

# Estrategia de Pruebas

## 1. Aplicación Bajo Pruebas

### 1.1. Nombre Aplicación: Ghost

### 1.2. Versión: 3.41.1

### 1.3. Descripción:

Ghost es una aplicación para gestionar sitios web, en la cual los usuarios pueden administrar, diseñar y manipular las páginas y publicaciones de sus sitios. También permite la gestión, notificación y visualización de los suscriptores asociados a un sitio publicado.

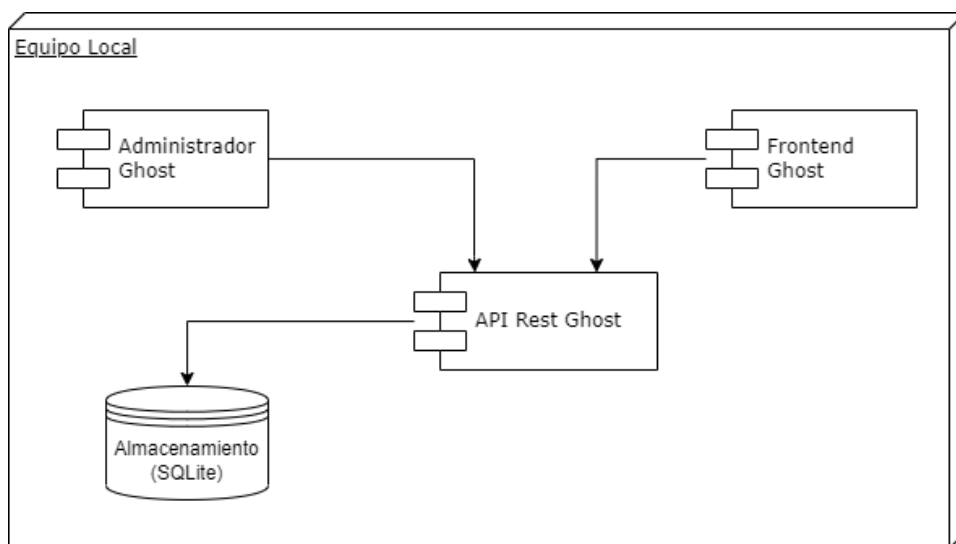
La aplicación Ghost es web y se puede acceder desde un dispositivo móvil o computador con un navegador instalado.

### 1.4. Funcionalidades Core:

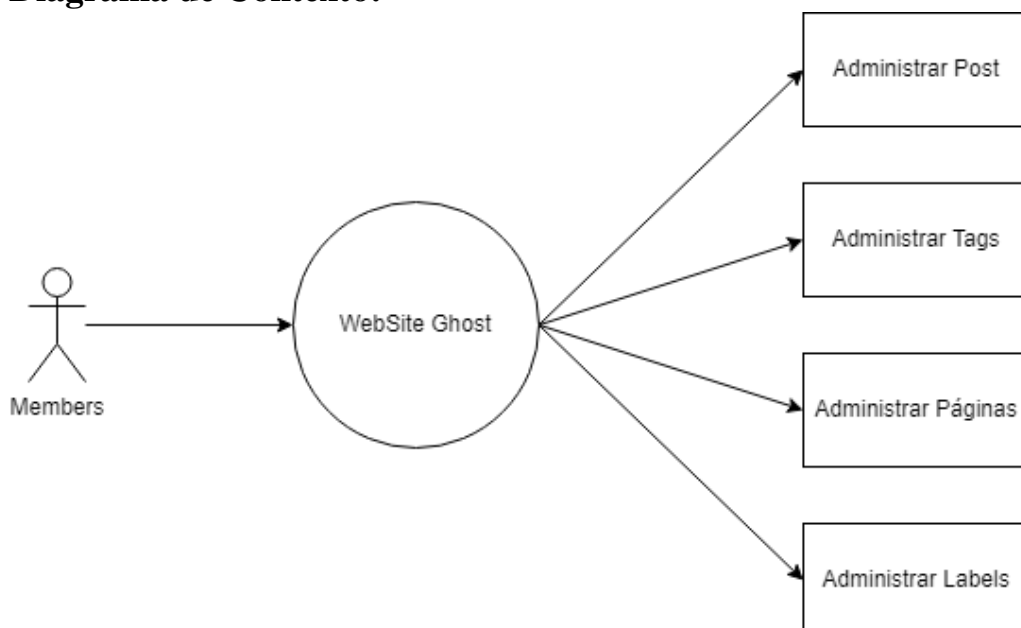
- **Inicio de Sesión:** Permite el acceso a la administración de Ghost para los usuarios creados previamente (Staff) con el correo y contraseña asignado.
- **Creación de Publicación:** Permite la creación de una nueva publicación (Post) para su visualización en la página inicial. Adicionalmente permite la modificación de la URL de publicación, la adición de tags para facilitar el agrupamiento y la fácil identificación, y modificar la visibilidad para restringir su visualización a usuarios.
- **Listar las etiquetas:** La aplicación Ghost permite a los usuarios crear etiquetas que sirven de marca para clasificar páginas y publicaciones del sitio web, de esta forma poder posteriormente realizar búsquedas fáciles y rápidas bajo el criterio de etiqueta. El objetivo de esta funcionalidad es poder listar las etiquetas creadas en el sitio web.
- **Creación de Página:** Permite la creación de una nueva página (Page) para su inclusión en el sitio. Adicionalmente permite modificar la visibilidad de la página, la inclusión de meta data para facilitar la búsqueda en internet y la configuración de código a inyectar para agregar seguimiento a usuarios y accesos.
- **Creación de Etiqueta:** Permite la creación de una nueva etiqueta (Tag) para facilitar el agrupamiento de publicaciones bajo esta etiqueta, además de permitir la personalización de la misma (colores e imágenes) y la adición de meta data para facilitar la búsqueda, y la integración con redes sociales.

- **Creación de Miembro:** Permite la creación de un nuevo miembro (Member) en la plataforma para el envío de boletines y el acceso a las publicaciones, adicionalmente permite la configuración de etiquetas (Labels) para facilitar la búsqueda y agrupamiento de los miembros.
- **Listar las páginas:** El objetivo de esta funcionalidad es listar las páginas de todos los tipos (publicadas, borradores y agendadas para publicación) que se hayan creado en el sitio web.
- **Editar Perfil:** Permite la modificación de la información de los usuarios internos (Staff), así como permitir el cambio de contraseña a otros usuarios desde perfiles administrativos y de la cuenta propia para todos los perfiles. También permite el cambio de la imagen de perfil y el registro de redes sociales para facilitar el contacto con los usuarios.

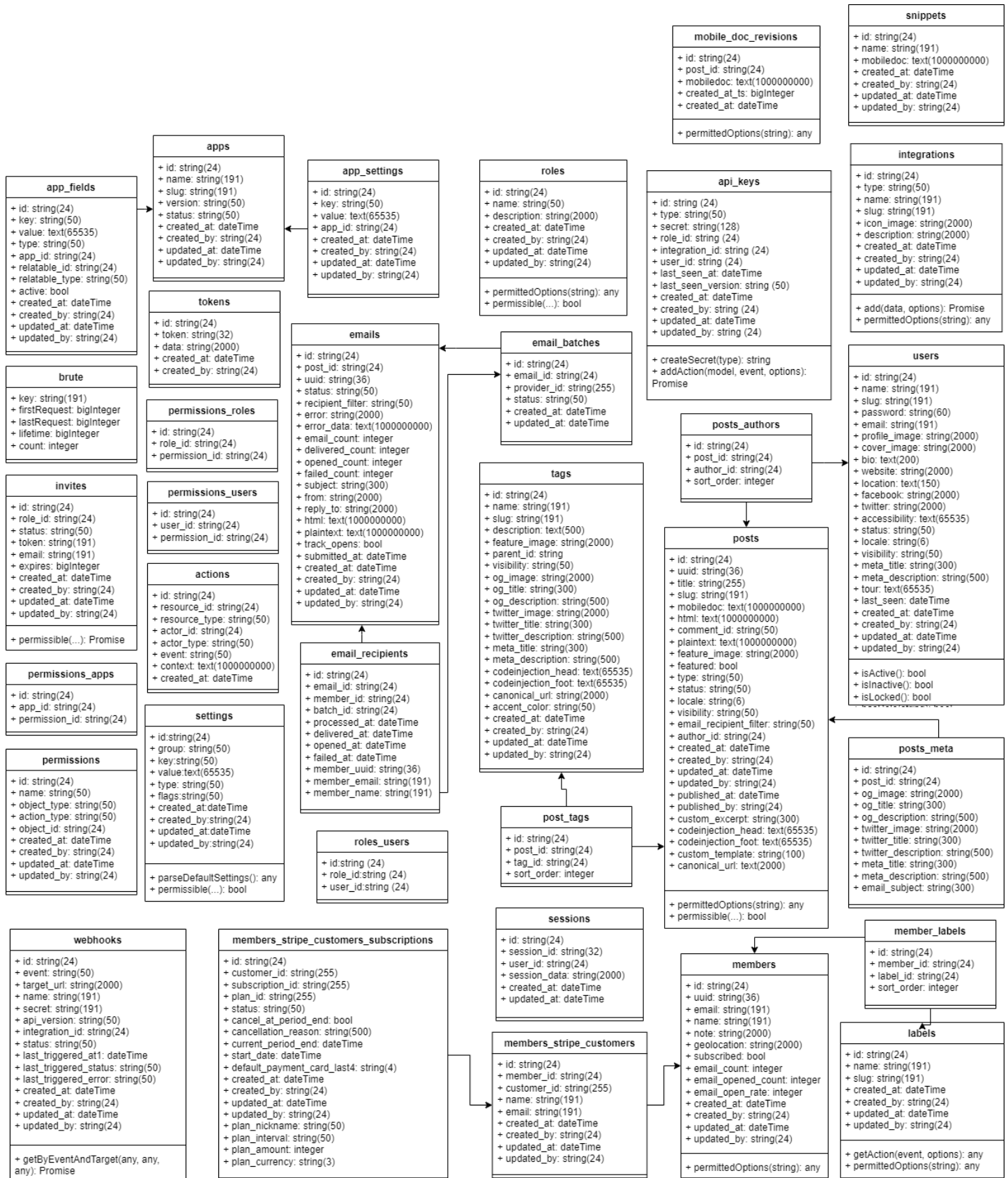
## 1.5. Diagrama de Arquitectura:



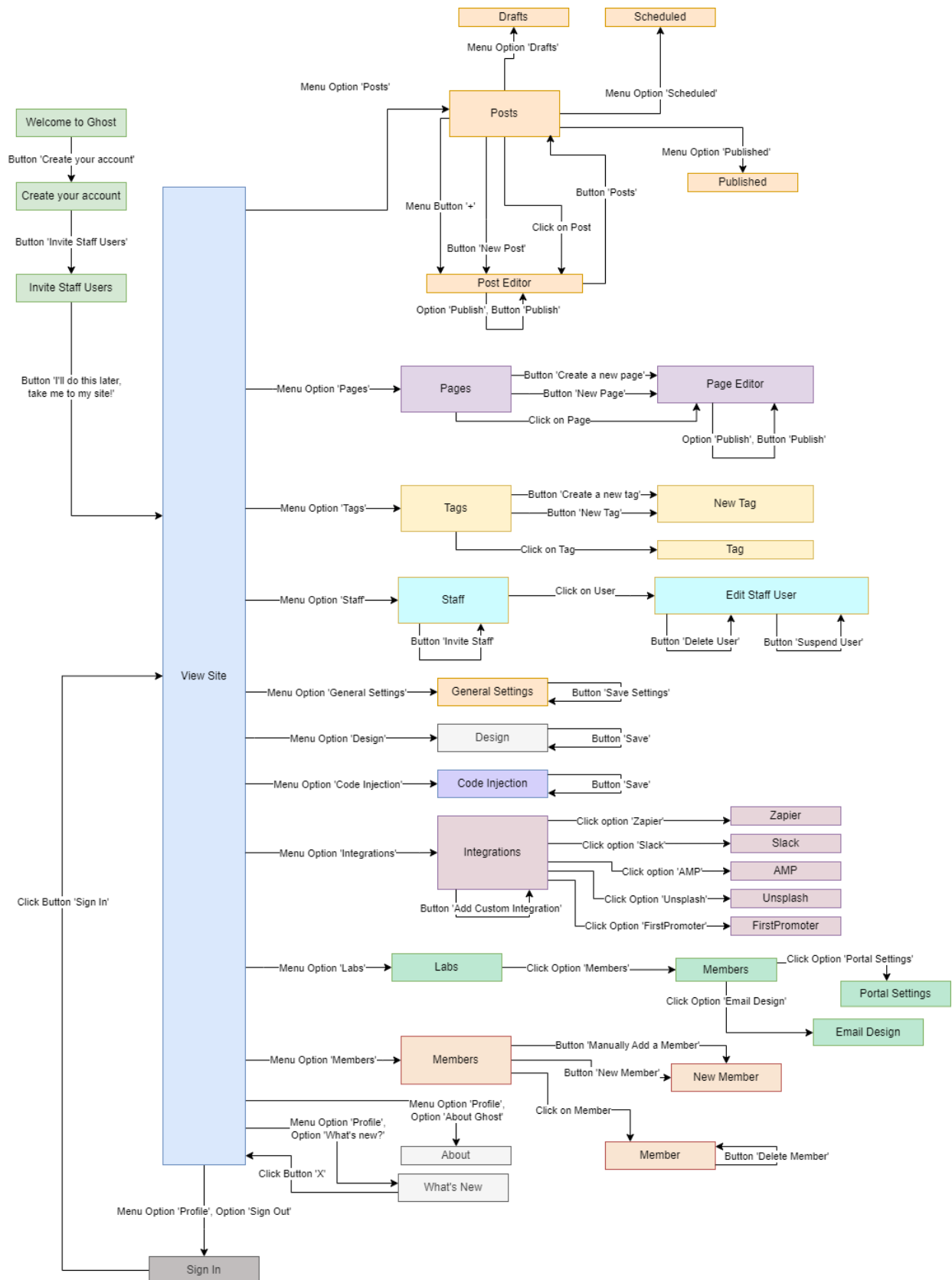
## 1.6. Diagrama de Contexto:



## 1.7. Modelo de Datos:



## 1.8. Modelo de GUI:



## 2. Contexto de la estrategia de pruebas

### 2.1. Objetivos:

- Adquirir un conocimiento básico sobre las funcionalidades de la aplicación por medio de las pruebas manuales; de esta manera, se podrá generar un listado de funcionalidades mínimas a probar que cumplan con los requerimientos funcionales.
- Generar un inventario de pruebas automatizadas mínimas que se deben ejecutar para garantizar un mínimo funcionamiento del sistema.
- Generar el diagrama GUI complete del sistema, navegando sistemáticamente la aplicación.
- Garantizar el correcto funcionamiento de la aplicación con escenarios no convencionales; de esta manera se verificará que no se generen errores con datos no esperados y el manejo correcto de los datos solicitados en los formularios para su funcionamiento.
- Garantizar que la aplicación funcione correctamente en los diferentes navegadores en los que se podría usar por los diferentes usuarios.

### 2.2. Duración de la iteración de pruebas:

Fechas: 22 mayo – 16 de julio

- 1 sprint 22 mayo – 28 mayo - 8 horas /persona – 24 horas máquina
- 2 sprint 29 mayo - 4 junio - 8 horas /persona – 24 horas máquina
- 3 sprint 5 junio – 11 junio - 8 horas /persona – 24 horas máquina
- 4 sprint 12 junio – 18 junio - 8 horas /persona – 24 horas máquina
- 5 sprint 19 junio – 25 junio - 8 horas /persona – 24 horas máquina
- 6 sprint 26 junio – 2 julio - 8 horas /persona – 24 horas máquina
- 7 sprint 3 julio – 9 julio - 8 horas /persona – 24 horas máquina
- 8 sprint 10 julio – 16 julio - 8 horas /persona – 24 horas máquina

### 2.3. Presupuesto de pruebas:

#### 2.3.1. Recursos Humanos

Contamos con 4 testers senior que reúnen experiencia en Cypress, Kraken, pruebas de extremo a extremo, Resemble para pruebas de regresión visual, Monkey (pruebas aleatorias), Rippers (pruebas de exploración sistemática), pruebas manuales y conocimiento en la aplicación Ghost, entre otras herramientas; cada ingeniero cuenta con experiencia entre 1 y 6 meses trabajando en el ámbito de pruebas tanto como manuales como automatizadas.

#### 2.3.2. Recursos Computacionales

Para optimizar el tiempo de ejecución de pruebas, se contará con 1 instancia de EC2 de Amazon (**r6g.2xlarge**), con las siguientes características:

- 8 virtual CPUs
- 64 GiB de RAM
- 50000 megabit de Red

Estas instancias estarán disponibles por 24 horas semanalmente para la ejecución de pruebas.

Adicionalmente se contará con un computador de escritorio por cada ingeniero automatizador con disco de estado sólido de por lo menos 500 GB, con 16 GB de RAM y un procesador Intel Core i7 de octava generación.

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

Cada ingeniero de la propuesta dispondrá de 64 horas, cada hora cuesta \$82.762; por lo tanto, el costo de los 4 ingenieros sería un total de \$21.187.072.

La información de precios brindada anteriormente es la unión de varios perfiles de una empresa Outsourcing que presta recursos especializados en el área; por temas de confidencialidad no es posible referenciar el nombre de esta.

Respecto a los recursos computacionales, la instancia de EC2 tiene un precio de USD\$38.71 mensual para un total de USD\$77.42.

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

Técnica	Nivel	Tipo	Objetivo
Pruebas manuales	Pruebas de sistema	<ul style="list-style-type: none"> <li>• Pruebas de caja negra</li> <li>• Funcionales</li> <li>• Positivas</li> <li>• Negativas</li> </ul>	<ul style="list-style-type: none"> <li>• Adquirir un conocimiento básico sobre las funcionalidades de la aplicación.</li> <li>• Generar un listado de funcionalidades mínimas a probar.</li> <li>• Garantizar que las pruebas que requieren de ciertas precondiciones especiales salgan correctamente.</li> </ul>
Pruebas de extremo a extremo	Pruebas de sistema	<ul style="list-style-type: none"> <li>• Pruebas de caja negra</li> <li>• Funcionales</li> <li>• Positivas</li> </ul>	<ul style="list-style-type: none"> <li>• Generar un inventario de pruebas automatizadas mínimas que se deben ejecutar para garantizar un mínimo funcionamiento del sistema.</li> </ul>
Pruebas de exploración sistemática	Pruebas de sistema	<ul style="list-style-type: none"> <li>• Requerimientos funcionales</li> <li>• Pruebas de caja negra</li> <li>• Positivas</li> </ul>	<ul style="list-style-type: none"> <li>• Generar el diagrama GUI complete del sistema, navegando sistemáticamente la aplicación.</li> </ul>
Pruebas aleatorias	Pruebas de sistema	<ul style="list-style-type: none"> <li>• Funcionales</li> <li>• Pruebas de caja negra</li> <li>• Negativas</li> </ul>	<ul style="list-style-type: none"> <li>• Garantizar el correcto funcionamiento de la aplicación con escenarios no convencionales.</li> </ul>
Pruebas de robustez para validación de datos	Pruebas del sistema	<ul style="list-style-type: none"> <li>• Funcionales</li> <li>• Pruebas de caja negra</li> <li>• Negativas</li> </ul>	<ul style="list-style-type: none"> <li>• Validar que la aplicación tenga un manejo correcto de los datos solicitados en los formularios para su funcionamiento.</li> </ul>
Pruebas con escenarios de frontera para validación de datos	Pruebas del sistema	<ul style="list-style-type: none"> <li>• Funcionales</li> <li>• Pruebas de caja negra</li> <li>• Negativas</li> </ul>	<ul style="list-style-type: none"> <li>• Validar que el manejo de límites en los formularios esté correcto de acuerdo a lo especificado con el usuario.</li> </ul>

Pruebas de Record and Replay	Pruebas del sistema	<ul style="list-style-type: none"> <li>Requerimientos funcionales</li> <li>Pruebas de caja negra</li> <li>Positivas</li> </ul>	<ul style="list-style-type: none"> <li>Automatizar los escenarios en las pruebas manuales, aumentando las pruebas automatizadas para garantizar el correcto funcionamiento de la aplicación.</li> </ul>
Pruebas de regresión visual	Pruebas del sistema	<ul style="list-style-type: none"> <li>Requerimientos funcionales</li> <li>Pruebas de caja negra</li> <li>Positivas</li> </ul>	<ul style="list-style-type: none"> <li>Garantizar que la aplicación funcione correctamente en los diferentes navegadores en los que se podría usar.</li> </ul>

## 2.5. Distribución de Esfuerzo

La distribución de esfuerzo que planeamos va a ser dividida en sprints de una semana, donde el primer sprint contará con distribución de 8H semanales para los ingenieros automatizadores y 24H para las maquinas, con el fin de realizar cobertura de código, pruebas funcionales y no funcionales, enfocándonos en realizar pruebas de sistema, por lo tanto, es obligatorio que todos los componentes de este estén disponibles y funcionales. A continuación, se observa un diagrama de Gantt de las actividades realizadas por cada uno de los recursos asignados del presupuesto:

