**ESTRATEGIA DE PRUEBAS**

**PRUEBAS AUTOMATIZADAS DE SOFTWARE**

**Jose Daniel Garcia Arias - 202423665**

**Alberto Silva Rueda - 202423031**

**Maestría en ingeniería de software**

**Universidad de los Andes**

**2024**

Tabla de contenido

[1. Aplicación Bajo Pruebas 3](#_Toc183880970)

[1.1. Nombre Aplicación: 3](#_Toc183880971)

[1.2. Versión: 3](#_Toc183880972)

[1.3. Descripción: 3](#_Toc183880973)

[1.4. Funcionalidades Core: 3](#_Toc183880974)

[1.5. Diagrama de Arquitectura: 3](#_Toc183880975)

[1.6. Diagrama de Contexto: 4](#_Toc183880976)

[1.7. Modelo de Datos: 4](#_Toc183880977)

[1.8. Modelo de GUI: 5](#_Toc183880978)

[2. Contexto de la estrategia de pruebas 5](#_Toc183880979)

[2.1. Objetivos: 5](#_Toc183880980)

[2.2. Duración de la iteración de pruebas: 5](#_Toc183880981)

[2.3. Presupuesto de pruebas: 5](#_Toc183880982)

[2.3.1. Recursos Humanos 6](#_Toc183880983)

[2.3.2. Recursos Computacionales 6](#_Toc183880984)

[2.3.3. Recursos Económicos para la contratación de servicios/personal: 7](#_Toc183880985)

[2.4. TNT (Técnicas, Niveles y Tipos) de pruebas: 7](#_Toc183880986)

[2.5. Distribución de Esfuerzo 8](#_Toc183880987)

# Aplicación Bajo Pruebas

## Nombre Aplicación:

Ghost

## Versión:

La versión de Ghost bajo pruebas es 5.96

## Descripción:

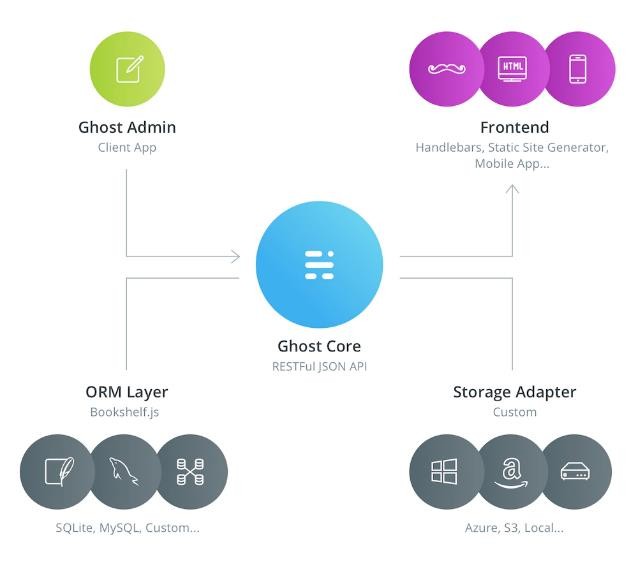
Ghost es una poderosa aplicación para que los editores profesionales creen, compartan y hagan crecer un negocio en torno a su contenido. Viene con herramientas modernas para crear un sitio web, publicar contenido, enviar boletines informativos y ofrecer suscripciones pagas a los miembros.

## Funcionalidades Core:

* Gestión de suscripciones y membresías: Permite crear contenido gratuito y de pago.
* Envío de boletines por correo: Integrado directamente desde la plataforma.
* Análisis de audiencia: Para rastrear el rendimiento de publicaciones y suscriptores.
* Creación y edición de Posts: permite la creación de post con diseños actuales, permitiendo la carga de contenido en diferentes formatos y guardarlo como borrador para su posterior publicación.
* Creación y edición de Pages: permite la creación de pages con diseños actuales, permitiendo la carga de contenido en diferentes formatos y guardarlo como borrador para su posterior publicación.
* Publicar y despublicar los posts construidos en la página principal consultada por la audiencia.
* Crear tags: Permite la creación de tags para asociar varios posts por un tag especifico y permitirle a la audiencia consultar información de su interés rápidamente.
* Programación de post: Permite la configuración para la publicación de un post en una hora específica.
* Soporte para múltiples usuarios y roles: Ideal para equipos colaborativos.
* Login: permite la autenticación de los usuarios y sus roles.
* Inyección de código: Permite la inyección de código personalizado en las publicaciones realizadas

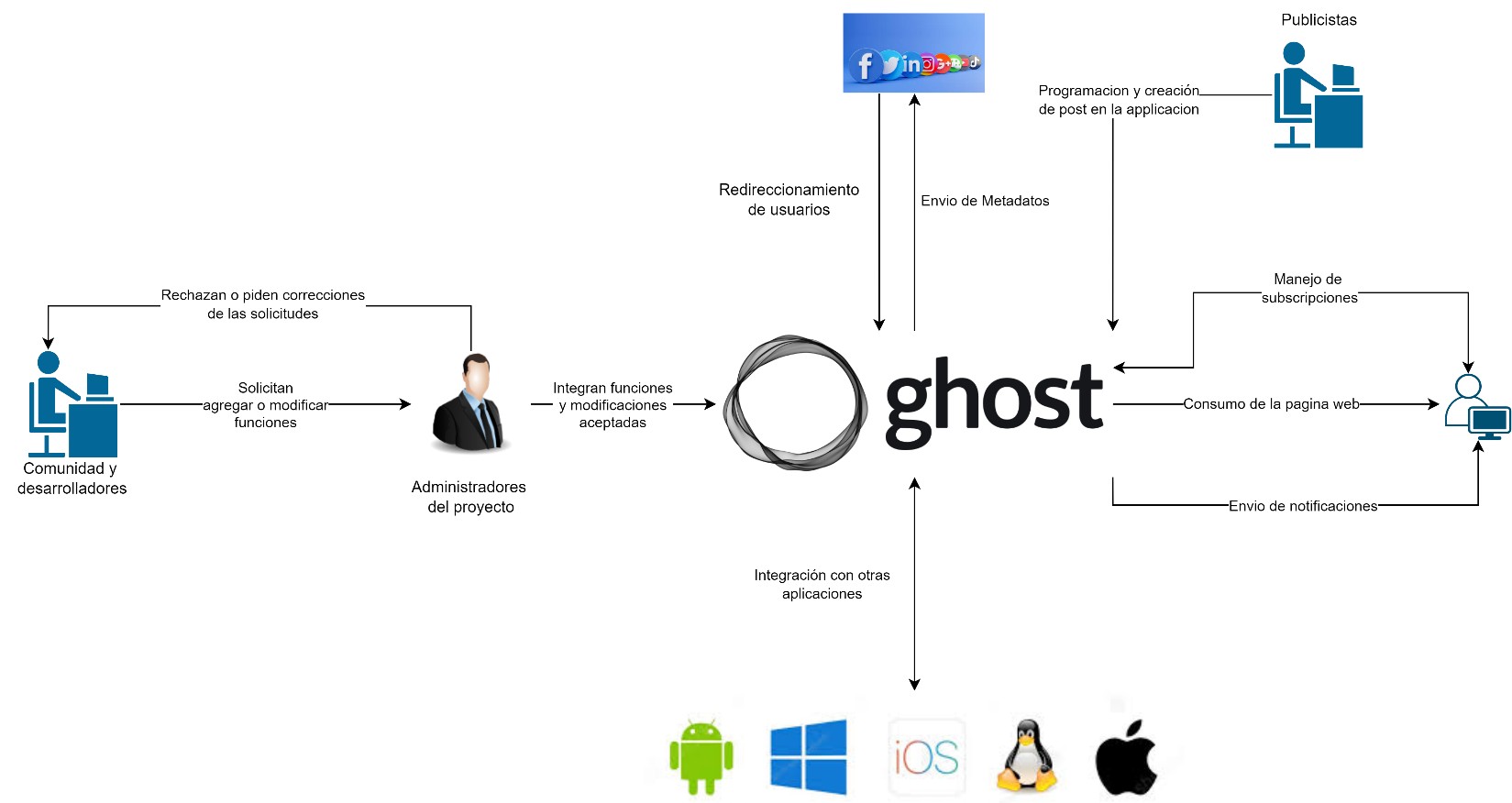
## Diagrama de Arquitectura:

De la página oficial de [Ghost](https://ghost.org/) vemos que la arquitectura viene definida por:



*Ilustración 1:* Arquitectura consultada el 27/10/2024 en https://ghost.org/docs/architecture/

## Diagrama de Contexto:



*Ilustración 2:* Diagrama de contexto.

## Modelo de Datos:

En el modelo de datos se puede consultar en el siguiente en los siguientes links según el formado conveniente

* [GHOST\_DIAGRAMA\_ER.jpg](https://uniandes-my.sharepoint.com/:i:/g/personal/jd_garciaa1_uniandes_edu_co/EUk9p3va2UxNt_6UZjVcTwQBx62B5sNDwthrbDk5ONOlrw?e=uzUOHt)
* [GHOST\_DIAGRAMA\_ER.pdf](https://uniandes-my.sharepoint.com/:b:/g/personal/jd_garciaa1_uniandes_edu_co/EUzA3JIhrfxOgelYHP6Czl4BKIcKuOFOGnjPwsM7c0rejw?e=KD5sPt)

## Modelo de GUI:

El diagrama GUI se puede consultar en el siguiente enlace:

* [DiagramaGUI.jpg](https://uniandes-my.sharepoint.com/:i:/g/personal/jd_garciaa1_uniandes_edu_co/EamWrmNCbcVKvDmlAwhy5JcBIOqCJEJIuComtu2LIzzd6w?e=1J2fPA)

Enlace al tablero MIRO:

* [DiagramaGUI en MIRO](https://miro.com/app/board/uXjVLTPR02c=/?share_link_id=340910632)

# Contexto de la estrategia de pruebas

Con el objetivo general de asegurar la calidad funcional de la aplicación Ghost en su versión 5.9.6 se plantean los siguientes objetivos:

## Objetivos:

Para se tienen los siguientes objetivos siguiendo la metodología SMART:

1. Identificar al menos 10 defectos en la aplicación a través de la exploración libre.
2. Verificar que el sistema detecte y maneje correctamente al menos el 99% de los eventos atípicos producidos por los *monkey testing’s*.
3. Asegurar que todos los flujos de negocio críticos se ejecuten correctamente del principio al fin en un 100% de los casos, según los scripts de prueba automatizados que se ejecutarán diariamente.
4. Garantizar que no haya más de 5 diferencias visuales significativas entre la versión actual y la anterior de la aplicación, según los resultados de las pruebas automatizadas de regresión visual que se ejecutarán después de cada despliegue.
5. Verificar que el sistema rechace el 100% de los datos de entrada inválidos definidos en las reglas de negocio y procese correctamente los datos válidos.
6. Garantizar que el sistema el sistema cumpla con el 100% de los criterios de usabilidad exigidos por The Software Design Lab, según los resultados de las pruebas de usabilidad que se realizarán.

## Duración de la iteración de pruebas:

Por directiva de laboratorio de investigación The Software Design Lab y de la Universidad de los Andes, el tiempo abarcado por esta estrategia de pruebas es:

* **Fecha inicio:** Noviembre 25 del 2024
* **Fecha Fin:** Enero 23 del 2025 (8 semanas descontando los tres días festivos)

## Presupuesto de pruebas:

Para este plan de pruebas, The Software Design Lab y la Universidad de los Andes han planteado el siguiente presupuesto de recursos humanos, computacionales y económicos:

* **Recursos humanos:** 2 ingenieros de automatización senior (8 horas/persona por semana).

### Recursos Humanos

Se cuenta con dos ingenieros de automatización senior contratado con una carga laboral descrita en la siguiente tabla:

Tabla 1: Tabla de recursos humanos.

|  |  |  |
| --- | --- | --- |
| **Horas laborales del recurso humano** | |  |
| **Horas/ persona por semana** |  | 8 |
| **Horas/ persona por mes** |  | 32 |
| **Horas/ persona**  **totales del proyecto** |  | 64 |

### Recursos Computacionales

Para los ingenieros de automatización se tienen asignado el siguiente hardware:

Tabla 2: Recurso de cómputo disponible para el ingeniero de automatización 1.

|  |  |
| --- | --- |
| **Característica** | **Descripción** |
| **Procesador** | Intel(R) Core(TM) i7-8750H CPU @ 2.20GHz 2.20 GHz |
| **Memoria RAM** | 16 GB DDR4 |
| **Almacenamiento** | SSD de 512 GB |
| **Tarjeta gráfica** | Integrada |
| **Pantalla** | Monitor de 1920x1080 píxeles |
| **Sistema operativo** | Windows 10 o 11, o una distribución de Linux |
| **Otros** | Teclado ergonómico, ratón de precisión, conexión a internet estable |

Tabla 3: Recurso de cómputo disponible para el ingeniero de automatización 2.

|  |  |
| --- | --- |
| **Característica** | **Descripción** |
| **Procesador** | Intel(R) Core(TM) Ultra 7 165H 3.80 GHz |
| **Memoria RAM** | 32 GB DDR4 |
| **Almacenamiento** | SSD de 512 GB |
| **Tarjeta gráfica** | Integrada |
| **Pantalla** | Monitor de 1920x1080 píxeles |
| **Sistema operativo** | Windows 10 o 11, o una distribución de Linux |
| **Otros** | Teclado ergonómico, ratón de precisión, conexión a internet estable |

Dado el presupuesto se cuentan además con las siguientes herramientas de software libre permitidas:

*Tabla 4*: Recursos de software open source disponibles.

|  |  |  |
| --- | --- | --- |
| **Nombre** | **Tipo Licencia** | **Tipo** |
| **Cypress** | Open Source | E2E |
| **Playwright** | Open Source | E2E |
| **Puppeteer** | Open Source | Reconocimiento |
| **Kraken** | Open Source | E2E |
| **Selenium** | Open Source | Record & Replay |
| **Resemble JS** | Open Source | VRT |
| **Backstop JS** | Open Source | VRT |
| **PixelMatch** | Open Source | VRT |

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

Para este presupuesto, no se cuenta con recursos económicos disponibles, solo se cuenta con los recursos humanos ya mencionados anteriormente y el hardware descrito.

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

Para asegurar la calidad de la aplicación bajo pruebas, se proceden a especificar las técnicas, niveles y tipos de pruebas a utilizar teniendo en cuenta lo anteriormente establecido. Para ello se presenta una relación entre el TNT aquí presentado y los objetivos planteados.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Técnica | Nivel | Tipo de Prueba | Objetivo específico de la prueba | Objetivo general relacionado |
| Pruebas exploratorias | Sistema | Funcionales | Descubrir defectos inesperados mediante la exploración libre de la aplicación. | Objetivo 1 |
| Pruebas de reconocimiento | Sistema | Funcionales | Determinar la robustes del sistema ante eventos poco comunes | Objetivo 2 |
| Pruebas de extremo a extremo | Sistema | Funcionales | Verificar funcionamiento de flujos de inicio a fin de la aplicación | Objetivo 3 |
| Pruebas de regresión visual | Sistema | No Funcionales | Verificar la consistencia del software entre versiones diferentes de la aplicación | Objetivo 4 |
| Pruebas de validación de datos | Aceptación y sistema | Funcionales | Verificar robustes del sistema ante datos extraños, válidos y no válidos | Objetivo 5 |
| Pruebas de usabilidad | Aceptación | No Funcional | Determinar la facilidad de uso de la aplicación para los usuarios finales. | Objetivo 6 |

## Distribución de Esfuerzo

Se plantea que los dos desarrolladores senior sigan la siguiente distribución de esfuerzos semana por semana de los dos desarrolladores seniors destinados a esta labor:

**Semana 1:**

* Configuración inicial del entorno de pruebas.
* Exploración de la interfaz para identificar casos de prueba relevantes.
* Diseño de los primeros casos de prueba E2E para funcionalidades core.

**Semana 2:**

* Automatización de los casos de prueba E2E iniciales.
* Ejecución de pruebas exploratorias para descubrir defectos mediante monkeys.
* Revisión y refinamiento de los casos de prueba diseñados.

**Semana 3:**

* Diseño e implementación de pruebas de regresión visual (VRT) utilizando herramientas como Resemble JS, PixelMatch, entre otras.
* Validación de datos en escenarios críticos relacionados con suscripciones y publicaciones.

**Semana 4:**

* Ejecución de pruebas de regresión visual en distintos navegadores y resoluciones.
* Documentación de incidencias encontradas y ajustes necesarios en los casos de prueba.

**Semana 5:**

* Ejecución de pruebas E2E en escenarios más complejos, como programación de publicaciones y gestión de membresías.
* Modificación de los scripts E2E para el uso de pools de datos que permitan evaluar el sistema ante datos inválidos, válidos y poco comunes.
* Análisis preliminar de resultados y optimización de los scripts automatizados.

**Semana 6:**

* Consolidación de resultados obtenidos en pruebas E2E y regresión visual.
* Revisión cruzada de incidencias con el equipo de desarrollo.
* Inicio de reclutamiento de usuarios finales para participar en las pruebas de usabilidad.
* Preparación del reporte final de calidad.

**Semana 7:**

* Evaluación de la aplicación por parte de los usuarios reclutados, para evaluar la usabilidad de la aplicación.
* Análisis de los resultados de las pruebas de usabilidad realizadas.
* Validación de la documentación en el repositorio e integración de los resultados de usabilidad al reporte final de calidad.

**Semana 8:**

* Entrega del reporte final de calidad, listado de incidencias y listado de posibles mejoras.
* Generación del video explicativo para el CTO, cubriendo estrategia, resultados, pros/contras y lecciones aprendidas.
* Última revisión de los entregables antes de la fecha de cierre.