

# Estrategia de Pruebas

## 1. Aplicación Bajo Pruebas

**1.1. Nombre Aplicación:** Ghost

**1.2. Versión:** 3.41.1

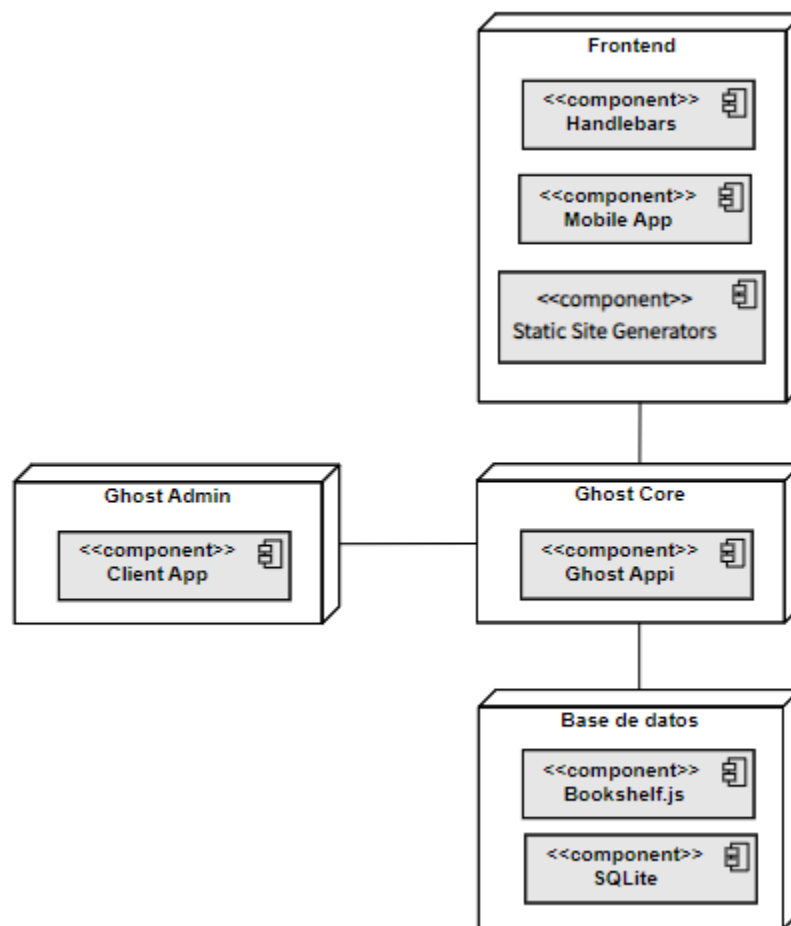
**1.3. Descripción:** Ghost es un CMS que permite de manera fácil crear páginas, publicarlas, configurar la navegación, administrar miembros y otros recursos.

### 1.4. Funcionalidades Core:

- **Configuración / General:** La funcionalidad se encuentra en la opción de configuración, permite la administración de la información del sitio como título, descripción, zona horaria, lenguaje de publicación, icono de publicación, logo de la publicación, imagen de portada, metadatos, estructura de los datos del sitio con sus links para Twitter y Facebook, también puede activar el modo privado para que únicamente se pueda ver el sitio mediante una palabra clave.
- **Configuración / Diseño:** La funcionalidad se encuentra en la opción de configuración, permite la administración de la navegación del sitio, adquirir un tema comprando o disponible de forma gratuita.
- **Posts:** La funcionalidad se encuentra en la sección de administración, allí se pueden crear, modificar, publicar, quitar la publicación, dejar como borrador y borrar un post, además es posible asignar un autor, un tag, programar la hora de publicación, previsualizar y marcar como importante.
- **Páginas:** La funcionalidad se encuentra en la sección de administración, allí se pueden crear, modificar, publicar, quitar la publicación, dejar como borrador y borrar una página, además es posible asignar un autor, un tag, programar la hora de publicación, previsualizar y marcar como importante. Las páginas se diferencian de los posts porque se utilizan para publicar contenido estático, la única forma de mostrarlos en el sitio es mediante enlaces manuales, no permiten feeds para seguimiento de la publicación.
- **Adicionales:** La funcionalidad se encuentra en la parte inferior del menú al seleccionar el nombre del usuario, las opciones que se permiten desde allí primero se encuentra información sobre Ghost como la versión instalada, base de datos, el equipo desarrollador, acceso a la documentación, un foro de ayuda y como contribuir para mejorar Ghost, la segunda opción permite ver información sobre lo nuevo que están haciendo en Ghost, la tercera opción es el acceso directo a la administración del perfil, la cuarta opción tiene un a Twitter para crear una publicación, la quinta opción tiene un acceso directo a una página oficial de Ghost donde indica como se puede usar, por último aparece la opción para salir de la administración del sitio.
- **Miembros (staff):** La funcionalidad se encuentra en la sección de administración, allí se pueden invitar personas, modificar y eliminar los usuarios activos, actualizar contraseña, revisar las invitaciones, revocar invitaciones o enviar nuevamente. Los invitados pueden tener el rol de administrador, editor, autor y contribuidor, la invitación se envía al correo electrónico.

- **Tags:** La funcionalidad se encuentra en la sección de administración, allí se pueden listar, crear, modificar, borrar tags y ver cuántos posts se encuentran asociados al tag. La información del Tag es nombre, descripción, slug e información de meta data.
- **Previsualizar sitio:** esta opción permite ver de manera embebida y en tiempo real la versión actual del sitio web, también es posible lanzar el sitio en una nueva pestaña del navegador.
- **Registro de Usuario:** Esta es la pantalla inicial que la aplicación muestra cuando aún no existe algún usuario y el sitio no se ha creado, a través de un formulario se podrá dar de alta a un nuevo usuario que será el dueño del sitio, los valores que se pueden agregar en esta pantalla son: imagen del avatar del dueño del sitio, nombre completo del dueño del sitio, nombre del sitio web y contraseña del dueño del sitio web.
- **Code injection:** Esta característica permite inyectar código en el header o footer del documento HTML del sitio que será ejecutado cuando la aplicación sea recargada en el navegador, el código valido es de tipo HTML y puede contener estilos en CSS y scripts en lenguaje javascript. Cuando el código inyectado no es válido o contiene errores, este será ignorado.

## 1.5. Diagrama de Arquitectura:



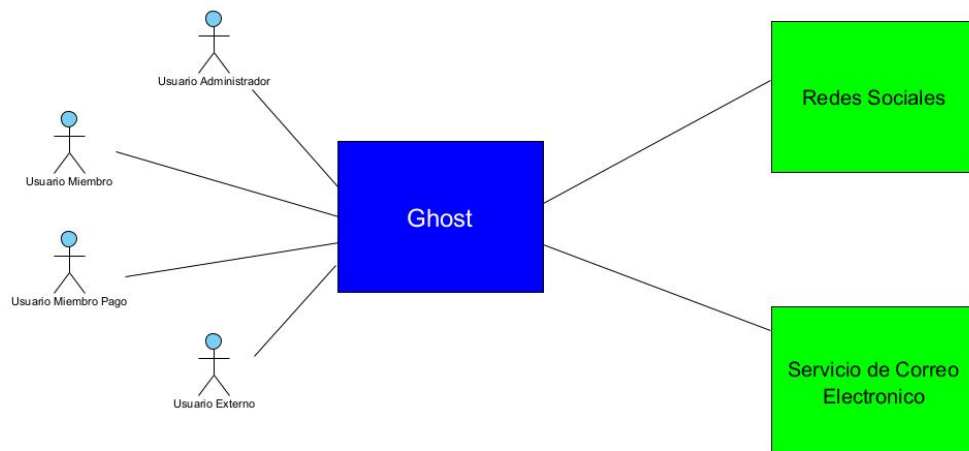
- ✓ Lenguajes usados en la aplicación: Javascript, Nodejs (express), CSS
- ✓ Estructura de directorios

El CMS mantiene en un solo repositorio de Código un monolito que contiene los siguientes

recursos:

1. Content: contenido de la página creada por el usuario
  2. Current: versión actual del CMS que está siendo usada.
  3. versión: contiene todas las versiones del CMS disponibles en el repositorio actual.
- ✓ El código que se puede encontrar en Current/Core:
    - Frontend
    - Server API
    - Build: scripts de compilación
    - CLI
    - Shared
  - ✓ Patrones arquitectónicos y de diseño
    - \* Monolito
    - \* Structural Patterns: Adapter

## 1.6. Diagrama de Contexto:



## 1.7. Modelo de Datos:

<https://github.com/hmauryuniandes/proyecto-pruebas-automatizadas/wiki/Modelo-Entidad-Relacion>

## 1.8. Modelo de GUI:

<https://coggle.it/diagram/ZCW5uUGwoIWe9DaX/t/ghost-login/329d927d4df78dc97b3fda8cf822162f339ae318b83386460b710f6bd3dd2a4e>

## 2.Contexto de la estrategia de pruebas

### 2.1. Objetivos:

- 1) Explorar las funcionalidades de aplicación de forma general e identificar posibles defectos, validando el funcionamiento del CMS en diferentes tipos de dispositivos.
- 2) Verificar el funcionamiento individual de las siguientes partes de la aplicación, para garantizar el cumplimiento de los requisitos técnicos y de negocio:
  - Configuración general
  - Diseño
  - Post
  - Páginas
  - Perfil
  - Autenticación
  - Tag
  - Inyección de código
- 3) Construir pruebas automatizadas para usarlas como herramienta de validación y detección de errores en etapas siguientes de desarrollo de la app Ghost.
- 4) Generar un reporte de pruebas para analizar y documentar los incidentes encontrados con la ejecución de las pruebas automatizadas y manuales.
- 5) Encontrar posibles cambios en la interfaz gráfica luego de la corrección de errores, o luego de que se lance una nueva versión.

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

Las pruebas tendrán una duración de 8 semanas donde los recursos tendrán la siguiente participación:

Semana	Actividad	Recurso Humano	Tiempo horas/pers ona
1	Planeación de la estrategia de pruebas	Automatizador 1, 2, 3 y 4	2
1	Pruebas de Reconocimiento Con Monkey	Automatizador 1, 2, 3 y 4	6
2	Pruebas de Reconocimiento Con Ripper	Automatizador 1, 2, 3 y 4	8
3	Pruebas e2e	Automatizador 1, 2, 3 y 4	8
4	Pruebas e2e	Automatizador 1, 2, 3 y 4	8
5	Pruebas de regresión visual	Automatizador 1, 2, 3 y 4	8
6	Pruebas e2e con generación de datos a-priori	Automatizador 1, 2, 3 y 4	8
7	Pruebas e2e con generación de datos aleatorios y pseudo aleatorios	Automatizador 1, 2, 3 y 4	8
8	Pruebas Manuales de las funcionalidades no automatizadas	Automatizador 1, 2, 3 y 4	8

2.2.1 Planeación del proyecto

Listado de Actividades

☰

Plan del proyecto

☆

○ Configurar el estado

FA SC hm nm

Compartir

🔍 Buscar

+

Periodo de prueba: quedan 30 días.  
Agregar información de facturación

FA

Resumen

Lista

Tablero

Cronograma

Calendario

Más...

+ Agregar tarea

Filtrar

Ordenado por: Fecha de inicio

Ocultar

Público para la web: activado

Personalizar

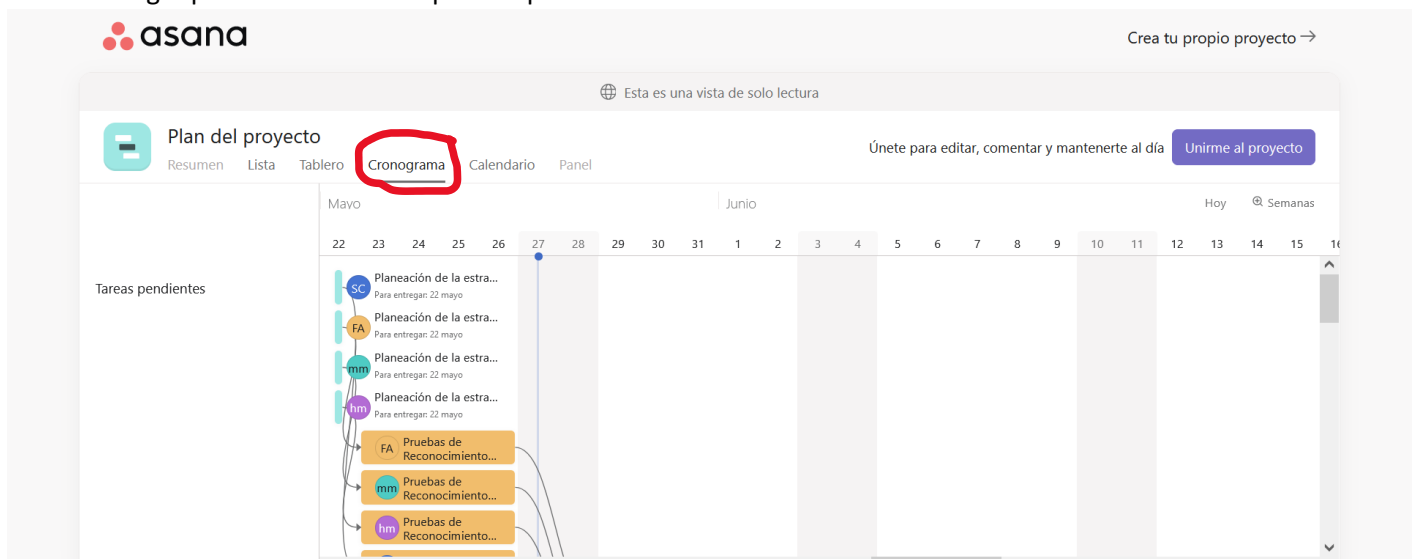
Nombre de la tarea	Responsable	Fecha de...	Prioridad	Estado	
▼ Tareas pendientes					
✓ Planeación de la estrategia de pruebas	SC Santiago Ro...	22 mayo	Bajo	Finalizado	
✓ Planeación de la estrategia de pruebas 1	FA Fredy Alarcon	22 mayo	Bajo	Finalizado	
✓ Planeación de la estrategia de pruebas	nm ma.munozb1...	22 mayo	Bajo	Finalizado	
✓ Planeación de la estrategia de pruebas	hm h.maury@un...	22 mayo	Bajo	Finalizado	
✗ Pruebas de Reconocimiento Con Monkey	FA Fredy Alarcon	23 – 26 mayo	Medio	Finalizado	
✗ Pruebas de Reconocimiento Con Monkey	nm ma.munozb1...	23 – 26 mayo	Medio	Finalizado	
✗ Pruebas de Reconocimiento Con Monkey	hm h.maury@un...	23 – 26 mayo	Medio	Finalizado	
✗ Pruebas de Reconocimiento Con Monkey	SC Santiago Ro...	23 – 26 mayo	Medio	Finalizado	
✗ Pruebas de Reconocimiento Con Ripper	FA Fredy Alarcon	29 mayo – 2 jun	Alto	En curso	
✗ Pruebas de Reconocimiento Con Ripper	nm ma.munozb1...	29 mayo – 2 jun	Alto	En curso	
✗ Pruebas de Reconocimiento Con Ripper	hm h.maury@un...	29 mayo – 2 jun	Alto	En curso	
✗ Pruebas de Reconocimiento Con Ripper	SC Santiago Ro...	29 mayo – 2 jun	Alto	En curso	
✗ Pruebas e2e 1	FA Fredy Alarcon	5 – 16 jun	Alto	Pendient...	
✗ Pruebas e2e	nm ma.munozb1...	5 – 16 jun	Alto	Pendient...	
✗ Pruebas e2e	hm h.maury@un...	5 – 16 jun	Alto	Pendient...	
✗ Pruebas e2e	SC Santiago Ro...	5 – 16 jun	Alto	Pendient...	
✗ Pruebas de regresión visual	FA Fredy Alarcon	19 – 23 jun	Bajo	Pendient...	
✗ Pruebas de regresión visual	nm ma.munozb1...	19 – 23 jun	Bajo	Pendient...	
✗ Pruebas de regresión visual	hm h.maury@un...	19 – 23 jun	Bajo	Pendient...	
✗ Pruebas de regresión visual	SC Santiago Ro...	19 – 23 jun	Bajo	Pendient...	
✗ Pruebas e2e con generación de datos a-priori 1	FA Fredy Alarcon	26 – 30 jun	Medio	Pendient...	
✗ Pruebas e2e con generación de datos a-priori	nm ma.munozb1...	26 – 30 jun	Medio	Pendient...	
✗ Pruebas e2e con generación de datos a-priori	hm h.maury@un...	26 – 30 jun	Medio	Pendient...	

☒ Pruebas e2e con generación de datos a-priori 1	FA	Fredy Alarcon	26 – 30 jun	Medio	Pendient...	
☒ Pruebas e2e con generación de datos a-priori	mm	ma.munozb1...	26 – 30 jun	Medio	Pendient...	
☒ Pruebas e2e con generación de datos a-priori	hm	h.maury@un...	26 – 30 jun	Medio	Pendient...	
☒ Pruebas e2e con generación de datos a-priori	SC	Santiago Ro...	26 – 30 jun	Medio	Pendient...	
☒ Pruebas e2e con generación de datos aleatorios y pseudo aleatorios	FA	Fredy Alarcon	3 – 7 jul	Alto	Pendient...	
☒ Pruebas e2e con generación de datos aleatorios y pseudo aleatorios	mm	ma.munozb1...	3 – 7 jul	Alto	Pendient...	
☒ Pruebas e2e con generación de datos aleatorios y pseudo aleatorios	hm	h.maury@un...	3 – 7 jul	Alto	Pendient...	
☒ Pruebas e2e con generación de datos aleatorios y pseudo aleatorios	SC	Santiago Ro...	3 – 7 jul	Alto	Pendient...	
☒ Pruebas Manuales de las funcionalidades no automatizadas	FA	Fredy Alarcon	10 – 14 jul	Bajo	Pendient...	
☒ Pruebas Manuales de las funcionalidades no automatizadas	mm	ma.munozb1...	10 – 14 jul	Bajo	Pendient...	
☒ Pruebas Manuales de las funcionalidades no automatizadas	hm	h.maury@un...	10 – 14 jul	Bajo	Pendient...	
☒ Pruebas Manuales de las funcionalidades no automatizadas	SC	Santiago Ro...	10 – 14 jul	Bajo	Pendient...	

- **Cronograma y distribución de actividades**

En el siguiente enlace se podrá encontrar en detalle la planeación de la estrategia junto con los recursos y duración de cada actividad.

**Nota:** La herramienta cuenta un menú de opciones que permiten visualizar y detallar diferentes aspectos del proyecto. Puede navegar por los diferentes aspectos que desee observar.



<https://app.asana.com/read-only/Plan-del-proyecto/983980147401890/8c52d9249eb342a006612f92fcdab48b/list>

## 2.3. Presupuesto de pruebas:

### 2.3.1. Recursos Humanos

- Cuatro ingenieros automatizadores Senior:
  - ✓ Cada uno cuenta con estimado de trabajo de 64 horas.
  - ✓ Deben contar con más de 3 años de experiencia en automatización de pruebas.

### 2.3.2. Recursos Computacionales

- Cuatro computadores portátiles:
  - ✓ Lenovo Thinkpad Intel core i7 – SSD – 16 gb ram
- Servicio en AWS
  - ✓ Cuenta con 320 horas / maquina
  - ✓ Tipo Amazon Elastic Compute Cloud (EC2)
  - ✓ Máquina m7g.4xlarge
  - ✓ 32 GB de memoria

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

Recurso	Presupuesto mensual	Valor hora	Cantidad	Horas	Presupuesto
Ingeniero automatizador Senior	\$ 7.000.000	\$ 43.750	4	64	\$ 11.200.000
Computadores portátiles x 2 meses	\$ 1.200.000		4		\$ 9.600.000
Servicio AWS		\$ 2.525	1	320	\$808.000
<b>Total Presupuesto</b>					<b>\$21.608.000 COP</b>

- Recursos humanos:
  - ✓ Para la determinación de los recursos económicos se realiza en base en los salarios del mercado colombiano, se determinó que el salario promedio para el ingeniero automatizador senior con 3 años de experiencia es de 7.000.000 COP para 1 mes de trabajo. Valor hora Senior = 43.750 COP
  - ✓ Se usan los siguientes links como ejemplo para la definición del presupuesto de los recursos humanos.

<https://co.computrabajo.com/trabajo-de-automatizador-de-pruebas#86ADFC13DDB9B35661373E686DCF3405>

[https://www.elempleo.com/co/ofertas-trabajo/automatizador-ga/1885749706?utm\\_campaign=google\\_jobs\\_apply&utm\\_source=google\\_jobs\\_apply&utm\\_medium=organic](https://www.elempleo.com/co/ofertas-trabajo/automatizador-ga/1885749706?utm_campaign=google_jobs_apply&utm_source=google_jobs_apply&utm_medium=organic)

[https://co.sercanto.com/detail/a/eua355-automatizador-de-pruebas-c4\\_colombia\\_60948793?utm\\_campaign=google\\_jobs\\_apply&utm\\_source=google\\_jobs\\_apply&utm\\_medium=organic](https://co.sercanto.com/detail/a/eua355-automatizador-de-pruebas-c4_colombia_60948793?utm_campaign=google_jobs_apply&utm_source=google_jobs_apply&utm_medium=organic)

- Recursos computacionales:
  - ✓ Computadores portátiles: para los ingenieros junior pueden alquilarse. Basados en la experiencia del equipo contratando con la empresa que está en el enlace (MilenioPc), el valor del alquiler ronda \$1.200.000 COP/mes y normalmente no es posible alquilarlos por menos tiempo, por lo que para los 3 equipos asumiríamos un pago de \$3.600.000 COP. <https://www.mileniopc.com/productos-y-servicios/alquiler-de-portatiles/>
  - ✓ Servicio AWS: el servicio tiene un valor hora \$0.5448 USD aproximadamente \$2.525 COP, el valor correspondiente a las 320 horas que estaría disponible el servicio es de \$808.000 COP de acuerdo a la página oficial de AWS: <https://aws.amazon.com/es/ec2/spot/pricing/?p=ft&c=wa&z=2>.

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

Nivel	Tipo	Técnica	Objetivo
Sistema	Funcionales, Caja Negra, Enfoque positivo	Automatizadas, Exploratorias, Monkey	1,3,4
Sistema	Funcionales, Caja Negra, Enfoque positivo	Automatizadas, Exploratorias, Ripper	1,3,4
Sistema	Funcionales, Caja negra, enfoque positivo y negativo	Automatizadas e2e	2,3,4
Sistema	Funcionales, Caja negra, enfoque positivo y negativo	Automatizadas, e2e, usando generación de datos a priori	2,3,4
Sistema	Funcionales, Caja negra, enfoque positivo y negativo	Automatizadas, e2e, usando generación de datos dinámicos	2,3,4
Sistema	Funcionales, Caja negra, enfoque positivo y negativo	Automatizadas, e2e, usando generación de datos aleatorios	2,3,4
Aceptación	No Funcionales, Caja Negra, enfoque positivo	Automatizadas, Regresión Visual	4,5
Sistema	No Funcionales, Caja negra	Manual, Exploratoria, Eventos creados manualmente	1,4
Sistema	Funcionales, Caja negra	Manual, verificación del correcto funcionamiento de las funcionalidades que utilizan diferentes módulos	1,4

Para evaluar el funcionamiento general de la aplicación, se llevarán a cabo pruebas exploratorias a nivel de sistema. Estas pruebas serán automatizadas, utilizando Monkeys y Rippers, ajustando los mismos manualmente para evitar que se queden en bucles sin avanzar. Estas pruebas serán de enfoque positivo con el fin de identificar posibles defectos y explorar las funcionalidades de la aplicación de manera completa.

Las pruebas de reconocimiento permitirán la detección de errores poco comunes con el uso de pruebas aleatorias y con las pruebas sistemáticas se busca abarcar la mayor parte de la aplicación a través del GUI.

Luego, se realizarán pruebas de extremo a extremo utilizando Cypress. Se crearán escenarios de prueba que ejecuten cada uno de los pasos disponibles en las siguientes funcionalidades de la aplicación:

- Configuración general
- Diseño
- Post
- Páginas
- Perfil
- Autenticación
- Tag
- Inyección de código



Se realizarán pruebas tanto de enfoque positivo como de enfoque negativo. Inicialmente utilizando datos estáticos, escritos manualmente por los automatizadores, posteriormente se implementarán 3 estrategias de generación de datos:

- A priori: se creará un conjunto de datos, aunque también serán estáticos, se tendrá la posibilidad de escoger aleatoriamente entre las opciones provistas, lo que diversifica los datos de entrada de la prueba
- Dinámicos (Pseudo Aleatorios): Antes de iniciar las pruebas se generará un pull de datos aleatorios, estos se utilizarán de forma similar a los anteriormente descritos, con la diferencia de que serán diferentes cada vez que se ejecuten las pruebas
- Aleatorios: Se generarán los datos en el tiempo de ejecución, y serán completamente aleatorios.

Como parte del proceso de pruebas, se llevará a cabo un inventario de las pruebas realizadas para verificar la cobertura de las funcionalidades de la aplicación. Después de esto, se realizarán pruebas manuales para las secciones de código que no hayan sido automatizadas. Se realizarán pruebas no funcionales de caja negra con el objetivo de validar el correcto funcionamiento y la usabilidad de la aplicación en diferentes dispositivos como celulares y tabletas. De esta manera, se asegura una correcta experiencia de usuario en múltiples dispositivos y se identifican posibles problemas de visualización y accesibilidad. Adicionalmente, se probará manualmente la integración de los diferentes módulos en funcionalidades específicas, aprovechando la capacidad de los automatizadores de entender el contexto completo de la ABP para generar escenarios diferentes a los ya automatizados.

Cuando se reporten correcciones de errores o el lanzamiento de una nueva versión, se realizarán pruebas de regresión visual, las cuales permitirán verificar que la interfaz gráfica se mantiene igual (o lo suficientemente similar, para que sea tolerable), a menos que los desarrolladores informen de cambios intencionales en la misma.

Al finalizar las 8 semanas, como resultado de la ejecución de las actividades anteriores se espera lograr obtener los siguientes artefactos:

- ✓ Inventario de pruebas y escenarios de pruebas documentados.
- ✓ Reporte de incidentes encontrados.
- ✓ Repositorio con el código de las diferentes pruebas realizadas.
- ✓ Reporte con los resultados de las pruebas automatizadas.
- ✓ Reporte con los resultados de las pruebas manuales.
- ✓ Reporte con los resultados de las pruebas de regresión visual

## 2.5. Distribución de Esfuerzo

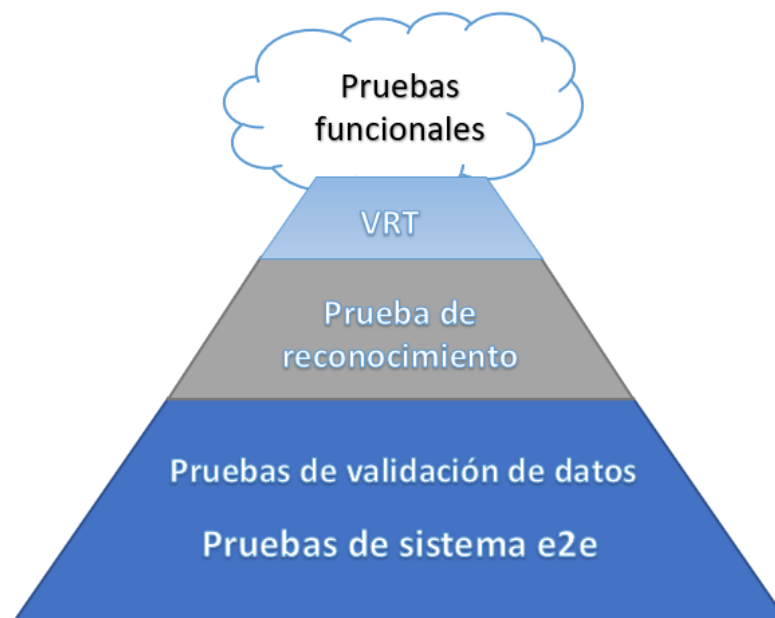
La estrategia escogida teniendo en cuenta el presupuesto asignado y el tiempo disponible total de 64 horas/persona que no permite contar con muchos tipos de pruebas y construir una pirámide completa, en este caso se le dará prioridad a la creación de las pruebas automatizada de sistema (e2e) haciendo el uso de una sola API de automatización (Cypress) con el fin de contar al final del proceso con los casos de pruebas que permitan validar las funcionalidades descritas en los objetivos. Posteriormente los escenarios e2e serán ajustados para contemplar la validación de datos generados aleatoriamente y creados a-priori con el uso de las herramientas Faker y Mockaroo.

Con el fin de aumentar las posibilidades de detectar más incidentes el sistema bajo pruebas también el equipo de pruebas construirá y ejecutará pruebas de reconocimiento basadas en Monkeys y Rippers.

Luego de la creación de los escenarios anteriores estos serán ejecutados en la última versión del sistema con el fin de detectar diferencias significativas que ayuden a la detección de nuevos incidentes a través de las pruebas de regresión visual.

Por último, se crearán escenarios de pruebas manuales para las funcionalidades del sistema que por su complejidad no fueron cubiertas por la automatización de pruebas de sistema.

Teniendo en cuenta lo anterior la distribución de esfuerzo se sería algo similar a la siguiente imagen:



**Anexo:**

1. Inventario de prueba exploratorias
2. Resultados estrategia semana 1
3. Análisis de los pros contras de las herramientas usadas
4. Limitaciones, costos adicionales al presupuesto, ventajas y desventajas del proceso de pruebas propuesto para las 8 semana.
5. Vídeo estrategia link.

Repositorio: <https://github.com/fredyalarcon/pruebas-grupo-15>

Wiki: <https://github.com/fredyalarcon/pruebas-grupo-15/wiki>