

Estrategia de Pruebas – Presupuesto 2

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:

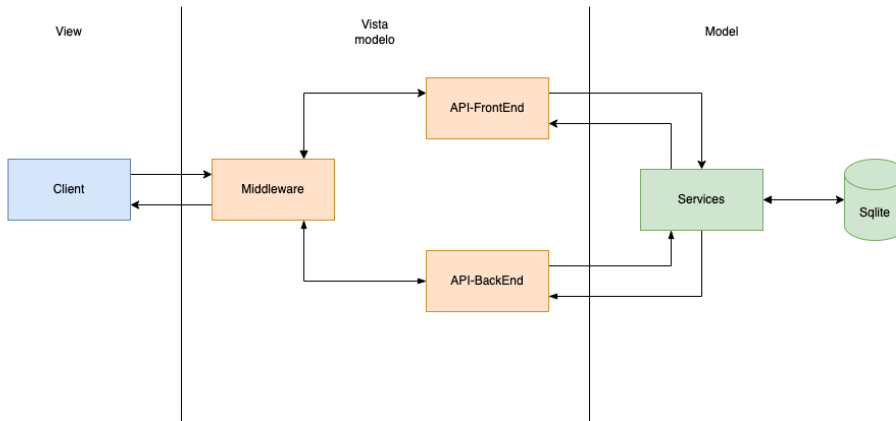
- Post: Crear, Editar, Visualizar, Publicar, Despublicar, Draft, Scheduler
- Pages: Crear, Editar, Visualizar, Publicar Despublicar
- Tag: Crear, Editar, Eliminar
- Members: Crear, Editar, Eliminar, Importar
- DashBoard
- Login: Sign In, Sign out, Recordar contraseña
- Personalizar sitio:

Members: Membership, Email Newsletter

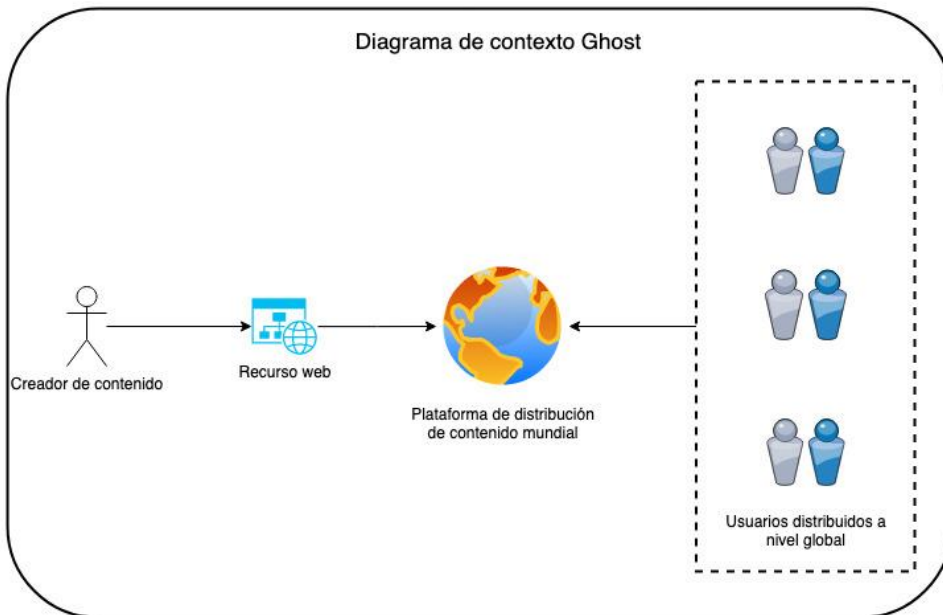
WebSite: General, Design, Navigation, Staff

Advanced: Integration, Code Injection, Labs

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 han 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. La empresa busca optimizar tiempos asociados al personal dedicado exclusivamente a pruebas manuales de aceptación que se repiten en cada release.

Respecto a la entrega pasada, se ha incluido el uso de un Ripper como parte de esta estrategia de pruebas teniendo en cuenta que se tienen muchas horas servidor disponibles y pocas horas ingeniero.

Dado que los monkeys llevan una cantidad significativa de tiempo (en horas ingeniero) en ser ajustados para un correcto funcionamiento y no suelen tener un impacto representativo en las primeras iteraciones de un nuevo proceso de pruebas (alta tasa de falsos positivos, alta tasa de errores de ejecución) se ha decidido no incluir el uso de ninguna herramienta de este tipo.

Por otro lado, el Ripper requiere una inversión de tiempo (en horas de ingeniero senior) baja para ser configurado correctamente, y puede aportar de manera significativa (generar nuevos casos de prueba, exploración inicial automática de nodos de la aplicación) a la cobertura de las pruebas en esta etapa inicial, por lo que se ha decidido incluirlo como parte del set de pruebas y objetivos de la iteración.

Los cambios anteriormente mencionados no requieren ningún tipo de ajuste en el presupuesto, pero si suponen una mejora significativa en los resultados de esta primera iteración optimizando los recursos disponibles (humanos y tecnológicos).

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 5:

En la semana actual (semana 5) se decidió incluir el uso de Cypress para el desarrollo de pruebas E2E (2 escenarios de pruebas).

Estas pruebas se incluirán como parte de las pruebas de sistema ya existentes en el cronograma y se destinaron 3 horas del ingeniero junior para tal fin.

Respecto a la entrega de la semana 6:

En la semana actual (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 decidió incluir pruebas con datos aleatorios al set de pruebas ya existentes, así como la validación del manejo de datos no previstos en longitud y tipo en los diferentes campos de formularios.

Este cambio no modifica los tiempos previamente previstos, pero si cambia el alcance y detalle de la estrategia de pruebas planteada.

2.1. Objetivos:

Entender el funcionamiento del 100% de la aplicación aplicando pruebas de descubrimiento, con una aproximación de caja negra durante la primera iteración (1 semana).

Pasar de un 0% a un 5% de cobertura de código bajo el uso de pruebas unitarias durante la primera iteración (1 semana). Estas estarán enfocadas en validar y verificar las funcionalidades core del sistema, permitiendo entregar el mayor valor posible al usuario con los recursos entregados.

Consideramos que dadas las horas ingeniero disponibles y el tamaño del código fuente de la aplicación, el 5% del código es una cobertura razonable con el tiempo disponible

Aplicar pruebas automáticas de tipo Ripper utilizando la herramienta RiPuppet al menos en un nivel de profundidad 3 para aumentar la cobertura de las pruebas.

Integrar el 100% de las pruebas unitarias desarrolladas en la iteración 1 (1 semana) en el flujo de ci/cd, garantizando la ejecución de las mismas de manera continua a lo largo del proceso de despliegue.

2.2. Duración de la iteración de pruebas:

Esta será la primera iteración de pruebas.

Teniendo en cuenta los recursos disponibles, esta iteración tendrá una duración de una semana (de lunes a viernes, días hábiles en Colombia) trabajando en un horario de 9 AM a 4PM con 1 hora de receso para almuerzo. Este horario será el marco de referencia para las horas trabajadas, sin embargo, se ajustará según la cantidad de horas contratadas para cada ingeniero dado que la cantidad de horas disponibles por ingeniero según el seniority son diferentes.

Esta iteración dará inicio el día 31 de octubre de 2022 y finalizará el 4 de noviembre de 2022.

2.3. Presupuesto de pruebas:

El presupuesto disponible para las pruebas es el siguiente:

- 500 USD para contratación de servicios de outsourcing de pruebas
- 30 horas de un ingeniero (nivel junior) automatizador
- 400 horas máquina en AWS.

2.3.1. Recursos Humanos

Consideramos que para esta primera iteración necesitaremos un perfil de ingeniero automatizador senior, que permita generar una base de código sólida para las pruebas, y además se encargará de hacer los ajustes necesarios a RiPuppet para incluir pruebas automáticas. Por otro lado, se tendrá un perfil junior que se encargará del desarrollo más detallado.

El ingeniero automatizador senior será contratado con los 500 USD disponibles para pruebas por outsourcing (ver detalles más adelante) y el ingeniero junior será contratado internamente durante un periodo inicial de 30 horas para la primera iteración. Adicionalmente, el perfil del ingeniero junior deberá ser el siguiente:

- Conocimientos básicos en creación sets de pruebas automatizados.
- Conocimientos en creación y edición de scripts de automatización.
- Conocimiento básico en pruebas unitarias
- Conocimientos básicos en métodos de automatización (Data-Driven, Capture & Replay y/o Scripting).
- Conocimientos de programación básicos en tecnologías web: HTML. XML. CSS.
- Conocimientos básicos de programación en Javascript, dado que es el lenguaje de programación en el que está desarrollado Ghost
- Conocimientos básicos en Cypress
- Conocimientos básicos en pruebas de regresión (incluidas pruebas visuales)

Por otro lado, el perfil del ingeniero de pruebas senior será el siguiente:

- Conocimientos avanzados en diseño y creación sets de pruebas automatizados.
- Conocimientos avanzados en creación y edición de scripts de automatización.
- Conocimiento avanzado en pruebas unitarias
- Conocimientos avanzados en métodos de automatización (Data-Driven, Capture & Replay y/o Scripting).
- Conocimientos avanzados de programación en tecnologías web: HTML. XML. CSS.
- Experiencia de al menos 8 años en entornos de pruebas y/o automatización de pruebas de software.
- Conocimientos avanzados de programación en Javascript, dado que es el lenguaje de programación en el que está desarrollado Ghost

2.3.2. Recursos Computacionales

Los recursos computacionales disponibles están divididos en 2 grupos:

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

Respecto al primer grupo, las especificaciones técnicas de los laptops que le serán entregados a los dos ingenieros serán las siguientes:

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

Para el segundo grupo, consideramos que vale la pena distribuir las 400 horas de máquina en múltiples servidores por las siguientes razones:

- Se aíslan posibles cruces de recursos para diferentes pruebas en el mismo servidor
- Optimización de uso de horas de cómputo disponibles de acuerdo a la duración de la iteración

Las especificaciones técnicas de los servidores que le serán aprovisionados para el despliegue en producción de las pruebas son:

Sistema Operativo (S.O)	Linux
Procesadores	8
Memoria RAM	32GB
Espacio de almacenamiento	500GB SSD

Las especificaciones mencionadas anteriormente corresponden al tamaño de una instancia EC2 m5.2xlarge

Recursos Económicos para la contratación de servicios/personal:

El ingeniero automatizador senior será contratado haciendo uso de los \$500 USD presupuestados para servicios de outsourcing.

Existe, dentro del mercado colombiano, un referente confiable para la revisión de salarios en diferentes posiciones en la industria tecnológica: Estudio de remuneración de Michael Page

QA				
No.	Posición	Años de experiencia	Salario Bruto Min	Salario Bruto Max
1	QA Automation	3 a 5	\$ 8,000,000	\$ 13,000,000
		5 a 8	\$ 13,000,000	\$ 18,000,000
		8 a 12	\$ 18,000,000	\$ 25,000,000
2	QA Functional	3 a 5	\$ 6,000,000	\$ 8,000,000
		5 a 8	\$ 8,000,000	\$ 10,000,000
		8 a 12	\$ 10,000,000	\$ 13,000,000

Figura 1. Fuente: Estudio remuneración Michael Page 2022

Teniendo en cuenta el estudio de remuneración mostrado en la Figura 1, el costo por hora de un ingeniero automatizador senior es de \$23 USD. Con el presupuesto inicial, se contratarán 21 horas.

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

Tipos	Niveles	Técnicas	Objetivo
-------	---------	----------	----------

Caja negra	Pruebas de sistema	Pruebas Manuales	Conocer el 100% del funcionamiento de Ghost
Caja Blanca	Pruebas de Unidad	Pruebas Automatizadas (API's automatización) con record & replay	Incrementar de 0% a 5% la cobertura de código
Caja negra	Prueba de sistema	Pruebas Automatizadas con Ripper RiPuppet + Pruebas E2E con Cypress utilizando datos aleatorios y validación de límites en campos de formularios Pruebas de VRT	Incrementar la cobertura de pruebas

2.5. Distribución de Esfuerzo

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

Tipo de prueba	Tarea	Detalle de tarea	Duración	Perfil
Pruebas de sistema	Entendimiento inicial del contexto	Exploración de las funcionalidades POST, PAGE y creación de Tags	2 horas	Ingeniero Junior
Pruebas de sistema	Creación de 2 escenarios para pruebas E2E	Codificación de dos escenarios de pruebas E2E utilizando Cypress	2 horas	Ingeniero Junior
Pruebas de sistema	Automatización de VRT	Codificación de la automatización de pruebas de regresión visual	3 horas	Ingeniero Junior
N/A	Sincronización interna	Sincronización entre el ingeniero junior y senior	4 horas	Ingeniero Junior + Ingeniero Senior
Pruebas unitarias	Capacitación framework de pruebas	Capacitación en pruebas unitarias como tecnología para desarrollo de pruebas	2 horas	Ingeniero Junior
Pruebas unitarias	Desarrollo de pruebas	Desarrollo de pruebas unitarias	10 horas	Ingeniero Junior
N/A	Documentación	Documentación interna de pruebas desarrolladas	4 horas	Ingeniero Junior
N/A	Entrega	Entrega y cierre de la primera iteración	1 hora	Ingeniero Junior + Ingeniero Senior
Pruebas de sistema	Entendimiento inicial del contexto	Exploración de las funcionalidades POST, PAGE y creación de Tags	2 horas	Ingeniero Senior
Ajuste de RiPuppet	Codificación de pruebas	Ajuste de la herramienta RiPuppet para garantizar un correcto funcionamiento con Ghost.	2 horas	Ingeniero Senior
Análisis y exploración de resultados RiPuppet	Análisis de resultados	Análisis de todos los hallazgos generados por RiPuppet	2 horas	Ingeniero Junior

Pruebas unitarias	Diseño de pruebas	Diseño y desarrollo de pruebas unitarias	10 horas	Ingeniero Senior
N/A	Integración en CI/CD	Integrar las pruebas unitarias desarrolladas en el ciclo de CI/CD	2 horas	Ingeniero Senior

Para distribuir las horas disponibles de cómputo, serán desplegados 5 servidores que serán usados tanto para la fase de desarrollo inicial de las pruebas como para la puesta en producción de las mismas en la iteración 1.

2 de los servidores en la nube (AWS) mencionados anteriormente serán dedicados exclusivamente para RiPuppet dado que se requiere el uso ininterrumpido para una correcta ejecución de dichas pruebas.

Las especificaciones de este servidor permiten ejecutar pruebas que pueden llegar a ser de un consumo medio/alto en términos de procesamiento, uso de RAM y espacio de almacenamiento. Este último es especialmente sensible para la ejecución del Ripper.

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 de pruebas automaticas						
Cantidad de Servidores	Ambiente	Cronograma	Horas destinadas	Horarios GMT - 5	Trigger	Proposito
2	Pruebas	31/10/2022 - 04/11/2022	100	22:00 - 01:00	Proceso de integración en CI/CD Manual	Pruebas unitarias
3	Producción	31/10/2022 - 04/11/2022	300	24 x 7	Proceso de integración en CI/CD	Pruebas unitarias