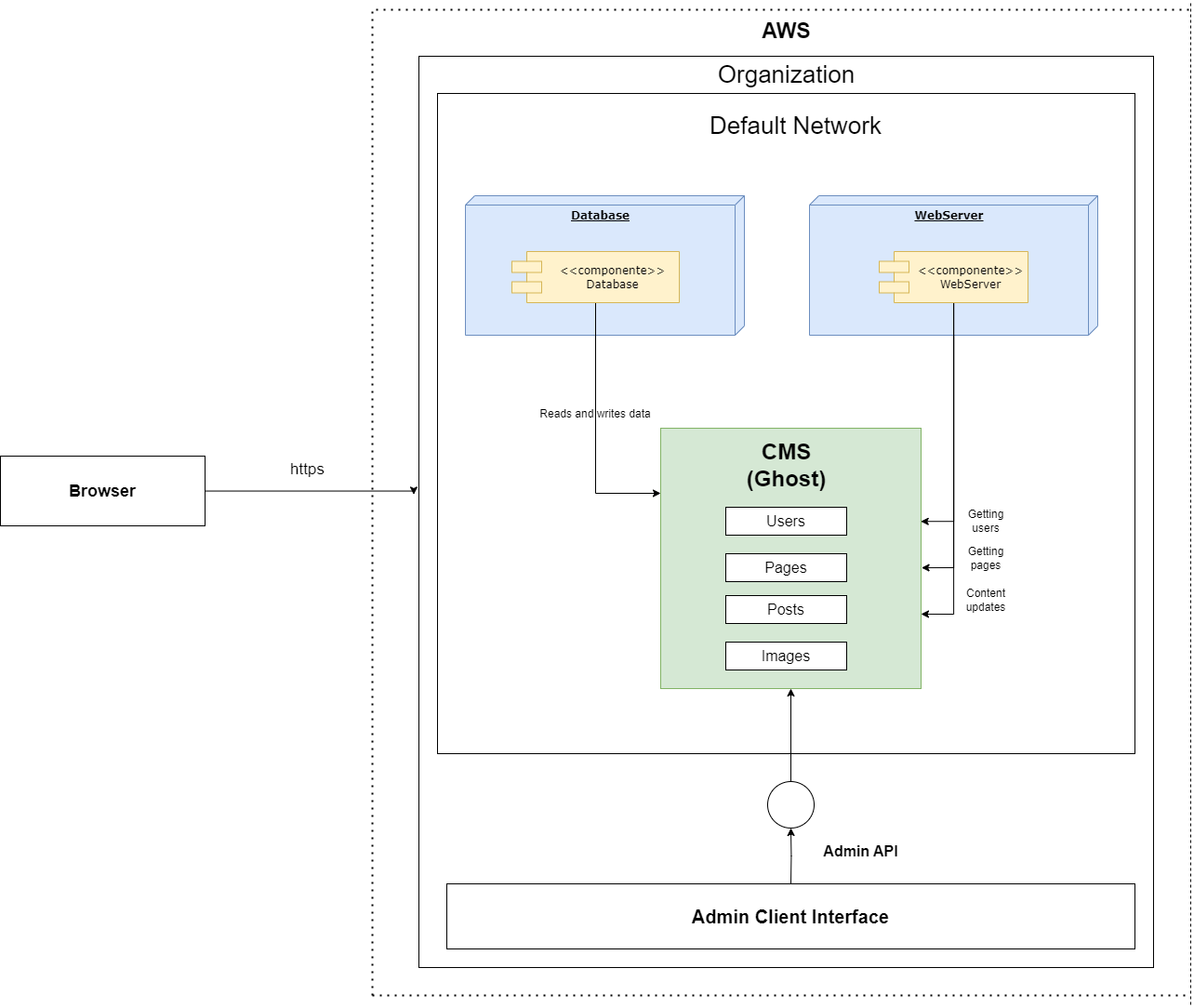
Estrategia de Pruebas

1. Aplicación Bajo Pruebas
   1. **Nombre Aplicación:** GHOST
   2. **Versión:** 1
   3. **Descripción**:

Ghost es un CMS diseñado para la publicación moderna. Su objetivo principal es ofrecer a los creadores las herramientas necesarias para desarrollar contenido en línea y asegurar que este contenido sea entregado de forma rápida y eficiente a los lectores.

* 1. **Funcionalidades Core**:
* Creación, edición, y eliminación de publicaciones.
* Creación, edición, y eliminación de páginas.
* Creación, edición, y eliminación de etiquetas
* Creación, edición, y eliminación de miembros.
* Ver listado de publicaciones
* Ver listado de páginas
* Ver listado de etiquetas por alcance
* Ver listado de miembros
* Manejo de categorías y etiquetas.
* Gestión de usuarios, roles y permisos.
* Adición, modificación y eliminación de plantillas y temas.
* Configuración SEO y parámetros del sitio.
* Extensiones, plugins e integraciones.
* Suscribirse al blog
* Ingresa.
  1. **Diagrama de Arquitectura:**



* 1. **Diagrama de Contexto:**

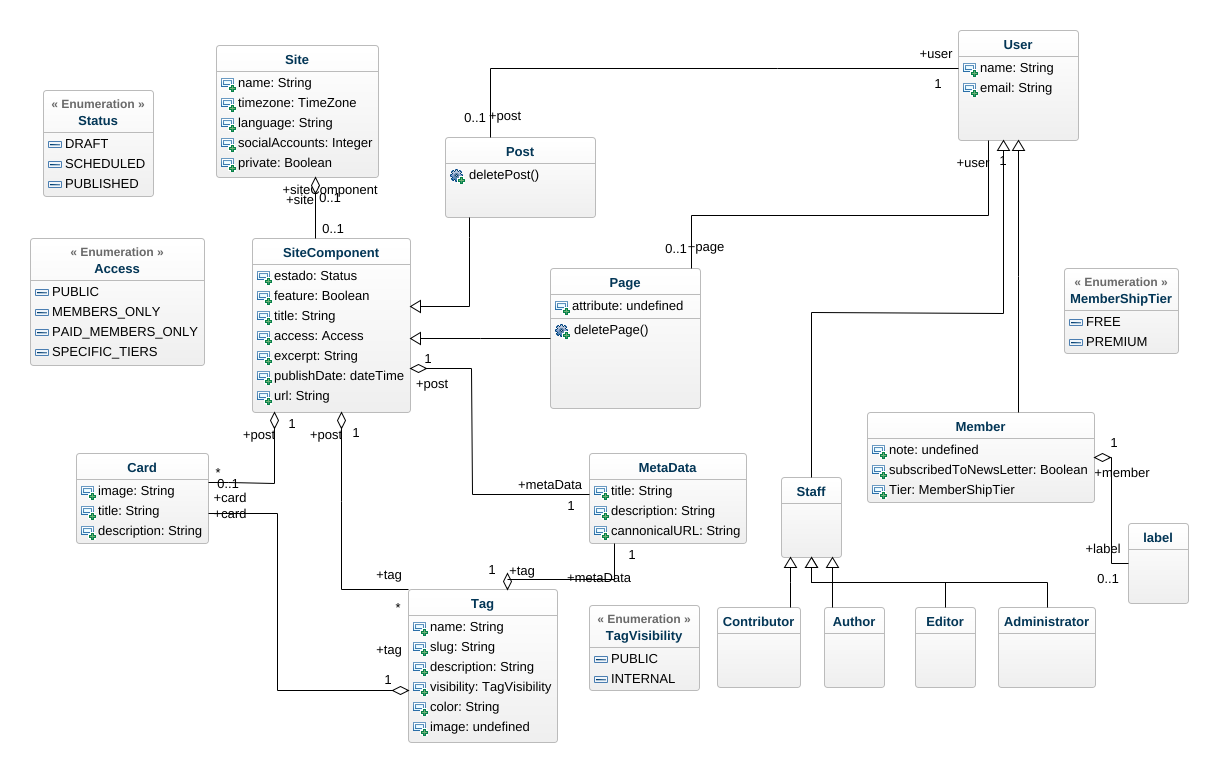
Interfaz de usuario gráfica

Descripción generada automáticamente

* 1. **Modelo de Datos:**

Diagrama, Esquemático

Descripción generada automáticamente



* 1. **Modelo GUI**

Diagrama

Descripción generada automáticamente

1. **Contexto de la estrategia de pruebas**
   1. **Objetivos:**

* Garantizar la calidad funcional de GHOST.
* Identificar defectos críticos que afecten la experiencia del usuario y reportarlos al equipo de desarrollo a través de la herramienta seleccionada GitHub - Issues.
* Validar la compatibilidad con navegadores web populares.
  1. **Duración de la iteración de pruebas:**
* Fecha inicio: 30/10/2023
* Fecha fin: 26/11/2023
* Total de horas: tres ingenieros automatizadores junior (50 horas/persona), 200 horas/máquina en Amazon AWS.
  1. **Presupuesto de pruebas:**
     1. **Recursos Humanos**
* Ingeniero 1: Graduado en Ingeniería de Software con conocimientos básicos en automatización de pruebas. Apasionado por la calidad del software y el proceso de desarrollo ágil. Busca aprender y crecer en un entorno colaborativo.

Habilidades clave:

* + Conocimiento básico de herramientas de automatización de pruebas como Cypress.
  + Comprensión de los conceptos de pruebas de software y metodologías ágiles.
  + Capacidad para escribir casos de prueba y mantener la documentación.
  + Habilidades de resolución de problemas y atención al detalle.
  + Excelentes habilidades de comunicación y trabajo en equipo.
  + En pruebas funcionales y de integración. **50 horas**.
* Ingeniero 2: Recién graduado en Ingeniería Informática con un interés especial en la automatización de pruebas de aplicaciones móviles. Motivado por optimizar la calidad de las aplicaciones y mejorar la experiencia del usuario.

Habilidades clave:

* + Conocimiento básico de lenguajes de programación.
  + Familiaridad con herramientas de automatización de pruebas.
  + Capacidad para diseñar, ejecutar y mantener pruebas automatizadas.
  + Mentalidad orientada a la resolución de problemas y aprendizaje continuo.
  + Colaborador efectivo en equipos multidisciplinarios. **50 horas**.
* Ingeniero 3: Graduado en Ingeniería de la Información con un fuerte interés en la automatización de pruebas. Comprometido con la detección temprana de errores y la mejora del código.

Habilidades clave:

* + Conocimiento básico de herramientas y técnicas de prueba.
  + Capacidad para desarrollar scripts de automatización para pruebas.
  + Comprensión de los principios de web.
  + Mentalidad ética y atención al detalle. **50 horas**.
    1. **Recursos Computacionales**
* **Amazon EC2 (Elastic Compute Cloud):**

**Descripción:** Amazon EC2 proporciona instancias virtuales escalables que permiten configurar y ejecutar entornos de pruebas personalizados. Ofrece la flexibilidad para seleccionar el tipo de instancia que se ajuste a las necesidades en términos de CPU, memoria y sistema operativo. Es ideal para ejecutar pruebas de carga, pruebas funcionales y otros tipos de pruebas que requieran un entorno personalizado.

**Capacidades:**

* **Escalabilidad:** Permite escalar recursos de manera flexible.
* **Variedad de Instancias:** Diversos tipos de instancias disponibles según las necesidades.
* **Entornos de Pruebas Personalizados:** Adecuado para ejecutar pruebas exhaustivas y personalizadas.

**Costos:** Los costos de Amazon EC2 varían según la región, tipo de instancia, duración de la instancia y otros factores. Para obtener información detallada sobre los precios, se recomienda visitar la página de precios de Amazon EC2 <https://aws.amazon.com/es/ec2/?nc2=h_ql_prod_fs_ec2.>

**Uso:** 100 horas a instancias EC2 para crear y mantener entornos de pruebas a lo largo de la iteración de desarrollo. Esto permitirá ejecutar pruebas exhaustivas y personalizadas de la aplicación.

* **Amazon RDS (Relational Database Service):**

**Descripción:** Amazon RDS es un servicio de bases de datos gestionado que simplifica la creación y gestión de bases de datos relacionales. Permite configurar bases de datos como MySQL, PostgreSQL, SQL Server, entre otros, para las necesidades de pruebas de almacenamiento de datos.

**Capacidades:**

* **Gestión Simplificada de Bases de Datos:** Simplifica la creación y gestión de bases de datos relacionales.
* **Compatibilidad con Varios Motores de Bases de Datos:** Ofrece soporte para diferentes tipos de bases de datos.
* **Pruebas de Integridad, Rendimiento y Escalabilidad:** Permite pruebas exhaustivas en estos aspectos.

**Costos:** Los detalles sobre los precios de Amazon RDS están disponibles en la página oficial de precios de Amazon RDS <https://aws.amazon.com/es/rds/?nc2=type_a>

**Uso:** 50 horas a Amazon RDS para configurar bases de datos de prueba, cargar datos de referencia y realizar pruebas de integridad, rendimiento y escalabilidad de bases de datos.

* **AWS Device Farm:**

**Descripción:** AWS Device Farm es un servicio de pruebas móviles que permite probar aplicaciones en una amplia variedad de dispositivos y configuraciones, incluyendo dispositivos iOS y Android. Facilita la automatización de pruebas y proporciona informes detallados sobre el rendimiento de la aplicación en diferentes dispositivos.

**Capacidades:**

* **Pruebas en Diversos Dispositivos Móviles:** Amplia variedad de dispositivos y sistemas operativos.
* **Automatización de Pruebas:** Permite automatizar pruebas para garantizar la compatibilidad y el rendimiento en distintos entornos móviles.

**Costos:** Los costos de AWS Device Farm también dependen del tiempo de uso, la cantidad de dispositivos y otros factores. Los detalles sobre los precios de AWS Device Farm se encuentran en la página oficial de precios de AWS Device Farm <https://aws.amazon.com/es/device-farm/?nc2=type_a>.

**Uso:** 50 horas a AWS Device Farm para ejecutar pruebas exhaustivas de la aplicación móvil en una amplia gama de dispositivos y sistemas operativos. Esto garantizará que la aplicación sea compatible y funcione de manera óptima en diferentes entornos móviles.

* + 1. **Recursos Económicos para la contratación de servicios/personal:**

No aplicable, ya que contamos con ingenieros y recursos ya mencionados, los cuales fueron asignados previamente.

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Técnicas** | **Niveles** | **Tipos** | **Objetivo Esperado** |
| **Pruebas basadas en requerimientos:** Estas pruebas se diseñan a partir de los requerimientos o especificaciones del sistema. | **Pruebas unitarias:** Se centran en unidades individuales de software como funcionales o métodos. | **Funcionales:** Verifican que el software cumpla con los requisitos y especificaciones definidas. | **Garantizar la calidad funcional de GHOST:** Validar que cada función o característica del sistema actúa conforme a los requisitos.  **Identificar y corregir defectos críticos:** Detectar fallos en las funcionalidades más esenciales del sistema. |
| **Pruebas exploratorias:** Se realizan sin un script predefinido permitiendo al tester descubrir defectos basándose en su conocimiento y experiencia. | **Garantizar la calidad funcional de GHOST:** Asegurarse de que el sistema pueda manejar la carga esperada y responder adecuadamente bajo presión. |
| **Pruebas basadas en navegadores**: Se centran en garantizar la compatibilidad del software con diferentes navegadores web. | **Compatibilidad:** Evalúan si el software funciona correctamente en diferentes sistemas plataformas y navegadores. | **Validar la compatibilidad con navegadores populares**: Confirmar que GHOST funcione correctamente en los principales navegadores y no presente problemas de diseño o funcionalidad. |
| **Pruebas de caja negra:** Se realizan sin considerar la estructura interna del software se basan únicamente en los requerimientos y especificaciones. | **Pruebas de integración:** Se centran en verificar la interacción entre diferentes módulos o componentes del software  **Pruebas de sistema:** Se centra en evaluar el sistema completo en un entorno que simula la producción. | **Funcionales:** Estas pruebas garantizan que los módulos o partes del software trabajan correctamente juntas. | **Garantizar la calidad funcional de GHOST:** Validar que los componentes integrados del sistema interactúen correctamente.  **Identificar y corregir defectos críticos:** Identificar fallos que puedan surgir cuando diferentes partes del sistema interactúan. |
| **Pruebas de usabilidad:** Hoy evalúan la factibilidad de uso del software desde la perspectiva del usuario final. | **Funcionales:** Evalúan el sistema completo para garantizar que cumple con los requerimientos. | **Garantizar la calidad funcional de GHOST:** Asegurarse de que el sistema es intuitivo y amigable para el usuario.  **Identificar y corregir defectos críticos:** Detectar problemas que pueden afectar la experiencia general del usuario. |
| **Garantizar la calidad funcional de GHOST:** Determinar los límites de capacidad del sistema e identificar buscar posibles fallos cuando se alcanzan esos límites. |
| **Pruebas de aceptación:** se centran en validar que el software cumple con las expectativas y necesidades del cliente o usuario final. | **Funcionales:** Permiten a los usuarios finales probar el software en condiciones reales antes de su lanzamiento oficial. | **Garantizar la calidad funcional de GHOST:** Obtener retroalimentación directa de los usuarios finales antes de la liberación final.  **Identificar y corregir defectos críticos:** Reconocer y solucionar problemas desde la perspectiva del usuario. |

* 1. **Distribución de Esfuerzo**

**Pruebas Funcionales (55% - 110 horas):**

Las funcionalidades core son esenciales para la operación diaria y la percepción de calidad de GHOST. Se asigna la mayor parte del esfuerzo para garantizar que estas operaciones sean fiables y libres de defectos. Se utilizarán herramientas de automatización compatibles con AWS para desarrollar y ejecutar pruebas funcionales. Herramientas como Selenium para pruebas web y JUnit para pruebas unitarias se integrarán en un pipeline de CI/CD para la reutilización de pruebas y la reducción del esfuerzo manual en futuras versiones.

**Pruebas de Compatibilidad (25% - 50 horas):**

GHOST debe funcionar sin problemas en una variedad de navegadores y dispositivos. Por lo tanto, se utilizará BrowserStack para pruebas cruzadas y se verificará la compatibilidad en los navegadores web más populares y dispositivos móviles, asegurando accesibilidad y una experiencia de usuario coherente en todas las plataformas.

**Revisiones, Análisis de Resultados, y Preparación de Informes (20% - 40 horas):**

La revisión y análisis de los resultados de las pruebas son fundamentales para entender los defectos y preparar informes detallados para los stakeholders. Se emplearán herramientas como JIRA para la documentación y seguimiento de errores, facilitando la toma de decisiones informadas y la planificación de acciones correctivas y futuras iteraciones de pruebas.