

Estrategia de Pruebas

El presente documento presenta la estrategia de pruebas para la plataforma de gestión de contenidos GHOST, elaborada por The Software Design Company y teniendo en cuenta el Presupuesto 1, distribuido en recursos humanos, computacionales y económicos. La explicación audiovisual se encuentra en el siguiente enlace: https://drive.google.com/file/d/1YZmnqcfbPI_QwRivCOvhsevp9Ogy3mUB/view?usp=sharing

1. Aplicación Bajo Pruebas

1.1. **Nombre Aplicación:** GHOST

1.2. **Versión:** 3.42.5

1.3. **Descripción:** Ghost es una nueva plataforma centrada exclusivamente en la publicación profesional; puesto que provee las mejores herramientas de código abierto para periodistas y escritores independientes de todo el mundo, creando un impacto real en el futuro de los medios de comunicación online.

1.4. **Funcionalidades Core:**

A continuación, se presenta el listado del conjunto de funcionalidades seleccionadas para detallar y someter a pruebas.

- **Inicio de sesión:**
Funcionalidad de autenticación y acceso a la plataforma administrativa de Ghost.
- **Creación de post:**
La creación de contenido de tipo publicación es una de las funcionalidades centrales de Ghost. En esta característica es posible la construcción del post mediante texto plano, herramientas primarias y herramientas embebidas.
- **Edición de post:**
Esta funcionalidad permite a partir de un post seleccionado, editar su contenido con el mismo mecanismo que fue creado.
- **Creación de página:**
La creación de páginas es otra de las funcionalidades centrales de Ghost. En esta característica es posible la construcción de páginas adicionales en los módulos de presentación mediante texto plano, herramientas primarias y herramientas embebidas.
- **Edición de página:**
Esta funcionalidad permite a partir de una página seleccionada, editar su contenido con el mismo mecanismo que fue creado.
- **Creación de tag:**
Las etiquetas o categorías asociadas a los posts, sirven como mecanismo de búsqueda, organización y presentación de contenido. En esta característica es posible diligenciar el formulario de registro de un nuevo tag.
- **Invitación de usuarios (staff):**

La adición de nuevos miembros que participen de diferentes modos en la construcción, publicación y gestión de contenido, es otra de las funcionalidades principales de la aplicación bajo pruebas. En esta característica es posible diligenciar el formulario de invitación de nuevos usuarios.

- **Perfil de usuario (staff):**

El módulo de perfil de usuarios permite mantener la información de cada miembro del equipo actualizada y ordenada. En esta característica se realiza la gestión del formulario con los campos asociados al perfil de usuario.

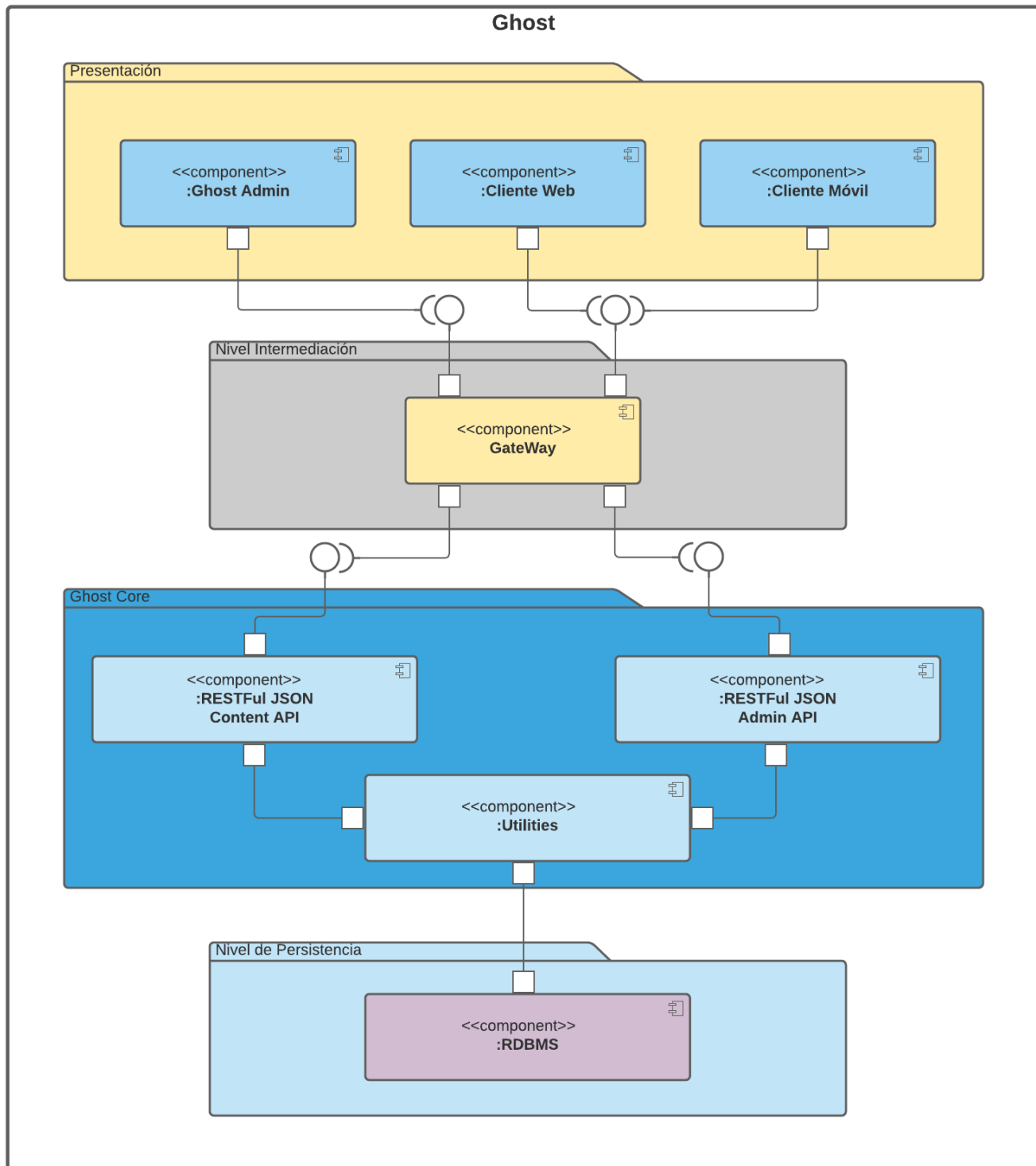
- **Cambio de contraseña:**

Esta funcionalidad permite actualizar las credenciales actuales del usuario.

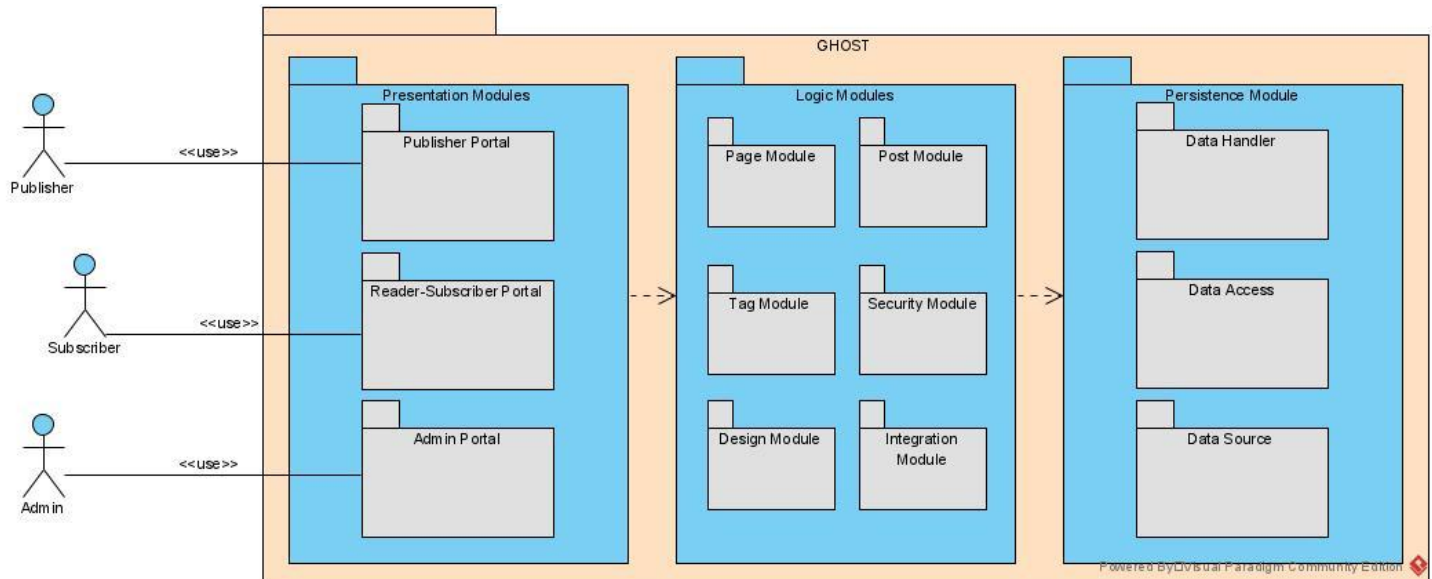
- **Diseño – sección navegación:**

La sección de diseño de navegación permite ordenar y referenciar las rutas asociadas a las páginas creadas.

1.5. Diagrama de Arquitectura:



1.6. Diagrama de Contexto:



1.7. Modelo de Datos:

El modelo de datos de GHOST enseña la gran distribución y el buen flujo de la información que persiste en el sistema. Dado el tamaño que presenta este modelo, se presenta como un enlace a un archivo externo para su correcta visualización.

<https://drive.google.com/file/d/1vjwfSjgdmCPxeHIEWhw-bS63kE71kDfV/view?usp=sharing>

1.8. Modelo de GUI:

El modelo de GUI de GHOST permite entrar en contexto sobre el enfoque y las funcionalidades principales de la aplicación. Dado el tamaño que presenta este modelo, se presenta como un enlace a un archivo externo para su correcta visualización.

<https://drive.google.com/file/d/1giFkre1k9do936P5hjQ-N8hNS6OL0NNG/view?usp=sharing>

2. Contexto de la estrategia de pruebas

2.1. Objetivos:

2.1.1. General:

Reducir la probabilidad de que se manifiesten defectos cuando el software está en producción por medio de la implementación de pruebas automatizadas que midan los atributos de calidad de latencia y escalabilidad, maximizando la utilización de recursos disponibles

2.1.2. Específicos

- Validar de forma automática y con la ayuda de un motor de integración continua, el funcionamiento de componentes de forma aislada.
- Validar comportamientos inesperados del sistema por medio de la generación de eventos y datos aleatorios sin intervención humana.
- Validar de forma manual las funcionalidades de la aplicación de ghost en busca de errores.
- Validar de forma automática las funcionalidades de la aplicación de ghost.
- Realizar una comparación entre las características visuales de ghost versión 3.3.0 y 3.42.5.
- Verificar de forma automática y con la ayuda de un motor de integración continua, escenarios de validación de datos generados a partir de colecciones generadas a-priori, pseudo aleatorias y aleatorias dinámicas.

2.2. Duración de la iteración de pruebas :

La duración de la iteración de pruebas será igual a la duración del sprint de desarrollo, la cual será de 1 semanas durante dos meses

Cada iteración de pruebas comprende los siguientes tiempos:

- Diseño del plan de pruebas
- Elaboración de Pruebas
- Elaboración de Informes y análisis de resultados

2.3. Presupuesto de pruebas:

2.3.1. Recursos Humanos

Para ejecutar la estrategia de pruebas se decide contratar cuatro ingenieros automatizadores senior, con una asignación de 8 horas por semana cada uno, las cuales serán distribuidas en el diseño, configuración y ejecución del proceso de automatización

2.3.2. Recursos Computacionales

Para la ejecución de las pruebas se propone el uso de 8 horas semanales de recursos de cómputo para ejecutar las pruebas realizadas de manera centralizada, esto con el fin de evitar la divergencia en resultados de las pruebas a causa del ambiente usado para la codificación de las mismas.

Se utilizará Amazon Elastic Compute Cloud (Amazon EC2) la cual proporciona capacidad de computación escalable en la nube permitiendo lanzar tantos servidores virtuales como necesite, configurar la seguridad y las redes y administrar el almacenamiento.

A continuación se referencian algunos recursos oficiales sobre la utilización de los servicios de AWS para la ejecución de pruebas:

<https://aws.amazon.com/ec2/pricing/>

The table shows current software and infrastructure pricing for services hosted in **US East (N. Virginia)**. Additional taxes or fees may apply.

Use of Local Zones or WaveLength infrastructure deployment may alter your final pricing.

Astadia UI Automation Test Engineering Environment			
EC2 Instance type	Software/hr	EC2/hr	Total/hr
<input checked="" type="radio"/> t2.medium	\$0	\$0.046	\$0.046
<input type="radio"/> t2.large	\$0	\$0.093	\$0.093
<input type="radio"/> t2.xlarge	\$0	\$0.186	\$0.186
<input type="radio"/> t2.2xlarge	\$0	\$0.371	\$0.371
<input type="radio"/> m3.medium	\$0	\$0.067	\$0.067
<input type="radio"/> m3.large	\$0	\$0.133	\$0.133

You are charged once a month for the following storage volumes.

Volume Type	Rate
EBS General Purpose SSD (gp2) volumes	• \$0.10 per GB-month of provisioned storage

The estimated cost assumes [on-demand EC2 instances](#) and does not include [data transfer fees](#). For lower prices you can utilize [reserved instances](#) or [spot instances](#).

Instancia EC2	Costo/hora (USD)	Total de horas por semana	Total de horas 8 semanas	Costo total por semana (USD)
t2.medium	0,046	8	64	2,994

Se solicita un presupuesto de 3 USD para ejecutar las pruebas desde un ambiente centralizado.

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

Para la presente estrategia de pruebas no se contratarán servicios de outsourcing externos, toda la ejecución de la estrategia correrá por cuenta del personal dispuesto para realizar las pruebas (cuatro testers senior).

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

A continuación se describen la técnica, los niveles y el tipo de prueba que serán implementados para la medición de atributos de calidad.

Nivel	Tipo	Técnica	Objetivo
Unidad	Funcionales de Caja Blanca, Positivas	API de automatización	Validar de forma automática y con la ayuda de un motor de integración continua, el funcionamiento de componentes de forma aislada
Sistema	Caja negra, positivas, negativas.	Manuales	Validar de forma manual las funcionalidades de la aplicación de ghost en busca de errores.
Sistema	No Funcionales	API de automatización (Reconocimiento)	Validar comportamientos inesperados del sistema por medio de la generación de eventos y datos aleatorios sin intervención humana
Sistema	Funcionales de Caja Negra	E2E API de automatización	Validar de forma automática las funcionalidades de la aplicación de ghost.
Sistema	No Funcionales de Caja Negra, Positivas	Pruebas VRT Regresión visual	Realizar una comparación entre las características visuales de ghost versión 3.3.0 y 3.42.5.
Sistema	No Funcional Positivas y Negativas	Validación de Datos	Verificar de forma automática y con la ayuda de un motor de integración continua, escenarios de validación de datos generados a partir de colecciones generadas a-priori, pseudo aleatorias y aleatorias dinámicas.

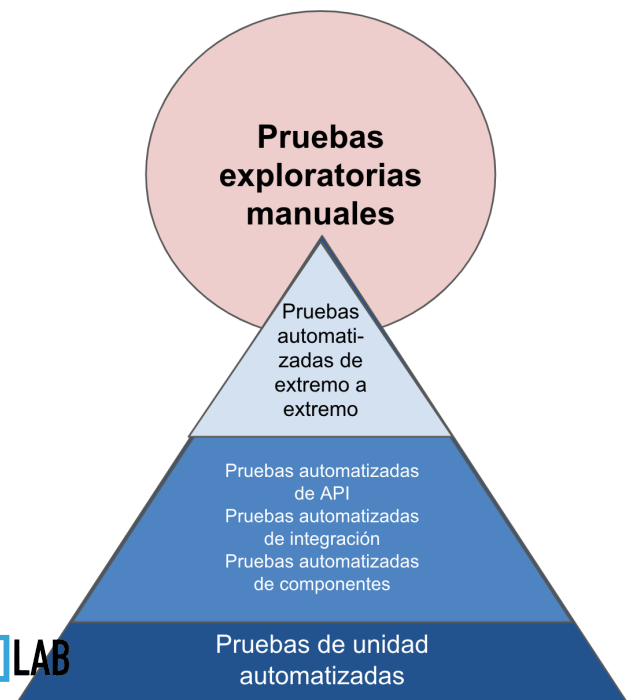
2.5. Distribución de Esfuerzo

La distribución de esfuerzo que se planea para la presente estrategia está basada en la pirámide de automatización, haciendo énfasis en las pruebas que requieren menor costo y que sean rápidas de implementar, además de la capacidad que tengan para detectar el origen de un defecto, todo ejecutado bajo un esquema de integración continua que permita identificar cuándo ocurre un fallo sin esperar al final del "sprint"

- Se implementará un enfoque de pruebas continuas implementando pruebas unitarias exigidas al equipo de desarrollo. Las pruebas de integración y a nivel de sistema se implementará

Plantilla elabora

THE SW DESIGN LAB



en paralelo durante todo el desarrollo del sprint utilizando un total de 32 horas de ingenieros automatizadores y hasta 8 horas de equipos e infraestructura semanales durante 2 meses.

- Se garantizarán los resultados de las pruebas por medio de la repetición de las mismas para identificar pruebas inestables (Flaky)

2.5.1 Pruebas Manuales y de Reconocimiento (Moneys y Rippers)

La estrategia de pruebas plantea iniciar en las primeras tres iteraciones (semanas) con la ejecución de pruebas manuales en paralelo con pruebas de reconocimiento (Monkey y Rippers), en primera instancia implementando pruebas aleatorias con la herramienta Monkey-Cypress la cual permitirá generar eventos y datos sin intervención humana, simulando la interacción con Ghost, inyectando un alto volumen de eventos y datos en muy poco tiempo, ayudando a detectar casos raros.

Se implementará una traza de los eventos generados que permita localizar el error en el código y entender el defecto con el fin de permitir al equipo hacer pruebas de forma continua.

Posteriormente se procederá con la implementación de la herramienta RIPuppet, la cual permitirá la ejecución autónoma de la aplicación con mínima intervención humana para definir los escenarios adecuados. En este caso la herramienta empezará a generar eventos automáticamente, buscando identificar combinaciones de entradas que llevaran a un comportamiento inesperado haciendo uso de una heurística de exploración predefinida al mismo tiempo que va recolectando información respecto a errores "crashes" o excepciones, construyendo un modelo de los componentes visibles, su ubicación, sus características y el tipo de interacción que se puede establecer entre éstos.

La herramienta almacenará un registro de los eventos que se irán ejecutando y los estados visitados, construyendo un grafo de estados y transiciones que facilitará la exploración sistemática y que podrá ser utilizado luego para generar casos de pruebas adicionales.

A continuación se presenta la distribución de tareas detallada para las pruebas de reconocimiento por semana:

Semana 1				
Fecha	Actividad	Responsable	Horas Hombre	Horas Maquina
06/01/2021	Diseñar la estrategia de pruebas	Ingeniero automatizador senior	4	
06/02/2021	Capacitación frameworks de automatización	Ingeniero automatizador senior	2	
06/03/2021	Diseñar casos de pruebas Manuales y de Reconocimiento (Monkeys & Rippers) #1 detallando los pasos, las pre condiciones y el ambientes de ejecución	Ingeniero automatizador senior	4	
06/04/2021	Documentación y Socialización del diseño para pruebas Manuales y de Reconocimiento (Monkeys & Rippers) #1	Ingeniero automatizador senior	4	

06/04/2021	Creación y configuración de cuenta AWS	Ingeniero automatizador senior	4	
06/04/2021	Creación de máquinas virtuales Amazon EC2	Ingeniero automatizador senior	4	
06/05/2021	Documentación y Configuración de laboratorio de pruebas Manuales y de Reconocimiento (Monkeys & Rippers) #1 y ambiente cloud del SUT	Ingeniero automatizador senior	4	
06/05/2021	Creación de set de pruebas Manuales y de Reconocimiento (Monkeys & Rippers) #1	Ingeniero automatizador senior	4	
06/06/2021	Ejecución de set de pruebas Manuales y de Reconocimiento (Monkeys & Rippers) #1	Amazon AWS		8
06/06/2021	Documentación e Informe de resultados de Monkeys y Ripper	Ingeniero automatizador senior	2	
		Total Semana 1	32	8

Semana 2				
Fecha	Actividad	Responsable	Horas Hombre	Horas Maquina
06/07/2021	Diseñar casos de pruebas Manuales y de Reconocimiento (Monkeys & Rippers) #2 detallando los pasos, las pre condiciones y el ambientes de ejecución	Ingeniero automatizador senior	8	
06/08/2021	Documentación y Socialización del diseño para pruebas Manuales y de Reconocimiento (Monkeys & Rippers) #2	Ingeniero automatizador senior	8	
06/09/2021	Creación de máquinas virtuales Amazon EC2	Ingeniero automatizador senior	2	
06/10/2021	Creación de set de pruebas Manuales y de Reconocimiento (Monkeys & Rippers) #2	Ingeniero automatizador senior	8	
06/11/2021	Ejecución de set de pruebas Manuales y de Reconocimiento (Monkeys & Rippers) #2	Amazon AWS	2	8
06/12/2021	Documentación e Informe de resultados de la iteración	Ingeniero automatizador senior	4	
		Total Semana 2	32	8

Semana 3				
Fecha	Actividad	Responsable	Horas Hombre	Horas Maquina

06/13/2021	Diseñar casos de pruebas Manuales y de Reconocimiento (Monkeys & Rippers) #3 detallando los pasos, las pre condiciones y el ambientes de ejecución	Ingeniero automatizador senior	8	
06/14/2021	Documentación y Socialización del diseño para pruebas Manuales y de Reconocimiento (Monkeys & Rippers) #3	Ingeniero automatizador senior	8	
06/15/2021	Creación de máquinas virtuales Amazon EC2	Ingeniero automatizador senior	2	
06/16/2021	Creación de set de pruebas Manuales y de Reconocimiento (Monkeys & Rippers) #3	Ingeniero automatizador senior	8	
06/17/2021	Ejecución de set de pruebas Manuales y de Reconocimiento (Monkeys & Rippers) #3	Amazon AWS	2	8
06/18/2021	Documentación e Informe de resultados de la iteración	Ingeniero automatizador senior	4	
		Total Semana 3	32	8

2.5.2 E2E

Hemos decidido incluir en nuestra estrategia de pruebas la realización de pruebas e2e usando un api de automatización, ya que este tipo de pruebas nos permiten validar las diferentes funcionalidades que posee la aplicación. Para este tipo de pruebas hemos decidido utilizar el api de automatización Cypress debido al conocimiento que poseen los ingenieros en la herramienta lo cual nos evita destinar demasiado tiempo en la capacitación de su uso y a la facilidad de leer el reporte de los resultados obtenidos de la ejecución de la suite de pruebas.

Dado que este tipo de pruebas requieren de tiempo para la codificación de los escenarios, se decidió dedicar 3 semanas a la creación de los escenarios y se proponen 8 horas de recursos de cómputo cada semana para la ejecución centralizada de las pruebas realizadas.

A continuación se presenta la distribución de tareas detallada para las pruebas E2E por semana:

Semana 4				
Fecha	Actividad	Responsable	Horas Hombre	Horas Maquina
06/19/2021	Diseñar casos de pruebas E2E utilizando API de Automatización (Cypress) detallando los pasos, las pre condiciones y el ambientes de ejecución	Ingeniero automatizador senior	4	
06/20/2021	Documentación y Socialización del diseño para pruebas E2E utilizando API de Automatización (Cypress)	Ingeniero automatizador senior	4	
06/21/2021	Creación de máquinas virtuales Amazon EC2	Ingeniero automatizador senior	4	

06/22/2021	Creación de set de pruebas E2E utilizando API de Automatización (Cypress)	Ingeniero automatizador senior	9	
06/23/2021	Creación de set de pruebas E2E utilizando API de Automatización (Cypress)	Ingeniero automatizador senior	9	
06/24/2021	Ejecución de set de pruebas E2E utilizando API de Automatización (Cypress)	Amazon AWS		8
06/25/2021	Documentación e Informe de resultados de la iteración	Ingeniero automatizador senior	2	
		Total Semana 4	32	8

Semana 5				
Fecha	Actividad	Responsable	Horas Hombre	Horas Maquina
06/26/2021	Diseñar casos de pruebas E2E utilizando API de Automatización (Cypress) detallando los pasos, las pre condiciones y el ambientes de ejecución	Ingeniero automatizador senior	4	
06/27/2021	Documentación y Socialización del diseño para pruebas E2E utilizando API de Automatización (Cypress)	Ingeniero automatizador senior	4	
06/28/2021	Creación de set de pruebas E2E utilizando API de Automatización (Cypress)	Ingeniero automatizador senior	4	
06/29/2021	Creación de set de pruebas E2E utilizando API de Automatización (Cypress)	Ingeniero automatizador senior	9	
06/30/2021	Creación de set de pruebas E2E utilizando API de Automatización (Cypress)	Ingeniero automatizador senior	9	
07/01/2021	Ejecución de set de pruebas E2E utilizando API de Automatización (Cypress)	Amazon AWS		8
07/02/2021	Documentación e Informe de resultados de la iteración	Ingeniero automatizador senior	2	
		Total Semana 5	32	8

Semana 6				
Fecha	Actividad	Responsable	Horas Hombre	Horas Maquina
07/03/2021	Diseñar casos de pruebas E2E utilizando API de Automatización (Cypress) detallando los pasos, las pre condiciones y el ambientes de ejecución	Ingeniero automatizador senior	4	
07/04/2021	Documentación y Socialización del diseño para pruebas E2E utilizando API de Automatización (Cypress)	Ingeniero automatizador senior	4	
07/05/2021	Creación de set de pruebas E2E utilizando API de Automatización (Cypress)	Ingeniero automatizador senior	4	
07/06/2021	Creación de set de pruebas E2E utilizando API de Automatización (Cypress)	Ingeniero automatizador senior	9	
07/07/2021	Creación de set de pruebas E2E utilizando API de Automatización (Cypress)	Ingeniero automatizador senior	9	
07/08/2021	Ejecución de set de pruebas E2E utilizando	Amazon AWS		8

	API de Automatización (Cypress)			
07/09/2021	Documentación e Informe de resultados de la iteración	Ingeniero automatizador senior	2	
		Total Semana 6	32	8

2.5.3 Visual Regression Testing (VRT)

Teniendo en cuenta que Ghost es una aplicación que se encuentra en constante cambio, hemos incluido en esta estrategia las pruebas de regresión visual con el fin de verificar que dada una versión original n de la que conocemos el comportamiento y una versión $n+1$, los cambios en $n+1$ no incluye defectos. Con estas pruebas validamos que lo que funcionaba en una versión sigue funcionando una versión posterior.

Si tenemos una versión nueva del sistema bajo pruebas y necesitamos detectar regresiones en el código, usamos los mismos casos de pruebas que se usaron antes, lo que nos permite validar que la nueva versión no inyecta defectos en el sistema.

Dado que esta estrategia de pruebas es para la versión 3.42.5 de Ghost, las pruebas de regresión se realizan contra la versión 3.3.0 y se utiliza como base la suite de pruebas usadas en las pruebas e2e (puede ver la descripción de estas pruebas en este documento en el apartado 2.5.2).

Las herramientas utilizadas para la ejecución de las pruebas de regresión son *resemble* y *backstop*:

Resemble: Esta herramienta nos ayuda a generar las imágenes para realizar la comparación entre las versiones, pero no provee un reporte consolidado con los resultados, por lo que los testers contratados para la ejecución de esta estrategia desarrollaron un script que permite generar el reporte con los resultados.

A continuación, una breve descripción para ejecutar las pruebas y obtener el reporte de resultados:

- Ejecutar Cypress para las pruebas 3.3.0, cambiando la versión en las variables de ambiente en `cypress/cypress.json` por el valor 3.3.0 y el entorno en `fixture/environment.json`
- Ejecutar Cypress para las pruebas 3.42.5, cambiando la versión en las variables de ambiente en `cypress/cypress.json` por el valor 3.42.5 y el entorno en `fixture/environment.json` y posteriormente `cypress run` en la carpeta `cypress`.
- Ingresar a la carpeta `report` y ejecutar los siguientes comandos

```
npm install
node index.js
```

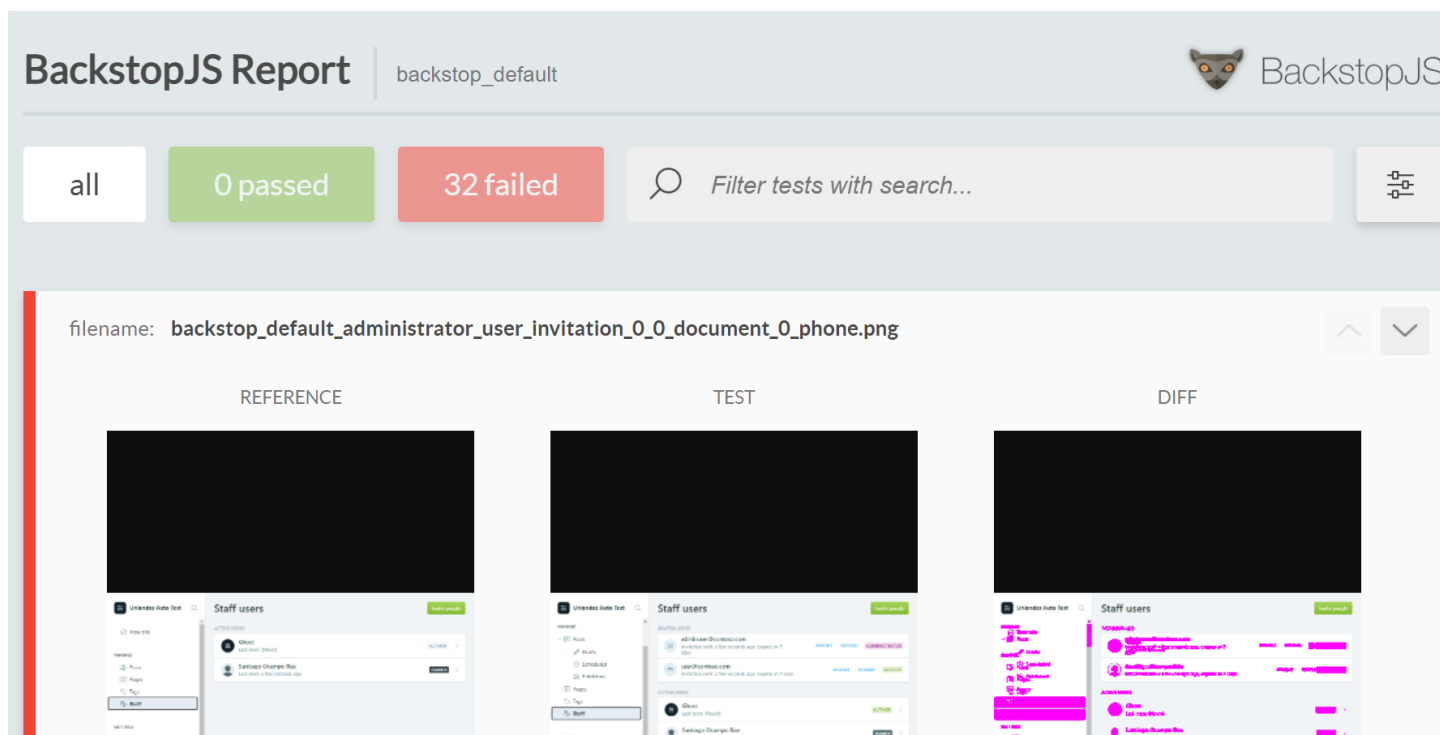
Backstop: Esta herramienta nos ayuda a generar las imágenes para realizar la comparación entre las versiones y nos permite generar el reporte con los resultados.

A continuación, una breve descripción para ejecutar las pruebas y obtener el reporte de resultados:

- Ejecutar Cypress para las pruebas 3.3.0, cambiando la versión en las variables de ambiente en `cypress.json` y el entorno en `fixture/environment.json`
- Ejecutar Cypress para las pruebas 3.42.5, cambiando la versión en las variables de ambiente en `cypress.json` y el entorno en `fixture/environment.json`
- Para ejecutar el reporte desde la terminal, se debe clonar el repositorio y ubicarse en la carpeta `backstop/` que se encuentra dentro del repositorio y ejecutar los siguientes comando:

backstop reference
backstop test

Esto realizará la comparación visual de entre las imágenes de referencia y las objetivo, posteriormente se generará un reporte en formato HTML. Para visualizarlo, es necesario ubicarse en la carpeta backstop/backstop_data/html_report en el archivo index.html e interactuar con los resultados.



A continuación se presenta la distribución de tareas detallada para las pruebas de regresión visual por semana:

Semana 7				
Fecha	Actividad	Responsable	Horas Hombre	Horas Maquina
07/10/2021	Ajustar los casos de prueba utilizados en las pruebas e2e para que se ejecuten de manera correcta en la nueva versión 3.42.5 de Ghost	Ingeniero automatizador senior	8	
07/11/2021	Creación del script que permite generar el reporte de resultados para análisis de regresión visual para resemble	Ingeniero automatizador senior	4	
07/12/2021	Creación del script que permite generar el reporte de resultados para análisis de regresión visual para backstop	Ingeniero automatizador senior	4	
07/13/2021	Documentación y socialización del diseño para pruebas de regresión visual	Ingeniero automatizador senior	6	

07/14/2021	Creación y configuración de máquinas virtuales Amazon EC2	Ingeniero automatizador senior	4	
07/14/2021	Ejecución de set de pruebas para regresión visual en resemble y backstop	Ingeniero automatizador senior	4	
07/14/2021	Ejecución de set de pruebas para regresión visual en resemble y backstop	Amazon AWS		8
07/15/2021	Documentación e Informe de resultados de la iteración	Ingeniero automatizador senior	2	
		Total Semana 7	32	8

2.5.4 Escenarios de Validación

Considerando todos los tipos de usuarios e interacciones que se tienen con el sistema bajo pruebas, es necesaria la construcción de escenarios de validación de datos, en donde se evalúen las capacidades de los formularios ante todo tipo de entradas.

En este punto nos apoyamos de 3 estrategias de generación de datos, las cuales son: pool de datos a-priori, pool de datos (pseudo) aleatorio dinámico y escenario aleatorio. Si bien recordamos que GHOST no es un sistema construido por nuestro equipo, no podemos asegurar la fortaleza o robustez en las validaciones de los campos a menos que sean realizadas rutinas de prueba que incorporen generación de datos. Para llevar esto a cabo nos apoyamos de la herramientas open-source mockaroo para generación de data-pools y faker para la generación de datos aleatorios.

Pensando en la generación de data-pools, es necesaria la definición de esquemas de datos (negativos, positivos) a ser usados en las pruebas; estos esquemas nos permiten adquirir por medio de un archivo fijo o una API, registros aleatorios generados con las especificaciones que requerimos.

My Schemas

CREATE A SCHEMA
RESTORE FROM BACKUP..
DELETE SELECTED

☐ Name

☐ element
title | empty_title | invalid_title | content | invalid_content | publish_date | invalid_publish_date | tags | invalid_tags | meta_description | invalid_meta_description | meta_title | invalid_meta_title

☐ staff-email
email | invalid_email | numeric_email | empty_email | role

☐ tag
name | invalid_name | empty_name | description | invalid_description

☐ test_schema
id | first_name | last_name | email | gender | ip_address | test_field

En el caso de nuestros escenarios, fueron generados archivos en formato JSON como data-pool a-priori, definidas un conjunto de API's como data-pool aleatorio dinámico y los módulos de faker como métodos de generación de datos.

<input type="checkbox"/>	Route
<input type="checkbox"/>	GET /element.json
<input type="checkbox"/>	GET /staff-email.json
<input type="checkbox"/>	GET /tag.json
<input type="checkbox"/>	GET /users.json

A continuación se presenta la distribución de tareas detallada para las pruebas de validación de datos por semana:

Semana 8				
Fecha	Actividad	Responsable	Horas Hombre	Horas Maquina
07/16/2021	Diseñar casos de pruebas de validación de datos de entrada detallando los pasos, las pre condiciones y el ambientes de ejecución	Ingeniero automatizador senior	8	
07/17/2021	Documentación y socialización del diseño para pruebas de validación de datos de entrada	Ingeniero automatizador senior	8	
07/18/2021	Creación y configuración de máquinas virtuales Amazon EC2	Ingeniero automatizador senior	8	
07/19/2021	Creación de set de pruebas de validación de datos de entrada	Ingeniero automatizador senior	6	
07/20/2021	Ejecución de set de pruebas de validación de datos de entrada	Amazon AWS		8
07/21/2021	Documentación e Informe de resultados de la iteración	Ingeniero automatizador senior	2	
		Total Semana 8	32	8

Enlace al video

https://drive.google.com/file/d/1YZmnqcfbPI_QwRivCOvhsevp9Ogy3mUB/view?usp=sharing

Plantilla elaborada por

THE SW DESIGN LAB