- Se incluye el uso de un ripper para la identificación del diagrama de GUI de transiciones de la aplicación bajo pruebas, así mismo como la detección de errores, esto sustentado en las siguientes ventajas:
  - La ejecución y navegación de las pruebas es completamente automática, lo que permite aprovechar las horas/maquina en mayor medida.
  - Permite identificar con claridad los nodos ejecutados.
  - Generación automática de screenshots de evidencias de los errores.
  - Visualización del grafo completo de la aplicación con marcas que permiten identificar con claridad las transiciones y fallas en las mismas.

Los cambios anteriormente mencionados requieren un ajuste significativo en el cronograma y la distribución de esfuerzos, esto suponen una mejora significativa en las iteraciones y los resultados de esta primera etapa, optimizando los recursos disponibles, tanto humanos como computacionales.

Adicional a los cambios anteriormente mencionados, se cambió la versión de ghost bajo pruebas, pasamos de la versión v5.20.0(latest) a la versión 3.41.1, esto por dos razones en particular, la primera es para tener una aplicación un poco más inestable que nos permitiera realizar un mejor uso y entendimiento de las herramientas de monkey y Ripper, como segunda razón es para tener en cuenta que, más adelante en el curso se quiere contrastar la versión bajo pruebas con una futura, con este cambio se podrá realizar sin ningún inconveniente.

**Respecto a la entrega de la semana 6:**

En la semana 6 se decidió incluir tiempo de los desarrolladores en implementar la automatización de pruebas de regresión visual (VRT).

Estas pruebas se incluirán como parte de las pruebas de sistema ya existentes en el cronograma y se destinaron 2 horas del ingeniero junior para tal fin.

**Respecto a la entrega de la semana 7:**

En esta semana se toma la decisión de incluir en las pruebas automatizadas la generación de diferentes pool de datos para la ejecución de los escenarios, con caracteres de diferentes tipos, numéricos, large string, short string y con caracteres especiales, para ampliar el espectro de posibilidades de las pruebas.

Lo anterior no modifica el manejo de los recursos dados previamente en la estrategia.

**2.1. Objetivos:**

Implementar durante el periodo del 31 de octubre al 2 de noviembre un inventario de pruebas que garantice la cobertura del 0% al 30% de funcionalidad del sistema bajo pruebas, mediante la validación de la documentación existente y la exploración de las funcionalidades POST, PAGE y creación de Tags.

Implementar al 10 de noviembre del 2022 el set de pruebas unitarias que genere un cubrimiento de código de un 55% a un 90% de las funcionalidades POST, PAGE y creación de Tags, de acuerdo, al inventario de pruebas disponible.

Reducir los prolongados tiempos de ejecución de pruebas manuales, de 2 semanas a aproximadamente 26 minutos mediante el uso de pruebas de aceptación automáticas y la integración de las mismas a los procesos internos de DevOps realizados en el equipo de QA, de las funcionalidades POST, PAGE y creación de Tags, de acuerdo, al inventario de pruebas disponible, lo anterior al 24 de noviembre del año 2022.

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

Descripción	Iteración	Fecha inicio	Fecha fin
Mejorar inventario de pruebas (Validar e incrementar)	1	31 de octubre del 2022	4 de noviembre del 2022
Mejorar cobertura de código	2	8 de noviembre del 2022	11 de noviembre del 2022
Implementación de pruebas de aceptación automáticas	3	15 de noviembre del 2022	24 de noviembre del 2022

**\*Horario de lunes a viernes de 8:00am a 5:00 pm, no se trabajan fines de semana ni feriados.**

### 2.3. Presupuesto de pruebas:

#### 2.3.1. Recursos Humanos

Se cuenta con 3 ingenieros junior que se encuentran en diferentes áreas de la empresa, quienes cumplen con los perfiles requeridos para la estrategia de pruebas que se desea realizar, actualmente están con contrato a término indefinido y directamente por la empresa, ellos tienen las siguientes habilidades:

- Conocimientos básicos en creación de sets de pruebas automatizados.
- Conocimientos básicos en creación y edición de scripts de automatización.
- Conocimientos básicos de técnicas de QA, herramientas de automatización y pruebas de desempeño, como Selenium, Cucumber, JMeter, **Cypress, Playwright** etc.
- Conocimientos en métodos de automatización (Data-Driven, Capture & Replay y/o Scripting).
- Conocimientos básicos en programación en Web. HTML. XML. CSS, NodeJS.
- Conocimientos básicos en integración continua y liberación continua.
- Conocimientos básicos en pruebas de regresión (incluidas pruebas de regresión visual)

Las tareas serán desarrolladas durante el tiempo definido para la estrategia por los ingenieros y durante la jornada laboral, posteriormente regresarán a sus respectivos equipos.

#### 2.3.2. Recursos Computacionales

Los recursos computacionales disponibles están divididos en 3 grupos:

1. Recursos de cómputo locales de los ingenieros (laptops)
2. Recursos de cómputo en nube para despliegue de pruebas productivas (servidores en AWS)
3. Herramientas de grabación

##### 1. Recursos de cómputo locales de los ingenieros (laptops)

Actualmente cada ingeniero cuenta con una máquina portátil con las siguientes características:

Sistema Operativo (S.O)	Windows 11
Procesador	Ryzen 5 5600X
Memoria RAM	16GB
Espacio de almacenamiento	500GB SSD

##### 2. Recursos de cómputo en nube para despliegue de pruebas productivas (servidores en AWS)

El equipo de trabajo contara máximo con 200 horas/servidor en a AWS, con servidores de las siguientes características:

Tipo	SO	vCPU	Memoria	Almacenamiento
EC2 m5.2xlarge	Linux	8	32GB	500GB SSD

### 3. Herramientas de grabación

Como dispositivo de grabación de los escenarios de pruebas se tiene disponible Cypress y PlayWright.

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

Todos los recursos asignados a la estrategia son directamente suministrados por la empresa.

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

Tipos	Niveles	Técnicas	Objetivo
Caja negra	Pruebas de sistema	Pruebas Manuales Pruebas Automatizadas (API's automatización)	Incrementar el inventario de pruebas
Caja Blanca	Pruebas de Unidad	Pruebas Automatizadas (API's automatización)	Mejorar la cobertura de código
Funcionales	Pruebas de Aceptación	Pruebas Automatizadas (API's automatización)	Automatizar las pruebas para reducir los esfuerzos manuales

Comentado [DP1]: No olviden que esto debe ir enlazado con la sección 2.1.

#### 2.5. Distribución de Esfuerzo

Las tareas propuestas se realizarán en un ambiente de desarrollo de acuerdo al siguiente cronograma por nivel de prueba:

N° Iteración	Nivel	Duración Horas/Persona	Alcance	Cronograma
1	Pruebas de sistema	42 horas	Exploración de las funcionalidades POST, PAGE y creación de Tags	31/10/2022 - 02/11/2022
2	Pruebas de unidad	29 horas	Implementar el set de pruebas unitarias que genere un cubrimiento de código de 90% de las funcionalidades POST, PAGE y creación de Tags.	03/11/2022 - 10/11/2022
3	Pruebas de aceptación	79 horas	Reducir tiempos de pruebas, de 1 semana a aproximadamente 26 minutos mediante la integración a los procesos internos de devops, de las funcionalidades POST, PAGE y creación de Tags. Implementar sistema automático de VRT	10/11/2022 - 24/11/2022

2.5.1 Distribución de esfuerzo por ingeniero y caso de prueba en horas

Construcción				
Nivel	Tipo	Tarea	Duración horas	Persona
Pruebas de sistema	Manuales	Configuración ripper	2	Ingeniero 1
		Configuración monkey	2	Ingeniero 1
		Ajustes ripper	2	Ingeniero 1
		Ajustes Monkey	2	Ingeniero 1
		Revisión ejecuciones	4	Ingeniero 1
		Actualización de escenarios	1	Ingeniero 1
		Reporte de incidentes	5	Ingeniero 2
		Exploración CRUD post	14	Ingeniero 1
		Exploración CRUD page	8	Ingeniero 1
		Exploración crear tag	1	Ingeniero 1
Pruebas de unidad	Automatizadas	CRUD Post	13	Ingeniero 2
		CRUD Page	13	Ingeniero 3
		Crear tag	2	Ingeniero 1
Pruebas de aceptación	Automatizadas	CRUD Post	32	Ingeniero 2
		CRUD Page	37	Ingeniero 3
		Crear tag	7	Ingeniero 1
		VRT Automático	5	Ingeniero 1

La estrategia aquí planteada permitirá entregar mediante 3 iteraciones el inventario de pruebas actualizado, un aumento de cobertura de código y la automatización de pruebas de las funcionalidades elegidas.

A lo largo de las iteraciones, los procesos automáticos se ejecutarán como se muestra a continuación con datos de entrada y oráculo de pruebas definido de forma estática, de la siguiente manera:

Ejecución				
Nivel	Tipo	Tarea	Duración horas total	Nº Maquinas
Pruebas de sistema	Automáticas	Ripper profundidad 1	2	2
		Ripper profundidad 2	4	2
		Ripper profundidad 3	8	2
		Monkey	4	2
Pruebas de unidad	Automatizadas	CRUD Post	10	1
		CRUD Page	10	1
		Crear tag	10	1
Pruebas de aceptación	Automatizadas	CRUD Post	66	5
		CRUD Page	66	5
		Crear tag	20	5

- Se dispondrá de un máximo de 5 servidores, y su uso dependerá del tiempo de ejecución de las pruebas vs el rendimiento de cada máquina, mediante una estrategia de escalamiento horizontal, que tiene como objetivo, ejecutar el sanity de las funcionalidades elegidas y los tiempos objetivo de ejecución del set completo máximo en 26 minutos.
- Con los logs resultantes de cada ejecución, se verificará la correcta funcionalidad de todos los casos de pruebas, en caso de presentarse inestabilidad en las mismas se realizará el análisis y reporte de incidencias necesarios de formar temprana, esto lo realizará uno de los 3 ingenieros con 2 horas dedicadas.
- Se destinan máquinas en AWS para la ejecución de pruebas de forma automatizada, lo que permite a los Ingenieros de automatización validar las entradas, procesos y salidas de los scripts desarrollados, antes de integrar los casos de pruebas al flujo de CI/CD, esto lo realizara el ingeniero 2 como se planteó en la distribución de esfuerzos.
- Se destinarán las 24 horas maquina disponibles del presupuesto, para la ejecución de los releases simulados ejecutando las pruebas de aceptación, durante diferentes horas del día.



2.5.2 Distribución de esfuerzo actual por caso de prueba versus esperado después de la automatización

Funcionalidad	Prueba	Tiempo actual de ejecución en horas	Tiempo esperado de ejecución mediante automatización en horas
post	Crear	6,5	0,166667
	Editar	13	0,166667
	Visualizar	13	0,166667
	Publicar	6,5	0,166667
	Despublicar	6,5	0,166667
	Draft	6,5	0,166667
	Scheduler	6,5	0,166667
pages	Crear	6,5	0,166667
	Editar	6,5	0,166667
	Visualizar	13	0,166667
	Publicar	6,5	0,166667
	Despublicar	6,5	0,166667
Tag	Crear	6,5	0,166667

En la actualidad, la ejecución del proceso de pruebas toma un tiempo lineal de 104 horas, estas pruebas ejecutan paralelamente y se requiere de 3 personas trabajando. El proceso finaliza regularmente al cabo de dos semanas de trabajo, lo que impide a la empresa adaptarse a la velocidad de la industria y realizar la entrega de valor en el menor tiempo posible.

Con el nuevo planteamiento se estima que la ejecución tarde un total de 2.16 horas lineales, sin embargo, mediante la ejecución automática de los casos de pruebas en paralelo, usando 5 servidores, se reducirán los tiempos a 26 minutos en total, con lo anterior se busca, integrar el proceso de prueba en las diferentes etapas del ciclo de desarrollo, de forma continua y permitiendo entregar valor con mayor frecuencia.