Estrategia de Pruebas

1. Aplicación Bajo Pruebas

1.1. **Nombre Aplicación:** Ghost

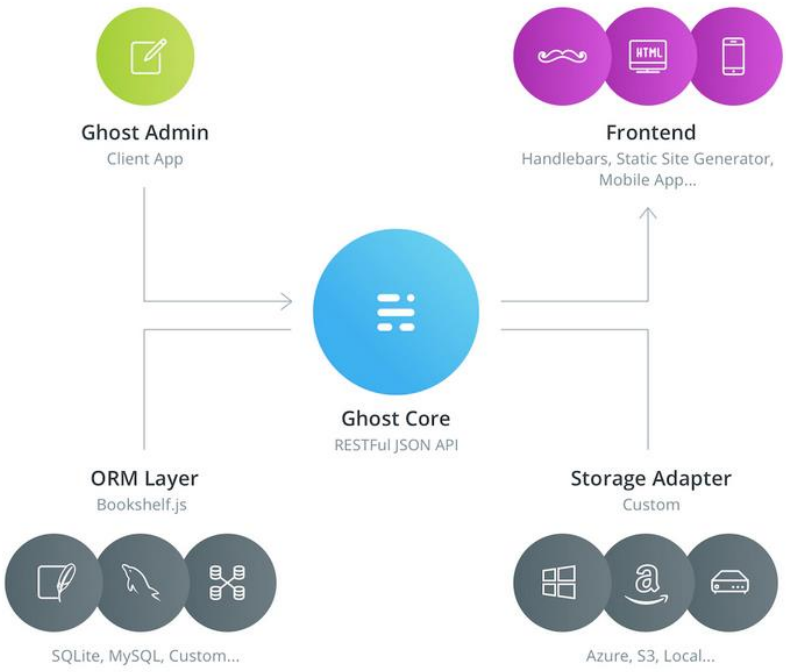
1.2. **Versión:** 4.42.0

1.3. **Descripción**: Ghost es un CMS orientado a generadores de contenido con opciones de publicación, edición, marcado y compartición de contenido.

1.4. **Funcionalidades Core**:

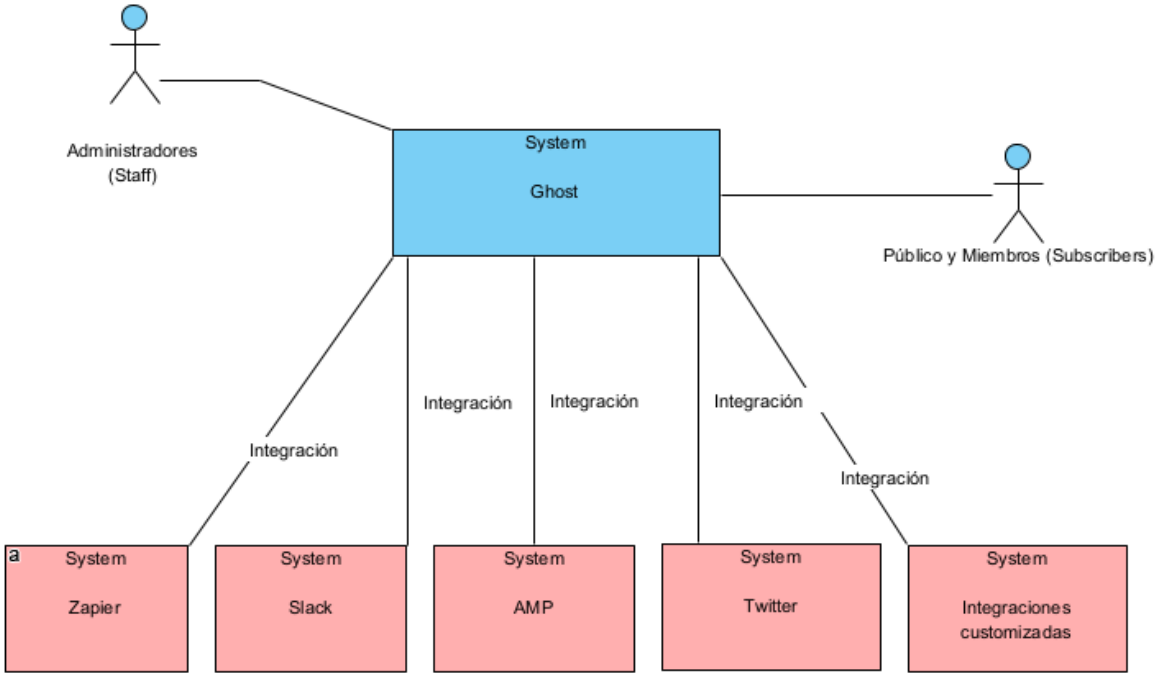
- Administración
  - Inicio de sesión
  - Escribir y publicar contenido (Posts), manejar el estado del contenido y publicar de manera programada (Schedule), etiquetar Posts.
  - Escribir y publicar páginas, publicarlas, etiquetarlas y administrar metadatos.
  - Administrar etiquetas
  - Administrar miembros de la comunidad, crearlos, importarlos masivamente.
  - Configuración: Permite administrar la configuración general de publicación y metadatos, el tema y propiedades de diseño, navegación y la lista de usuarios administradores, parámetros de administración de la membresía y envío de correos, administración de integraciones con otras aplicaciones.
- Público:
  - Navegación y consulta por el contenido.
  - Suscripción a la membresía.
  - Inicio de session.

1.5. **Diagrama de Arquitectura:**

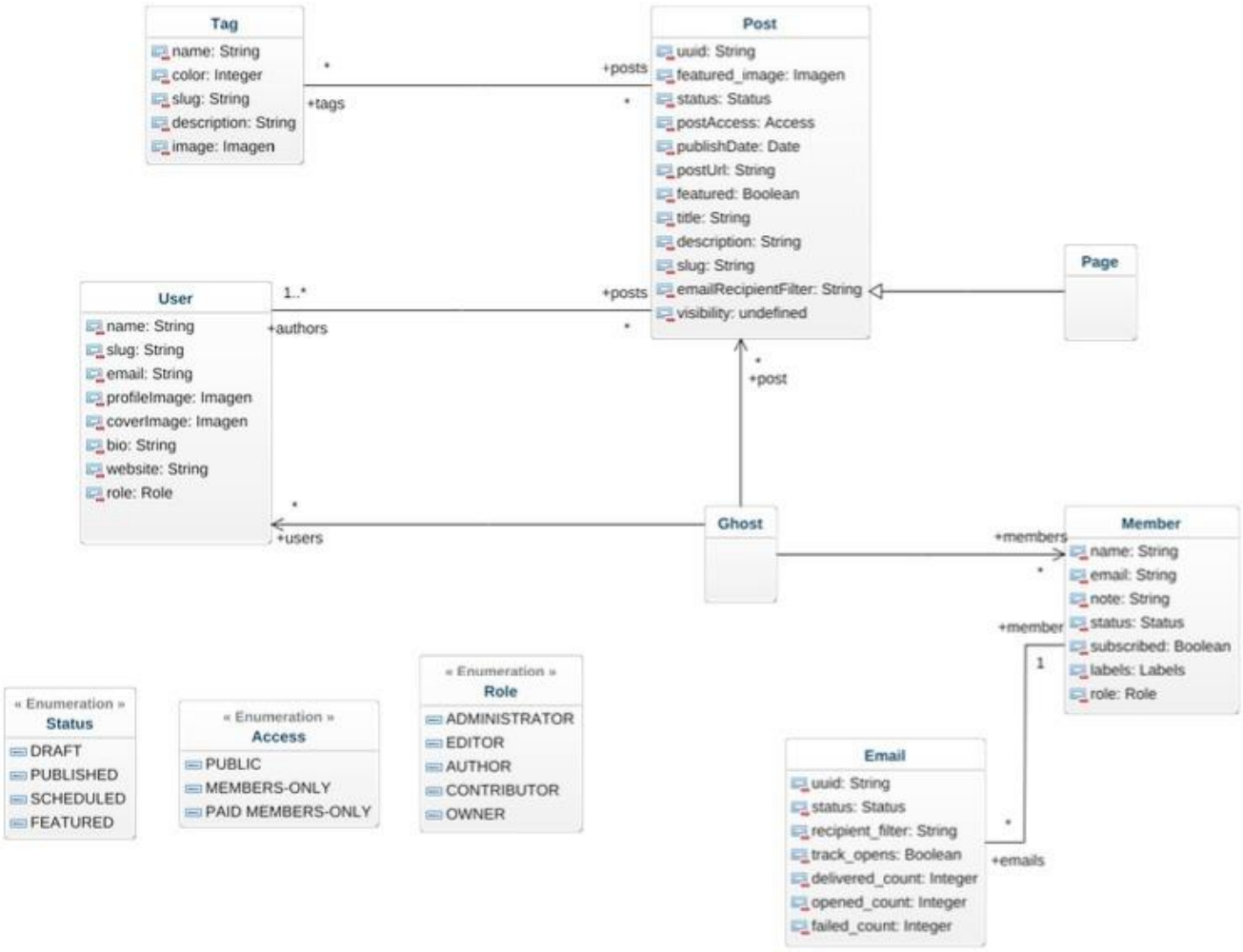


La descripción completa de la arquitectura se encuentra en https://ghost.org/docs/architecture/

1.6. Diagrama de Contexto:



1.7. Modelo de Datos:



1.8. Modelo de GUI:

El Modelo de GUI puede ser consultado directamente en https://miro.com/app/board/uXjVO_hS6-0=/

2. Contexto de la estrategia de pruebas

Se parte del supuesto de que cada desarrollador ha realizado sus respectivas pruebas unitarias, de tal manera que se tiene un coverage mínimo del 80%.

2.1. Objetivos:

En esta iteración de pruebas con una iteración máxima de 2 meses se identifican los siguientes objetivos:

1. Se desea alcanzar una cobertura de pruebas funcionales con un porcentaje superior al 90% sobre el funcionamiento total del aplicativo esto con el fin de cumplir con los estándares de calidad establecidos para identificar las partes críticas del código.

2. Establecer que la navegación entre las interfaces del sistema cumpla con su funcionalidad y usabilidad de acuerdo a los requisitos especificados y al flujo de negocio

3. Se desea utilizar un enfoque de verificación que permita comprender que los usuarios pueden acceder a las funciones y datos que solo tengan permitidos a su acceso.

2.2. Duración de la iteración de pruebas:

- **Pruebas exploratorias**

Dado que los testers son automatizadores, vamos a estimar un tiempo muy corto para realizar pruebas exploratorias y de esta manera tener un panorama general de la aplicación.

Horas: 120 horas (1 semana, 8 horas cada día).
Horas de máquina: 168 horas.
Fecha: Lunes, 30 de mayo.

- **Pruebas de reconocimiento**

Con estas pruebas se busca encontrar errores a través de pruebas aleatorias.

Horas: 120 horas (1 semana, 8 horas cada día).
Horas de máquina: 168 horas.
Fecha: Lunes, 06 de junio.

- **Pruebas de E2E**

Con estas pruebas se busca validar el correcto funcionamiento del sistema

Horas: 240 horas (2 semanas, 8 horas cada día).
Horas de máquina: 336 horas.
Fecha: Lunes, 13 de junio.

- **Pruebas VRT**

Con estas pruebas se busca hacer regresión entre dos versiones

Horas: 240 horas (2 semanas, 8 horas cada día).
Horas de máquina: 336 horas.
Fecha: Lunes, 27 de junio.

- **Pruebas de validación de datos**

Con estas pruebas se busca validar la entrada de datos

Horas: 240 horas (2 semanas, 8 horas cada día).
Horas de máquina: 336 horas.
Fecha: Lunes, 11 de julio.

2.3. Presupuesto de pruebas:

2.3.1. Recursos Humanos

Los testers son senior con experiencia de 4 años, tienen un recorrido en proyectos trabajados con:

- Pruebas de integración en tecnologías como testuff
- Pruebas de carga en tecnologías como Jmeter
- Pruebas e2e en tecnologías como Cypress
- Pruebas unitarias en tecnologías como Jest, Jasmine y Karma
- Pruebas con VRT con resemble JS y Backstop

El tiempo disponible son 8 semanas repartidas en 40 horas/semanales por desarrollador

2.3.2. Recursos Computacionales

- Una máquina de AWS
    - Familia de instancia C3, con 2 sockets y 20 núcleos físicos
    - 24 Gb de RAM
    - 1 Tb de disco duro

En esta máquina estará alojado tanto el back como el front.

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

Para este caso, no hay presupuesto para la contratación de servicios tercerizados ya que nos proporcionan un capital humano específico.

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

Nivel	Tipo	Técnica	Objetivo
Sistema	Funcionales	Manual	1
Aceptación	Funcionales	Manual	1
Integración	Caja negra	Automatizada	1
Sistema	Funcionales	Manual	2
Aceptación	Funcionales	Manual	2
Integración	Caja negra, positivos y negativos	Automatizada	2, 3

2.5. Distribución de Esfuerzo

Con esta estrategia de pruebas buscamos cumplir con el patrón de pirámide que debido a los recursos limitados tiende a ser un trapecio. Sin embargo, ya que partimos del supuesto de que se tiene un 80% de coverage en pruebas unitarias se parte de una base sólida y sobre esta focalizan los esfuerzos en el cuerpo y en el pico de la pirámide.