

Estrategia de Pruebas

1. Aplicación Bajo Pruebas

1. **Nombre Aplicación:** Ghost -The Software Design Company
2. **Versión:** 3.42.5
3. **Descripción:**

Ghost es una plataforma de gestión y creación de contenido para bloggers independientes y grandes equipos de escritores y editores concentrados en la creación y difusión de material informativo para la web en formato escrito. La aplicación permite administrar el contenido y publicar el mismo una vez está listo al igual que conformar equipos de trabajo o trabajar de forma individual.

Es una aplicación open source sin ánimo de lucro en la que trabajan constantemente múltiples equipos de personas alrededor del mundo.

4. Funcionalidades Core:

Identificador	Nombre	Descripción
F001	Gestión de contenido	La aplicación se encarga de la gestión de contenido.
F002	Artículos	Desde el panel de administración es posible crear, editar y eliminar artículos según se requiera.
F003	Etiquetas	Desde el panel de administración es posible crear, editar y eliminar etiquetas según se requiera.
F004	Páginas	Desde el panel de administración es posible crear, editar y eliminar páginas según se requiera.
F005	Autores	Es posible asignar los un autor o varios autores a los artículos.
F006	Gestión de usuarios	Se puede crear, editar o eliminar los usuarios según sea necesario.
F007	Configuración de cuenta	Se puede editar la información y configuración de la cuenta desde el panel de administración.
F008	Configuración de diseño	Desde el panel de administración es posible comprar nuevos temas asignarlos según se requiera.
F009	Agregar código personalizado	En el panel de administración es posible agregar código personalizado.
F010	Previsualización de página	Se puede tener una vista previa de la página desde el panel de administración.

5. Diagrama de Arquitectura:

[Link](#) al Diagrama de Arquitectura^[1].

El diagrama de arquitectura muestra como eje principal el API REST sobre la que se genera interacción con Ghost, llamada Ghost Core, la cual toma como apoyo funcionalidades de una capa de aplicativos SQL (ORM Layer) para el manejo y almacenamiento de la data. Adicionalmente, utiliza adaptadores para almacenamiento (Storage Adapter) dentro de los que se resaltan Azure y S3 entre otros. El API REST igualmente tiene interacción con el Ghost Admin y el Frontend para manejo de contenido y despliegue de estos al usuario final.

Obtenido de <https://ghost.org/docs/architecture/>

6. Diagrama de Contexto:

[Link](#) al Diagrama de Contexto

El diagrama de contexto muestra como usuario del aplicativo a los bloggers independientes al igual que a los equipos grandes de escritores y editores, ya que, según la página principal de Ghost estos son los principales usuarios de la herramienta, la cual a su vez ofrece componentes como Ghost Admin y Ghost Core para la creación y administración de contenido. Ghost a su vez, permite la expansión de funcionalidades a través de

extensiones desarrolladas por externos. A través de Ghost entonces, se despliega contenido que consume el usuario final.

7. Modelo de Datos:

[Link](#) al Modelo de Datos.

8. Modelo de GUI:

[Link](#) al Modelo de GUI (se incluye carpeta con imágenes a las vistas).

1. Contexto de la estrategia de pruebas

1.1. Objetivos:

- Implementar un proceso de pruebas tomando en cuenta pruebas de tipo de sistema, aceptación y funcionales.
- Desarrollar pruebas exploratorias manuales de las principales funcionalidades de la aplicación.
- Desarrollar pruebas de reconocimiento en las funcionalidades principales de la aplicación.
- Desarrollar pruebas e2e en las funcionalidades principales de la aplicación.
- Desarrollar pruebas de regresión visual tomando en cuenta la versión 3.42.5 de Ghost y la actual.
- Desarrollar pruebas e2e implementando generación de datos en las principales funcionalidades de la aplicación.
- Identificar errores, defectos y puntos de mejora en la aplicación.

1.2. Duración de la iteración de pruebas:

		07/06/21	14/06/21	21/06/21	28/06/21	5/07/21	12/07/21	19/07/21
Distribución de Esfuerzo	31/05/21 - 4/06/21	- 11/06/21	- 18/06/21	- 25/06/21	- 2/07/21	- 9/07/21	- 16/07/21	- 23/07/21
Iteración 1	32 horas	32 horas						
Iteración 2			32 horas	32 horas				

Iteración 3					32 horas	32 horas		
Iteración 4							32 horas	32 horas

1.3. Presupuesto de pruebas:

1.3.1. Recursos Humanos

Para este proceso se cuenta con 4 testers senior con conocimientos y habilidades equitativas. Cada tester tiene 8 horas semanales para trabajar en este proceso, dando un total de 32 horas a la semana entre los 4.


Nombre	Experiencia
Ginett Rosales	Tester Senior, conocimiento avanzado en pruebas automatizadas, manejo de frameworks de Api's de automatización de pruebas, conocimiento sobre patrones de pruebas, conocimiento avanzado en técnicas de pruebas automatizadas. Conocimientos en herramientas de pruebas de reconocimineto automaticas.
María Gaitán	Tester Senior, conocimiento avanzado en pruebas automatizadas, manejo de frameworks de Api's de automatización de pruebas, conocimiento sobre patrones de pruebas, conocimiento avanzado en técnicas de pruebas automatizadas. Conocimientos en herramientas de pruebas de reconocimineto automaticas.
Iván Amarillo	Tester Senior, conocimiento avanzado en pruebas automatizadas, manejo de frameworks de Api's de automatización de pruebas, conocimiento sobre patrones de pruebas, conocimiento avanzado en técnicas de pruebas automatizadas. Conocimientos en herramientas de pruebas de reconocimineto automaticas.
Carlos Pinto	Tester Senior, conocimiento avanzado en pruebas automatizadas, manejo de frameworks de Api's de automatización de pruebas, conocimiento sobre patrones de pruebas, conocimiento avanzado en técnicas de pruebas

automatizadas. Conocimientos en herramientas de pruebas de reconocimineto automaticas.

1.3.2. Recursos Computacionales

▪ Recursos físicos

- 4 computadores con las siguientes prestaciones



Computador Portátil ASUS ExpertBook 14" Pulgadas B5440FA-BM0819R Procesador Intel Core i5 - 8 GB RAM - Disco Estado Sólido 512 GB - Gris
★★★★☆ (9)

Especificaciones	
Memoria RAM	8 GB
Capacidad Disco Estado Sólido	512 GB
Tipo Procesador	Intel Core i5
Tamaño Pantalla	14 Pulgadas

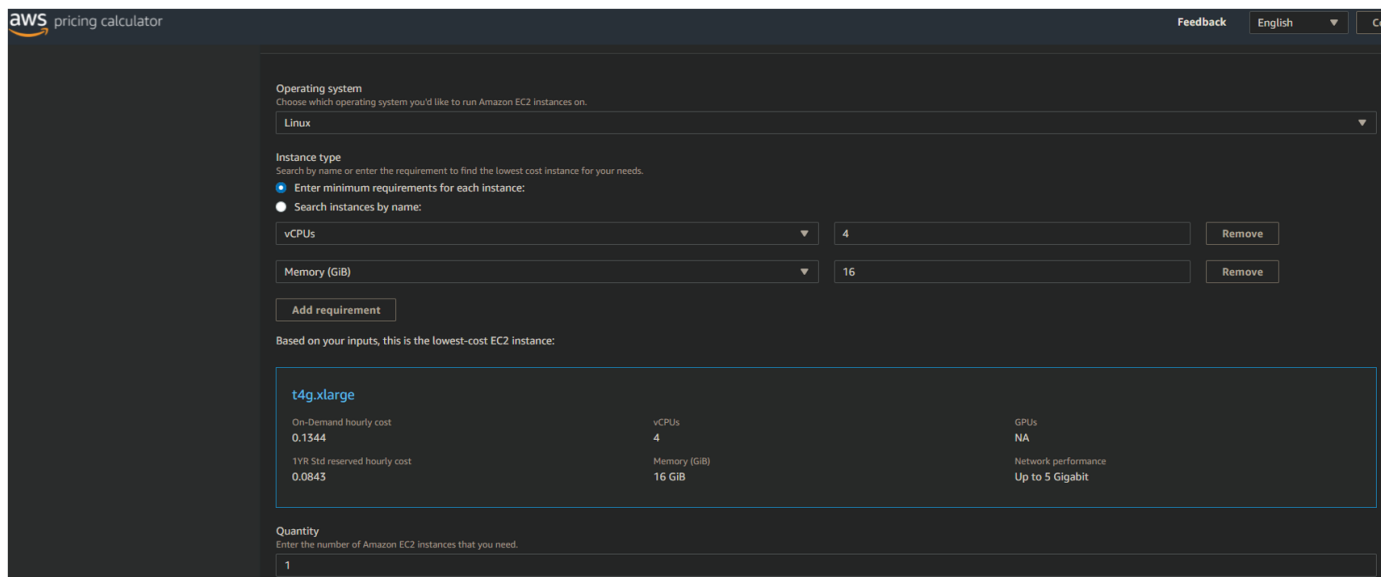
Características destacadas	
o	Dureza de grado militar MIL-STD 810G
o	Chasis ultradelgado y ultraliviano
o	Ángulos de visión flexibles con bisagra plana de 180°
o	Alta conectividad integral de puertos

▪ Herramientas

- Cypress 7.0.1
- RUPuppet 1.0.0
- Resemble
- Faker 5.5.3
- Mockaroo (<https://mockaroo.com/>)

▪ Servidor en la Nube

- Para la ejecución de las pruebas automáticas se utilizará una instancia de Amazon Web Service (AWS) la cual estimamos requerir por un total de 200 horas. Estas horas incluirán un tiempo de ejecución de pruebas de integración a lo largo de la iteración y con espacio suficiente para instalar diferentes sistemas operativos (macOS, Ubuntu, Windows10) y apalancar con ello la ejecución de las pruebas en diferentes sistemas operativos y browsers en conjunto con BrowserStack.



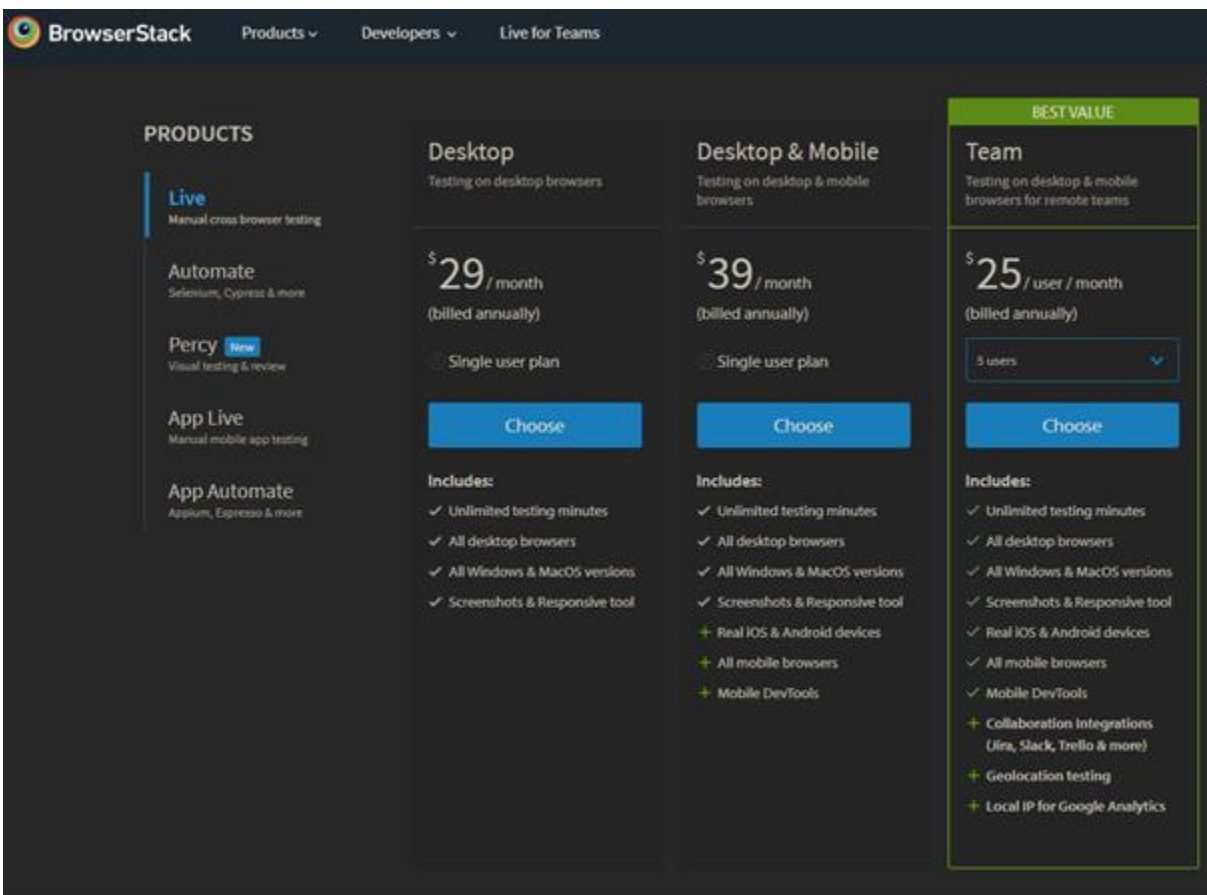
The screenshot shows the AWS Pricing Calculator interface. Under 'Operating system', 'Linux' is selected. Under 'Instance type', 'Enter minimum requirements for each instance:' is selected. Two requirements are added: 'vCPUs' set to 4 and 'Memory (GiB)' set to 16. The calculator recommends the 't4g.xlarge' instance type. The pricing table shows an on-demand hourly cost of 0.1344 and a 1-year standard reserved hourly cost of 0.0843. The instance has 4 vCPUs, 16 GiB of memory, and up to 5 Gigabit network performance.

On-Demand hourly cost	vCPUs	GPUs
0.1344	4	NA

1YR Std reserved hourly cost	Memory (GiB)	Network performance
0.0843	16 GiB	Up to 5 Gigabit

Obtenido de <https://calculator.aws/#/createCalculator/EC2>

- Por último, para el desarrollo del proyecto se requiere hacer uso de un Browser Stack con el fin de ejecutar las pruebas de funcionalidad en diferentes OS y Browsers.



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

- Para la contratación del servicio de AWS, se cotizó un presupuesto de \$94.900 pesos colombianos.
- Para la contratación de Browser Stack, se cotizó un presupuesto de \$142.350 pesos colombianos.

La suma de estos dos servicios que se requieren contratar nos da un total de \$237.250 pesos colombianos o \$63.83 dólares.

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

Nivel	Tipo	Técnica	Objetivo
-------	------	---------	----------

Pruebas de Sistema	Funcional	Pruebas exploratorias manuales	Entender la funcionalidad del sistema y su comportamiento. Con el resultado se puede definir un oráculo.
Pruebas de Sistema	Funcional	Reconocimiento con herramientas automáticas (Monkey Testing, Rippers)	Ayuda a detectar comportamientos no deseados o inesperados de la aplicación desde eventos aleatorios.
Pruebas de Sistema y de Aceptación	Funcional	Pruebas de extremo a extremo (e2e) con API's de automatización. Cypress	Validar escenarios de pruebas de las funcionalidades como un conjunto, simulando un ambiente real de uso.
Pruebas de Sistema y de Aceptación	Funcional	Regression visual con API's de automatización (ResembleJS)	Validar si el estado actual con una versión anterior. Con esto, se puede determinar si la aplicación está funcionando correctamente según los requisitos definidos.
Pruebas de Sistema y de Aceptación	Positivas y negativas	Pruebas de extremo a extremo (e2e) con API's de automatización. Cypress, Faker, Mockaroo	Validar escenarios de pruebas de las funcionalidades como un conjunto, simulando un ambiente real de uso pero con datos de entrada correctos e incorrectos.

Inicialmente se realizará las pruebas exploratorias manuales con el fin de identificar la funcionalidad del sistema, con los resultados obtenidos se realizará la creación del oráculo explícito para la creación de escenarios de pruebas.

Las pruebas de reconocimiento se hará uso de Monkey Tests y de Rippers corriendo en AWS con el fin de aprovechar los recursos tecnológicos disponibles y a su vez liberar al equipo para el desarrollo de pruebas e2e con Cypress. Se escoge dicha estrategia por su facilidad en implementación, por su facilidad en ejecución y porque permite ir corriendo las pruebas de forma autónoma liberando al equipo de las pruebas manuales.

Durante el desarrollo del proyecto, el equipo deberá crear reportes donde se detallen los resultados de las pruebas, esto debe incluir los puntos de mejora y defectos encontrados durante el proceso.

1.5. Distribución de Esfuerzo

			Actividades		Actividades	
Semana	Total Horas / Persona	Iteración	GINETT ROSALES	MARÍA GAITÁN	IVÁN AMARILLO	CARLOS PINTO
Semana 1	8 horas	Iteración 1	Ejecución pruebas exploratorias manuales	Ejecución pruebas exploratorias manuales	Configuración de pruebas de reconocimiento y monitoreo de resultados (monkeys)	Configuración de pruebas de reconocimiento y monitoreo de resultados (rippers)
Semana 2	8 horas	Iteración 1	Creación del oráculo explícito	Creación del oráculo explícito	Generación de reporte de resultados pruebas de reconocimiento	Generación de reporte de resultados pruebas de reconocimiento
Semana 3	8 horas	Iteración 2	Desarrollo de pruebas E2E en Cypress (Posts)	Desarrollo de pruebas E2E en Cypress (Staff)	Desarrollo de pruebas E2E en Cypress (Pages)	Desarrollo de pruebas E2E en Cypress (Tags)
Semana 4	8 horas	Iteración 2	Desarrollo de pruebas E2E en Cypress (Posts)	Desarrollo de pruebas E2E en Cypress (Staff)	Desarrollo de pruebas E2E en Cypress (Pages)	Desarrollo de pruebas E2E en Cypress (Tags)
Semana 5	8 horas	Iteración 3	Implementación de Resemble JS (VRT)	Implementación de Resemble JS (VRT)	Recopilación de datos y generación de reportes	Recopilación de datos y generación de reportes
Semana 6	8 horas	Iteración 3	Ejecución de VRT	Ejecución de VRT	Ejecución de VRT	Ejecución de VRT
Semana 7	8 horas	Iteración 4	Desarrollo de pruebas E2E con generación de datos (pool de datos)	Desarrollo de pruebas E2E con generación de datos (pool de datos)	Desarrollo de pruebas E2E con generación de datos (aleatorios)	Desarrollo de pruebas E2E con generación de datos (aleatorios)
Semana 8	8 horas	Iteración 4	Consolidación y presentación de resultados	Consolidación y presentación de resultados	Consolidación y presentación de resultados	Consolidación y presentación de resultados

Distribución de Esfuerzo								
	S1	S2	S3	S4	S5	S6	S7	S8
Iteración 1	Pruebas Exploratorias y de reconocimiento	Creación de oráculo y generación de reportes						

Iteración 2			Desarroll o de pruebas E2E	Desarrollo de pruebas E2E				
Iteración 3					Implemen tación VRT y recopilaci ón de datos	Ejecución VRT		
Iteración 4							Desarrollo pruebas E2E con pool de datos y datos aleatorios	Consolidaci ón y presentació n de resultados