

Estrategia de Pruebas

1. Aplicación Bajo Pruebas

1.1. Nombre Aplicación: Ghost

1.2. Versión: 3.42.5

1.3. Descripción:

Ghost es una aplicación open source para gestionar un blog, permitiendo a cualquier persona convertirse en un editor y publicar contenido por medio de posts. Está desarrollado en Node.js y cuenta con un RESTful JSON API, una zona de administración y un front-end público totalmente customizable. El front-end está desarrollado en Handlebars.js y Ember.js.

1.4. Funcionalidades Core:

Zona de administración:

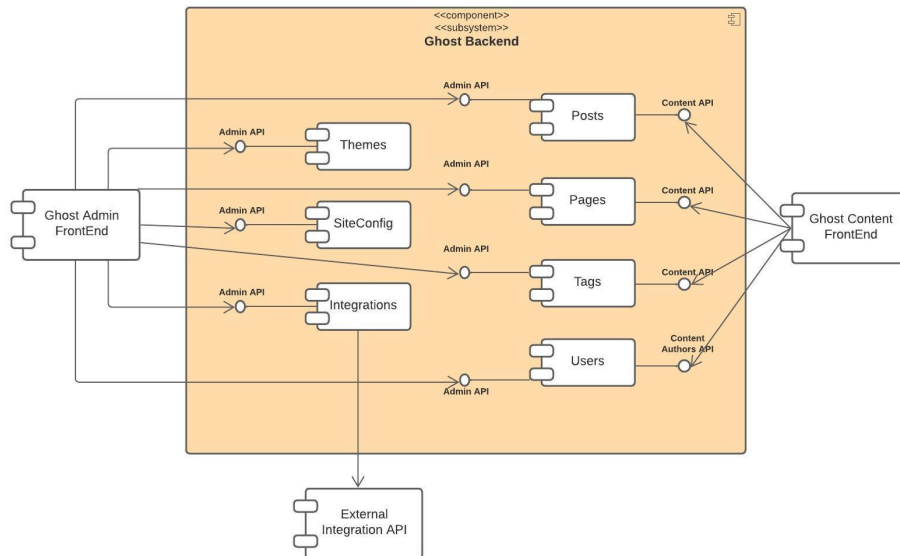
- **Id.1 Login**
Como administrador quiero ingresar mi usuario y contraseña para iniciar sesión como administrador
- **Id.2 Invitar persona**
Como administrador quiero invitar a una persona por email para que pueda publicar en el aplicativo.
- **Id.3 Crear nuevo post**
Como administrador quiero crear un nuevo post para poder publicarlo
- **Id.4 Publicar post**
Como administrador quiero publicar un post para que pueda ser visto públicamente.
- **Id.5 Cambiar configuración post**
Como administrador quiero cambiar la configuración del post para poder cambiar sus atributos.
- **Id.6 Crear Tag**
Como usuario quiero crear un Tag para poder clasificar posts
- **Id.7 Crear nueva página**
Como administrador quiero crear una página para que los visitantes a ghost pueda verla
- **Id.8 Cambiar configuración página**
Como administrador quiero cambiar la configuración de una página para poder cambiar sus atributos.
- **Id.9 Publicar nueva página**
Como administrador quiero publicar una nueva página para que los visitantes a ghost puedan verla
- **Id.10 Agregar integración**
Como administrador quiero agregar una integración con un aplicativo externo para que se comunique con Ghost

- **Id.11 Cambiar configuración del aplicativo**

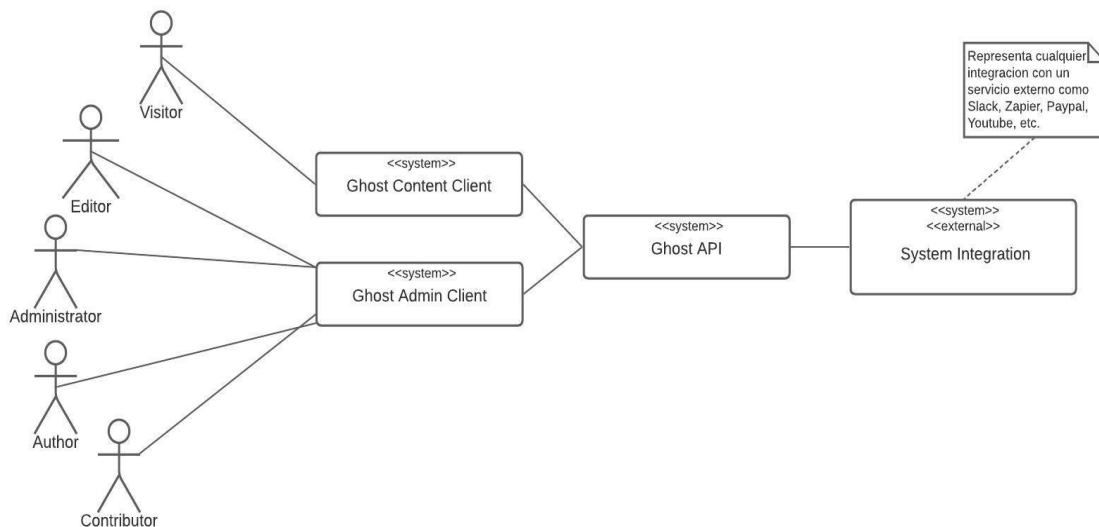
Como administrador quiero cambiar la configuración de mi aplicativo para poder cambiar sus valores.

1.5. Diagrama de Arquitectura:

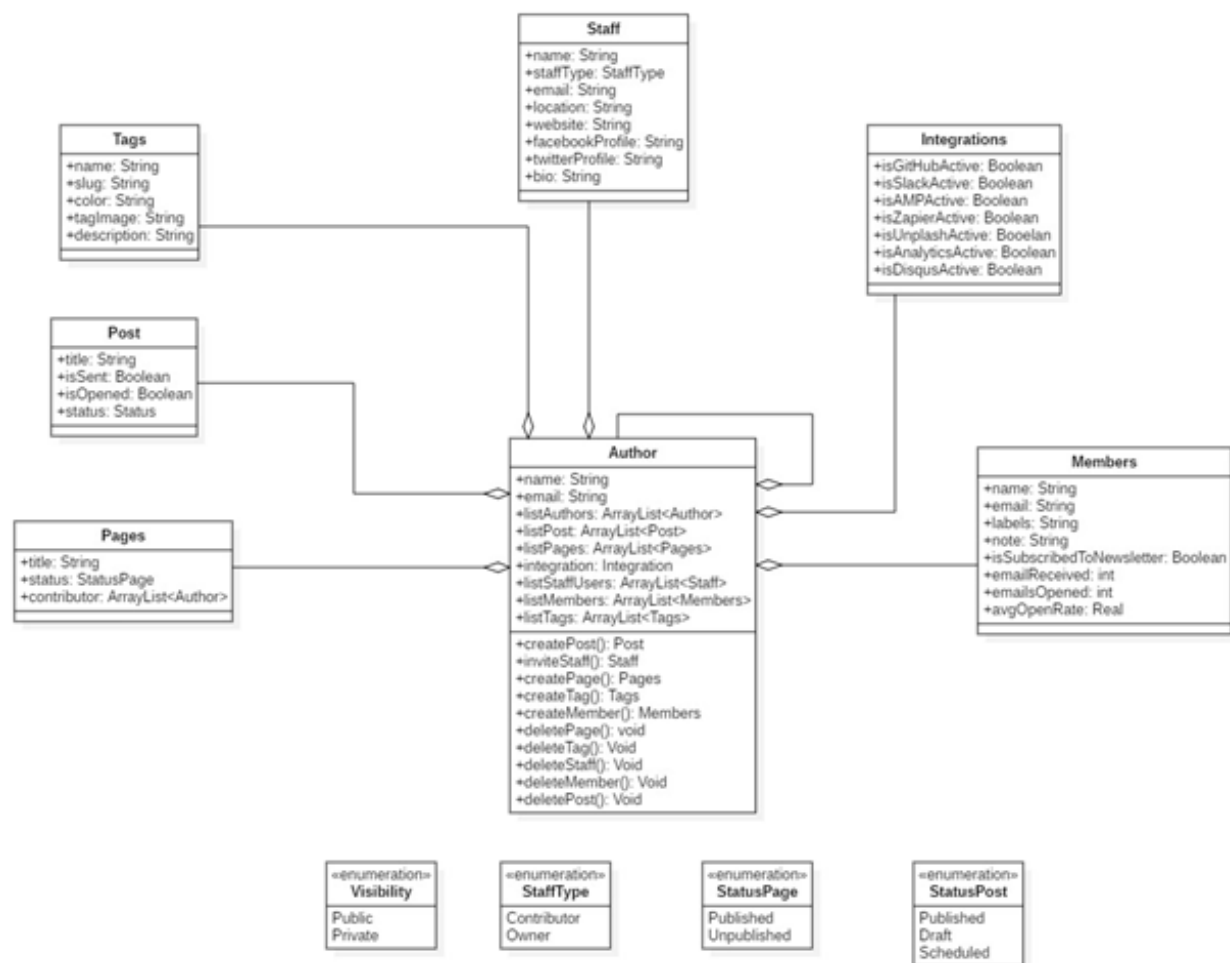
Se presenta un diagrama funcional.



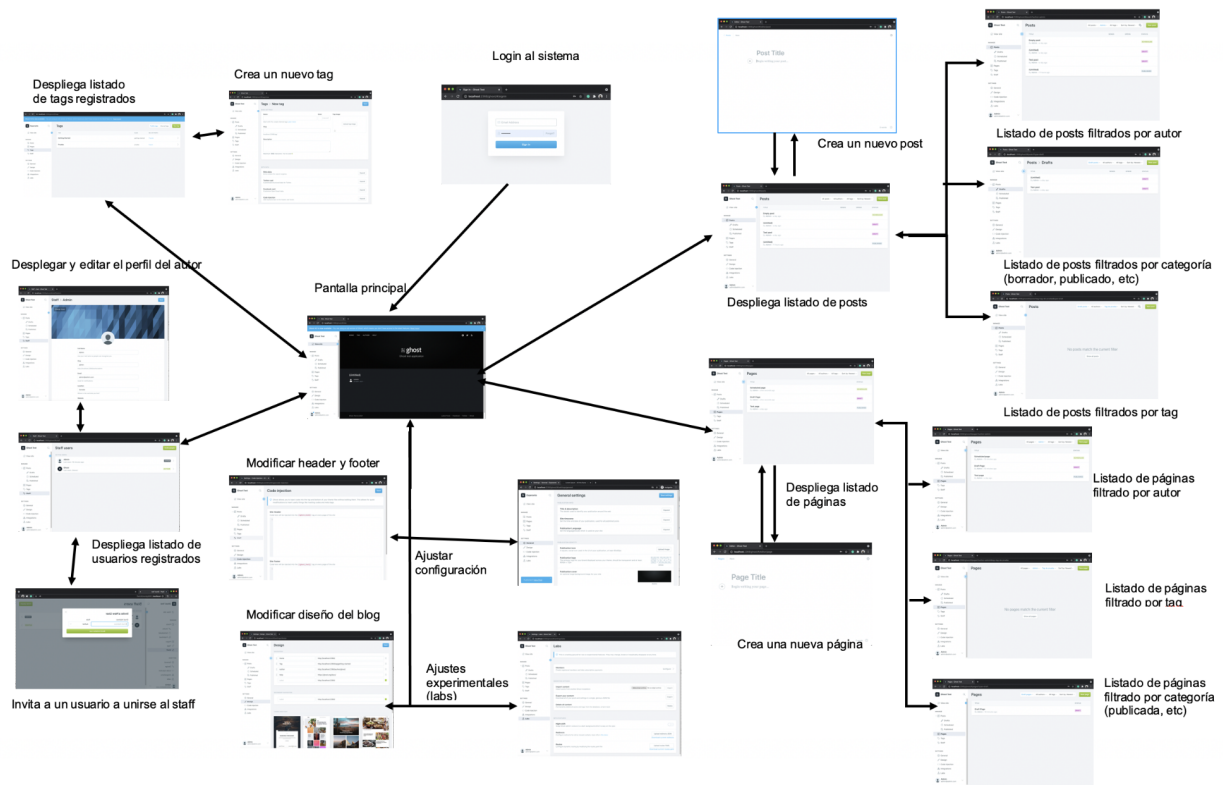
1.6. Diagrama de Contexto:



1.7. Modelo de Datos:



1.8. Modelo de GUI:



2. Contexto de la estrategia de pruebas

2.1. Objetivos:

- **O1** Detectar defectos en la aplicación bajo pruebas y familiarizarnos con el aplicativo por medio del uso de pruebas exploratorias manuales. Esto se realizará en un entorno similar al de producción.
- **O2** Realizar 60 horas de pruebas exploratorias autónomas en AWS(PE-1) para encontrar posibles comportamientos no deseados sobre el aplicativo. Para eso se usará pruebas aleatorias con la herramienta monkey-cypress y pruebas sistemáticas con la herramienta RUPuppet, cada una en un proceso en paralelo.
- **O3** Realizar pruebas E2E para automatizar las principales funcionalidades de GHOST. Las herramientas a utilizar serán Playwright y kraken-mobile.
- **O4** Se realizarán pruebas de regresión visual VRT para detectar cambios en la interfaz de usuario en cada nueva versión desarrollada. La herramienta a utilizar será ResembleJS.
- **O5** Se realizarán pruebas de validación de datos con ayuda de las pruebas E2E desarrolladas. Se generará un pool de datos a-priori y aleatorio con ayuda de la herramienta mackaroo.

2.2. Duración de la iteración de pruebas :

La duración de la estrategia de distribución de pruebas es de 8 semanas.

2.3. Presupuesto de pruebas:

2.3.1. Recursos Humanos

Se cuenta con 4 testers senior.

| Nombre | Cargo | Experiencia | Tiempo disponible |
|------------------------------------|--|---|-------------------|
| Wenceslao Crhistopher Paez Chavez | Ingeniero automatizador senior y coordinador del equipo de pruebas | Javascript, Cypress, Playwright, Cucumber, , Kraken, Capybara, Pruebas Manuales | 64 horas |
| Ivan Dario Peñaloza Rojas | Tester senior | Cypress, Playwright | 64 horas |
| Manuel Alejandro Sanchez Masferrer | Tester senior | Cypress, Playwright, Protractor, | 64 horas |
| Christtian Alfredo Manzo Parra | Tester senior | Cypress, Playwright | 64 horas |

2.3.2. Recursos Computacionales

- Cada uno de los tester senior cuenta con un computador con sistema operativo Linux.
- Cada uno de los computadores cuenta con un mínimo de memoria RAM de 8 GB, y un mínimo de disco duro de 500 GB.
- Cada uno de los equipos de computación tiene un espacio libre de 2 GB en memoria para poder correr las pruebas consideradas en la estrategia.
- Cada uno de los equipos puede ser usado hasta por 100 horas semanales.

Para realizar las pruebas manuales en un entorno similar al de producción llamado Staging-1, requerimos la contratación de los siguientes servidores:

| Servicio | Horas | Uso |
|------------------------------|--|------------------------------|
| Amazon EC2 <i>t4g.xlarge</i> | 60 horas(50 uso, 10 de configuracion) | Servidor de aplicación y web |
| Amazon RDS | 50 horas(50 uso) | Base de datos |

| | | |
|--------------------|--|-------------------------------|
| <i>db.t2.small</i> | | |
| Amazon SES | Free tier (Hasta 62000 emails mensuales) | Envío de emails de invitación |

Para la canalización de pruebas exploratorias autónomas se requerirá el server PE-1:

| Servicio | Horas | Uso |
|---------------------------------|--|---------------------|
| Amazon EC2 <i>t4g.medium</i> | 60 horas(50 uso, 10 de configuracion) | Servidor de pruebas |

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

Dado que los 4 tester senior son parte de la empresa, y estamos haciendo uso de nuestros propios equipos de cómputo, no requerimos de ningún gasto adicional en servicios, y ningún tipo de contratación externa.

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

Automatizadas

| Nivel | Tipo | Técnica | Objetivo |
|--|---|---|----------|
| Sistema: Se evaluará el sistema como un todo | Caja negra: Se evalúa las salidas del sistema Funcional | Pruebas de exploración autónomas con monkey-cypress y RIPuppet | O2 |
| Sistema: Se evaluará el sistema como un todo | Caja negra: Se evalúa las salidas del sistema Funcional Positivas y negativas | APIs de automatización con Playwright y kraken-mobile | O3 |
| Sistema: Se evaluará el sistema como un todo | Caja negra: Se evalúa las salidas del sistema Funcional Positivas y negativas | Pruebas de regresión VRT con la herramienta ResembleJS | O4 |
| Sistema: Se evaluará el sistema como un todo | Caja negra Funcionales Escenarios positivos y negativos | APIs de automatización con validación de datos. Uso de mackaroo | O5 |

Manual

| Nivel | Tipo | Técnica | Objetivo |
|---|-------------------------------------|--|----------|
| Aceptación : Se recrea un entorno staging-1 similar al de producción y se realizarán las pruebas como perspectiva de usuario. | Funcional Caja negra Positiva | Pruebas exploratorias Pruebas de regresión manuales | O1 |
| Sistema : Se recrea un entorno staging-1 donde se busca encontrar defectos. | Funcional Caja negra Negativa | Pruebas exploratorias Pruebas de regresión manuales | O1 |

2.5. Distribución de Esfuerzo

Semana 1:

- Diseño de la estrategia de pruebas

| Recurso humano | Horas |
|----------------|-------|
| Wenceslao | 2 |
| Ivan | 2 |
| Manuel | 2 |
| Christtian | 2 |

- Pruebas exploratorias manuales:

Las pruebas exploratorias se realizarán en el entorno Staging-1 y en el computador personal de cada miembro.

| Recurso humano | Horas |
|----------------|-------|
| Wenceslao | 6 |
| Ivan | 6 |
| Manuel | 6 |
| Christtian | 6 |

Semana 2:

- Configuración para pruebas de exploración automatizadas.

| Recurso humano | Horas |
|----------------|-------|
| Wenceslao | 8 |
| Ivan | 8 |

- Registro de incidencias reportadas por pruebas de exploración automatizadas.

| Recurso humano | Horas |
|----------------|-------|
| Manuel | 8 |
| Christtian | 8 |

Semana 3:

- Desarrollo de pruebas E2E con Playwright.

| Recurso humano | Horas |
|----------------|-------|
| Wenceslao | 8 |
| Ivan | 8 |
| Manuel | 8 |
| Christtian | 8 |

Semana 4:

- Desarrollo de pruebas E2E con kraken-mobile.

| Recurso humano | Horas |
|----------------|-------|
| Wenceslao | 8 |
| Ivan | 8 |
| Manuel | 8 |

| | |
|------------|---|
| Christtian | 8 |
|------------|---|

Semana 5:

- Pruebas de regresión VRT

| Recurso humano | Horas |
|----------------|-------|
| Wenceslao | 6 |
| Ivan | 6 |
| Manuel | 6 |
| Christtian | 6 |

- Pruebas de regresión manuales

| Recurso humano | Horas |
|----------------|-------|
| Wenceslao | 2 |
| Ivan | 2 |
| Manuel | 2 |
| Christtian | 2 |

Semana 6:

- Análisis de pruebas de regresión VRT

| Recurso humano | Horas |
|----------------|-------|
| Wenceslao | 2 |
| Ivan | 2 |
| Manuel | 2 |
| Christtian | 2 |

- Pruebas con generación de datos en Playwright

| Recurso humano | Horas |
|----------------|-------|
| Wenceslao | 6 |

| | |
|------------|---|
| Ivan | 6 |
| Manuel | 6 |
| Christtian | 6 |

Semana 7:

- Pruebas con generación de datos en kraken-mobile

| Recurso humano | Horas |
|----------------|-------|
| Wenceslao | 8 |
| Ivan | 8 |
| Manuel | 8 |
| Christtian | 8 |

Semana 8:

- Reporte de resultados e ingreso de incidencias

| Recurso humano | Horas |
|----------------|-------|
| Wenceslao | 4 |
| Ivan | 4 |
| Manuel | 4 |
| Christtian | 4 |

- Video, retrospectiva y otros.

| Recurso humano | Horas |
|----------------|-------|
| Wenceslao | 4 |
| Ivan | 4 |

| | |
|------------|---|
| Manuel | 4 |
| Christtian | 4 |

Cuadro de tipos de funcionalidades realizadas por tipo de prueba:

| Id funcionalidad | id Escenario | Pruebas exploratorias | P. E2E | VRT | V. Datos |
|-------------------------|------------------------------|------------------------------|---------------|------------|-----------------|
| 1 | P01-Login | X | | | |
| 2 | P02-Inv-Per | | | | X |
| 3 | P03A-Mk-post P03B-Mk-post | | X | X | X |
| 4 | P04A-Ps-post P04B-Ps-post | | X | X | |
| 5 | P05-Cf-post | X | | | |
| 6 | P06A-Mk-test P06B-Mk-test | | X | X | X |
| 7 | P07A-Mk-pag P07B-Mk-pag | | X | X | X |
| 8 | P05-Cf-pag | X | | | |
| 9 | P09A-Ps-pag P09B-Ps-pag | | | X | |
| 10 | P010-Cf-inte | X | | | |
| 11 | P011-Cf-app | X | | | |