**Estrategia de Pruebas No. 2**

|  |  |
| --- | --- |
| Fecha: | 17 de noviembre de 2024 |
| Autores: | Calletana López  Juan Tapia  Raúl Ramos  Sergio Andrés Gelves |
| Empresa: | The Software Design Company (TSDC) |

**Tabla de Contenido**

[1 Aplicación bajo prueba 2](#_heading=h.gjdgxs)

[1.1 Nombre de la Aplicación: 2](#_heading=h.30j0zll)

[1.2 Versión de la aplicación: 2](#_heading=h.1fob9te)

[1.3 Descripción: 2](#_heading=h.3znysh7)

[1.4 Listado de funcionalidades 2](#_heading=h.2et92p0)

[1.5 Diagrama de la arquitectura. 5](#_heading=h.tyjcwt)

[1.6 Diagrama de Contexto 6](#_heading=h.3dy6vkm)

[1.7 Modelo de datos 6](#_heading=h.1t3h5sf)

[1.8 Modelo de GUI 7](#_heading=h.4d34og8)

[2 CONTEXTO DE LA ESTRATEGIA DE PRUEBAS 8](#_heading=h.2s8eyo1)

[2.1 Objetivos: 8](#_heading=h.17dp8vu)

[2.2 Duración de la iteración de pruebas: 9](#_heading=h.3rdcrjn)

[2.3 Presupuesto de pruebas: 9](#_heading=h.26in1rg)

[2.4 TNT (Técnicas, Niveles y Tipos) de Pruebas: 10](#_heading=h.lnxbz9)

[2.5 Distribución de Esfuerzo: 11](#_heading=h.35nkun2)

[2.6 Justificación del uso de Monkey-Cypress y RIPuppet: 13](#_heading=h.1ksv4uv)

[2.7 Análisis de los Pros y Contras de las Herramientas: 13](#_heading=h.44sinio)

[2.8 Referencias: 14](#_heading=h.2jxsxqh)

# Aplicación bajo prueba

## Nombre de la Aplicación:

Ghost

## Versión de la aplicación:

5.96.0

## Descripción:

Ghost es un sistema de gestión de contenidos (CMS) que se usa principalmente para crear blogs y sitios web enfocados en publicaciones. Es conocido por ser fácil de usar y por permitir que las personas publiquen y gestionen contenido de manera sencilla

## Listado de funcionalidades

Inicio

1. Iniciar sesión: inicia sesión en el módulo administrativo y muestra el dashboard de la aplicación.

Configuración General.

1. Configurar título y descripción: permite cambiar el título y la descripción del sistema

Staff:

1. Agregar miembro: permite invitar otras personas para participar en el sistema
2. Editar miembro: permite editar los datos de un colaborador

Personalizar el diseño:

1. Personalizar marca: Permite cambiar descripción, color de botones y de fondo de la pagina
2. Personalizar ancho del sitio: permite cambiar el tipo de letra y ubicación del logotipo
3. Personalizar Home Page: permite cambiar la estructura de la página principal
4. Personalizar Post: permite activar opciones de personalización en la vista de los posts
5. Personalizar Navegación: Permite configurar los vínculos de navegación del menú principal
6. Modificar perfil: permite cambiar el correo y la contraseña entre otros datos del usuario que inicio sesión.

Post

1. Agregar post: permite crear una nueva publicación en estado borrador.
2. Editar post: permite modificar los datos de un post existente
3. Eliminar post: permite eliminar un post publicado
4. Configurar post: permite modificar la configuración de la publicación del post
5. Publicar post: permite publicar el post en la pagina
6. Despublicar post: permite ocultar un post publicado, el cual no se vera en la pagina

Paginas

1. Nueva página: permite agregar una nueva página en modo borrador.
2. Editar página: permite modificar la información de una página existente
3. Configurar página; permite modificar los datos como la url, fecha de publicación, tags, tipo de acceso, entre otros.
4. Eliminar página: permite eliminar una página existente
5. Publicar página: permite publicar una página en el sistema
6. Despublicar página: permite ocultar una página y volver al estado en borrador

Elementos web

1. Agregar imagen: permite agregar una imagen a una pagina
2. Agregar video: permite agregar un video a una pagina
3. Agregar producto: permite agregar una imagen con un título, botón y estrellas para calificación del producto
4. Agregar galería: permite agregar varias imágenes organizadas, máximo 9

Tags

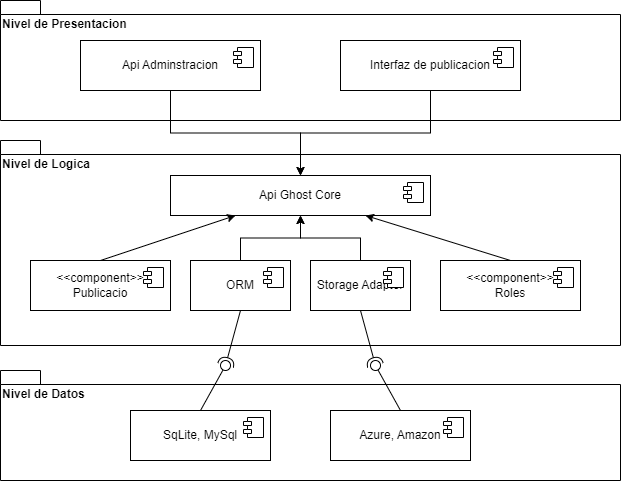
1. Agregar tag: permite agregar tag para agrupar los posts publicados
2. Editar tag: modifica un tag existente

Suscriptores

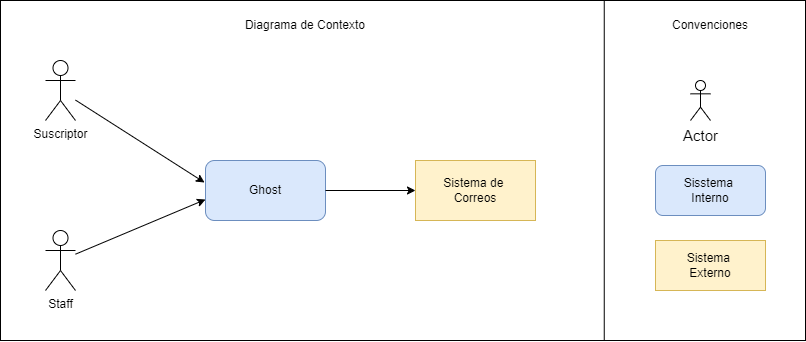
1. Agregar suscriptor: permite agregar un suscriptor por medio del correo, este es una persona que tiene acceso a contenido exclusivo
2. Editar suscriptor: Editar un suscriptor existente.

## Diagrama de la arquitectura.

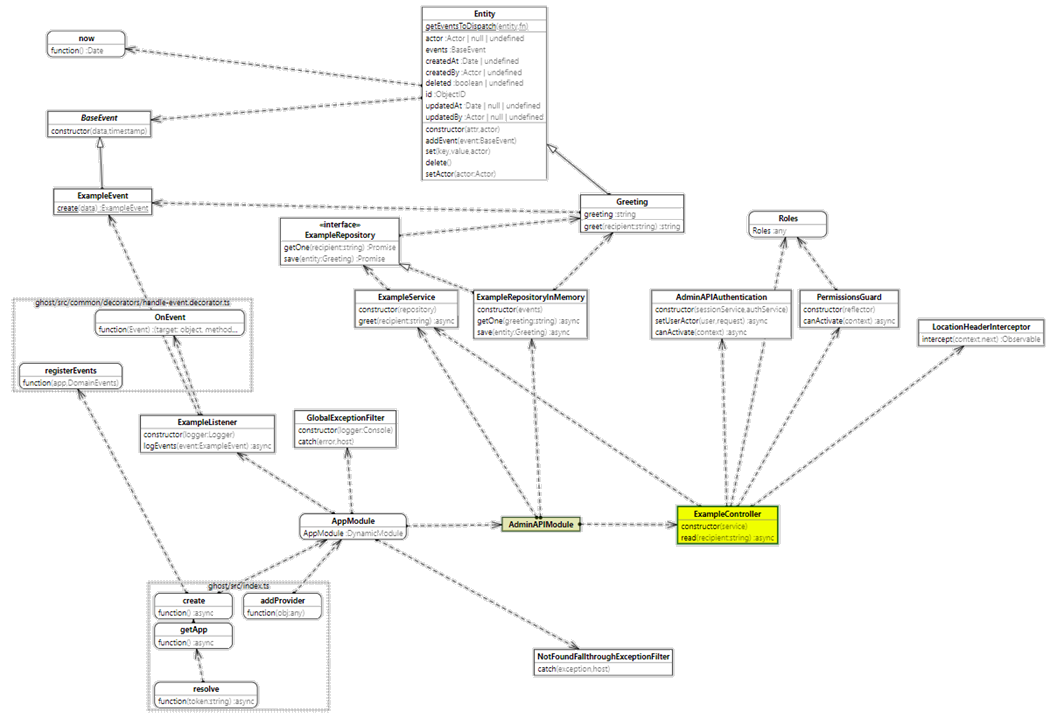
**Diagrama de Componentes**



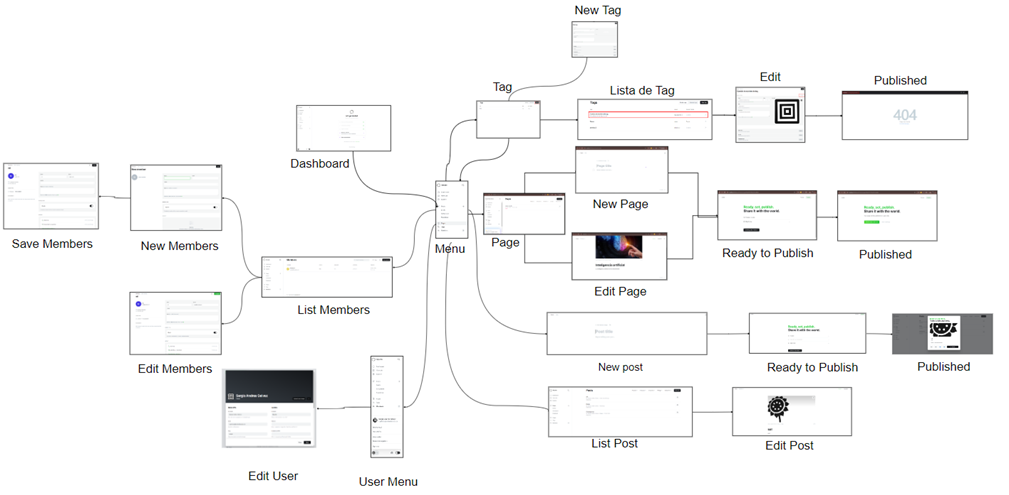
## Diagrama de Contexto



## Modelo de datos



## Modelo de GUI



# CONTEXTO DE LA ESTRATEGIA DE PRUEBAS

## Objetivos:

Se pretende validar aspectos funcionales del sistema GHOST mediante pruebas tanto de caja negra como de caja blanca, priorizando pruebas automatizadas para alcanzar la mayor cobertura posible. Los objetivos principales son:

1. Cubrir adecuadamente áreas de prueba manual donde la interacción del usuario pueda generar errores.
2. Realizar pruebas exploratorias para encontrar casos raros que se salgan del resto de pruebas que se plantean.
3. Verificar funcionalidades críticas del sistema, enfocándose en aquellas de alto impacto para el usuario usando pruebas de extremo a extremo.
4. Realizar pruebas de regresión visual para garantizar la estabilidad de funcionalidades en diferentes navegadores usando ResembleJs.
5. Automatizar pruebas de validación de datos de procesos clave, como la creación y publicación de contenido que permitan probar el buen funcionamiento de los diferentes formularios.

## Duración de la iteración de pruebas:

**Fechas:** Inicia el 25 de noviembre al 17 de enero (8 semanas). Horas totales: 256

**Ingenieros senior (4):** 32 horas semanales.

**Distribución de tiempo:** Las 256 horas se refieren al tiempo de trabajo de los ingenieros más no al tiempo de ejecución en máquina y están distribuidas de la siguiente manera:

Pruebas manuales: 12 horas

Pruebas Exploratorias: 12 horas

Pruebas e2e: 102 horas

Pruebas VRT: 40 horas

Validación de datos: 90 horas

Para mayor detalle de la distribución de las horas en las diferentes semanas ver el numeral **2.5 Distribución de Esfuerzo**

## Presupuesto de pruebas:

### Recursos Humanos:

**Ingeniero automatizador Senior:** 8 horas disponibles para la creación de scripts y configuración de herramientas de reconocimiento.

### Recursos Computacionales:

Amazon AWS: 144 horas en máquinas AWS para la ejecución de pruebas automatizadas y de reconocimiento.

### Recursos Económicos para la contratación de servicios/personal:

500 USD para outsourcing en módulos específicos donde la automatización aún no es viable.

## TNT (Técnicas, Niveles y Tipos) de Pruebas:

Las pruebas se estructuran de acuerdo con la pirámide de automatización y clasifican en pruebas de caja negra y caja blanca. Las pruebas de caja negra validan el sistema desde la perspectiva del usuario, sin necesidad de conocimiento interno, mientras que las de caja blanca verifican el funcionamiento del código. En esta estrategia se priorizan los siguientes niveles y tipos:

|  |  |  |  |
| --- | --- | --- | --- |
| Nivel | Tipo | Técnica | Objetivo |
| Sistema | Funcional | Pruebas Manuales | Objetivo 1 |
| Sistema | Funcional  Caja Negra | GUI Ripping (Ripper) | Objetivo 2 |
| Sistema | Funcional | APIs De Automatización (cypress) | Objetivo 3 |
| Sistema | Funcional | Pruebas Automatizadas de Regresión Visual (ResembleJs) | Objetivo 4 |
| Sistema | Funcional | APIs de Automatización (a priori, pool, ramdon)  (Cypress) | Objetivo 5 |

**Pruebas automatizadas de regresión:** pruebas automatizadas para validar que las funcionalidades críticas no se vean afectadas por cambios recientes en el código.

**Pruebas de rendimiento (No Funcional):** Scripts automatizados para evaluar cómo el sistema se comporta bajo carga moderada.

**Pruebas exploratorias manuales:** pruebas exploratorias para identificar defectos en áreas no cubiertas por la automatización.

**Pruebas de integración:** Se asegura que los diferentes módulos del sistema funcionen correctamente cuando se integran.

**Pruebas de sistema:** Validar el sistema completo desde el punto de vista del usuario final.

**Prueba Funcional:** Verifica que el sistema funcione según lo esperado.

**Prueba de caja blanca:** Evalúa la lógica interna del código, verificando su estructura y funcionamiento sin considerar las entradas y salidas.

**Prueba de caja negra:** Se centra en las entradas y salidas del sistema, validando que funcione según los requisitos sin necesidad de conocer su implementación interna.

## Distribución de Esfuerzo:

Los recursos se distribuyen conforme a la pirámide de automatización, maximizando pruebas de menor nivel (unitarias y de integración), apoyadas con herramientas de reconocimiento (Monkey y RIPuppet) en el nivel intermedio y pruebas manuales en la cúspide para asegurar una cobertura completa y eficiente:





### Pruebas de Reconocimiento (Tercer Nivel):

**Objetivo**: Ejecutar pruebas exploratorias con Monkey-Cypress y RIPuppet para simular interacciones aleatorias y detectar fallos en flujos de usuario típicos.

**Horas**: 5 horas para configuración y ejecución, 139 horas de AWS para simulaciones de carga.

**Pruebas de Caja Negra:** Monkey-Cypress simula acciones aleatorias en áreas críticas y RIPuppet explora rutas de usuario con variación en interacciones.

### Pruebas Manuales Exploratorias (Nivel Superior):

**Objetivo**: Cubrir áreas de configuración general y permisos donde la interacción humana es clave.

**Horas**: 10 horas de outsourcing.

**Pruebas de Caja Negra:** Validar funciones críticas desde el punto de vista del usuario en la configuración general del sistema y gestión de permisos.

## Justificación del uso de Monkey-Cypress y RIPuppet:

Estas herramientas se utilizaran para complementar la cobertura de pruebas de automatización:

* **Monkey-Cypress** permitirá realizar pruebas de tipo monkey testing mediante interacciones aleatorias en áreas no cubiertas por pruebas específicas, logrando descubrir fallos inesperados o potenciales errores en zonas menos previsibles del sistema.
* **RIPuppet** se utiliza para rippers testing, que recorre las interfaces con interacciones aleatorias, pero con cierta lógica para explorar rutas de usuario comunes. Esto es útil para identificar rutas inusuales y errores de navegación en GHOST.

## Análisis de los Pros y Contras de las Herramientas:

|  |  |
| --- | --- |
| Monkey-Cypress: | RIPuppet |
| Pros  No requiere configuración previa de los scripts, permitiendo una configuración y ejecución rápida.  Ofrece una cobertura amplia y aleatoria, lo que es ideal para pruebas exploratorias en zonas inesperadas. | **Pros**  Permite definir algunas rutas o interacciones específicas dentro de la aleatoriedad, lo que da una exploración guiada y permite identificar errores en flujos de usuario más reales.  Útil para evaluar la estabilidad del sistema en secuencias comunes y verificar que el sistema maneje errores de navegación. |
| Contras  Las interacciones pueden ser demasiado aleatorias, lo cual puede reducir la relevancia de ciertos errores detectados.  Requiere análisis adicional para categorizar los errores generados | **Contras**  Puede requerir más tiempo de configuración inicial que Monkey-Cypress.  Las rutas predefinidas pueden limitar el alcance aleatorio de las pruebas si no se ajustan adecuadamente |

Anexos

* cotización outsorcing.pdf
* Link de video presentación: <https://youtu.be/c8PeyOzRXOw>
* Link de prueba monkey: https://youtu.be/cdeP\_LdMUms