

# Estrategia de Pruebas

## 1. Aplicación Bajo Pruebas

Video Presentación:

[PruebasAutomatizadasPresentacion.mp4](#)

### Grupo 4 – Integrantes

Jose Luis Garay Urbina

Juan Sebastián Becerra Bautista

Alexander Mora Ruiz

**1.1. Nombre Aplicación:** Ghost

**1.2. Versión:** 3.41.1

**1.3. Descripción:**

Ghost es un sistema de gestión de contenidos de código abierto escrito en Javascript y distribuido bajo la licencia MIT, fue creado por John O’Nolan antiguo miembro del equipo de Wordpress. Está desarrollado utilizando Node.js para el servidor y Ember.js para el cliente y puede ser utilizado como CMS tradicional o *headless*.

**1.4. Funcionalidades Core:**

#	Funcionalidad	Descripción
1	Crear una Entrada	Permite crear una Entrada para el sitio web, se debe al ingresar un título y el contenido. También se puede agrega una Imagen destacada, Etiquetas, Metadatos para redes sociales y se puede configurar la fecha de publicación
2	Ver todas las Entradas	Nos permite ver todas las Entradas creadas, con su respectivo título, estado y qué usuario la publicó y en qué fecha. Además permite filtrar por estado, tipo de acceso, autor y Etiquetas
3	Publicar una Entrada	Permite publicar una Entrada cambiandola a estado Publicado. Se puede configurar la forma de publicación, si se desea enviar un correo a interesados o no, y también se puede programar la fecha de publicación.
4	Eliminar una Entrada	Permite eliminar Entradas, es necesario confirmar la eliminación de una Entrada para proceder.
5	Crear una Etiqueta	Se pueden crear Etiquetas para asignarla a las Entradas. Se puede ingresar un nombre, una descripción, un color, una imagen, la url de referencia y Metadatos para las redes sociales.
6	Eliminar una Etiqueta	Permite eliminar una Etiqueta. Muestra si la Etiqueta ya está asignada a una Entrada y se debe confirmar la eliminación.
7	Ver todas las Etiquetas	Muestra el listado de Etiquetas que se tiene, permite ver el nombre de la Etiqueta, la url de referencia y el número de entradas a la que está asignada

8	Crear un Usuario	Permite crear Usuarios para que hagan uso de la plataforma. Se debe agregar un nombre, correo, una etiqueta, una nota y se puede elegir si se desea inscribir el correo en el Newsletter
9	Crear una Página	Permite crear una nueva página para el sitio web, se debe al ingresar un título y el contenido. También se puede agrega una Imagen destacada, la Url, Etiquetas, Metadatos para redes sociales y se puede cofigurar la fecha de publicación
10	Agregar una imagen a una Página	Permite agregar una imagen desde el dispositivo local a la página que se está editando

## 1.5. Diagrama de Arquitectura:

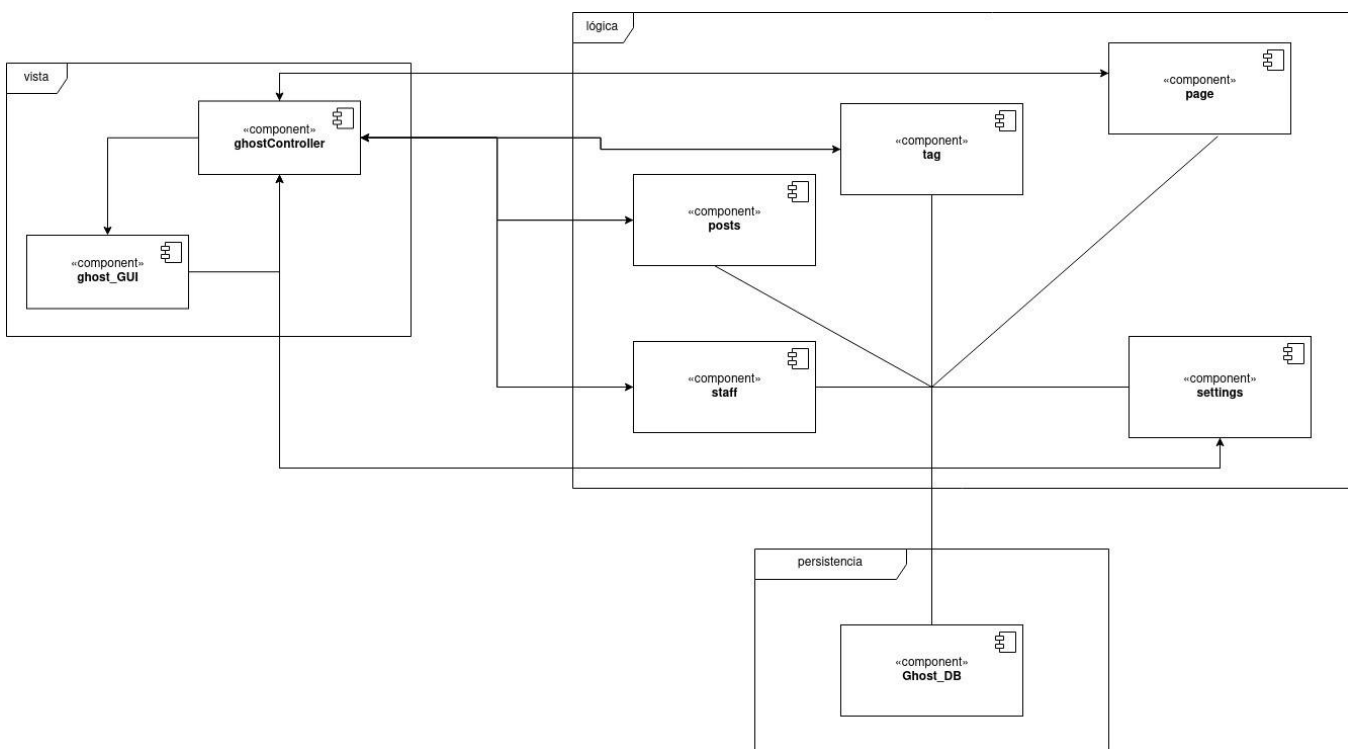


Diagrama componentes: [Modelo componentes Ghost](#)

## 1.6. Diagrama de Contexto:





- 5) Probar la robustez de la aplicación mediante la realización de pruebas E2E utilizando generación de datos a priori, pseudoaleatorios y aleatorios.

## 2.2. Duración de la iteración de pruebas: 8 Semanas

## 2.3. Presupuesto de pruebas:

### 2.3.1. Recursos Humanos

3 Ingenieros automatizadores con dedicación de 8 horas por semana durante 8 semanas

Integrantes	Perfil	Tiempo (h)
Alexander Mora	Ingeniero automatizador Senior	64
Jose Luis Garay	Ingeniero automatizador Senior	64
Juan Sebastián Becerra	Ingeniero automatizador Senior	64

Estos ingenieros pertenecen al proyecto, tienen un conocimiento adecuado sobre las funcionalidades a probar y el cómo están construidas por lo cual no es necesario utilizar tiempo de capacitación sobre la aplicación. Adicional se cuenta con una documentación de pruebas exploratorias la cual incluye videos de las funcionalidades, reporte de incidencias ya encontradas y el flujo de las funcionalidades de la aplicación GHOST.

### 2.3.2. Recursos Computacionales

Nombre de recurso	Tipo de recurso	Especificaciones
Lenovo thinkpad T14	Laptop	Procesador 4 núcleos, 8 GB RAM, 512 GB SSD
Lenovo thinkpad T14	Laptop	Procesador 4 núcleos, 8 GB RAM, 512 GB SSD
Lenovo thinkpad T14	Laptop	Procesador 4 núcleos, 8 GB RAM, 512 GB SSD

Nombre de recurso	Precio	Técnica	Uso
Cypress	\$0 Open Source	API de automatización, Monkey Testing, GUI Ripping	Pruebas E2E, Pruebas exploratorias
Playwright	\$0 Open Source	API de automatización, Monkey Testing, GUI Ripping	Pruebas E2E, Pruebas exploratorias
Kraken	\$0 Open Source	API de automatización	Pruebas E2E
BackstopJS	\$0 Open Source	API de automatización	Regresión Visual
Mockaroo	\$0 Open Source	API de automatización	Generación de datos A priori y pseudoaleatorio
Faker	\$0 Open Source	API de automatización	Generación de datos aleatorios

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

En esta estrategia de implementación de pruebas sobre la aplicación de GHOST ya se cuenta con el personal y el tiempo necesario, es por esto que no es necesario realizar contratos de servicios y/o recursos tercerizados para la correcta ejecución de la planificación de la estrategia de pruebas.

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

A continuación definimos las Técnicas, Niveles y Tipos de pruebas que se van a utilizar en la estrategia

Nivel	Tipo	Técnica	Objetivo
Sistema	Funcionales – Caja negra-positivas y negativas	Manuales	#1
Sistema	Funcionales – Caja negra-positivas	Monkey testing	#2
Sistema	Funcionales – Caja Negra – Positivas y Negativas	GUI Ripping	#2
Sistema	Funcionales– Caja Negra – Positivas y Negativas	E2E API de automatización	#2, #3
Sistema	Funcionales – Caja Negra – Positivas y Negativas	Validación de datos – API de automatización	#2, #3, #5
Aceptación	No Funcionales – Caja Negra	VRT API de automatización	#2, #3, #4

## 2.5. Distribución de Esfuerzo

De acuerdo con los recursos disponibles de tiempo e ingenieros a continuación se define la distribución de esfuerzo y actividades para cada una de las semanas del proyecto

- Semana 1

Para el inicio de la estrategia se planea realizar una exploración manual al aplicativo GHOST donde se pueda identificar las funcionalidades principales, al tener esto identificado se puede dar prioridades a lo que en realidad se necesita probar y a lo que le puede dar más valor al usuario final.

Con el fin de que esta tarea genere valor a la estrategia de pruebas es fundamental realizar una documentación completa sobre los hallazgos tanto de funcionalidades como de errores presentes en la aplicación.

Esta labor la hace cada uno de los ingenieros con el fin de tener varias perspectivas de lo revisado y consolidar las funciones encontradas y sus errores encontrados en una revisión a alto nivel.

Semana 1 - Pruebas exploratorias			
Actividad	Jose Garay	Juan Becerra	Alexander Mora
Instalación del aplicativo	1	1	1
Revisión manual del aplicativo	3	3	3
Definición del total funcionalidades	1	1	1
Documentación de revisión	2	2	2
Reporte de incidencias	1	1	1

Total esfuerzo	8	8	8
----------------	---	---	---

- Semana 2

En la semana 2 se planifica ejecutar pruebas aleatorias, donde las acciones a ejecutar sean eventos dados por un monkey test, para hacer esto es necesario ejecutarlo por un tiempo considerable.

Es necesario ejecutar las pruebas en distintos navegadores para poder tener un mayor espectro en donde encontrar los posibles errores que estén presentes en la aplicación GHOST, es por esto que en la distribución de las tareas se asigna a uno de los ingenieros ejecutar las pruebas durante un largo periodo de tiempo sobre el navegador chrome, otro ingeniero ejecuta las mismas pruebas sobre firefox y el tercer ingeniero va a ejecutar sobre los dos navegadores mencionados pero con un tiempo más corto, al final se hará un reporte sobre lo encontrado donde incluirá un reporte de los issues encontrados en la aplicación.

Semana 2 - Monkey			
Actividad	Jose Garay	Juan Becerra	Alexander Mora
Priorización de funcionalidades	1	1	1
Configuración del monkey test	1	1	1
ejecución del monkey en chrome	4	0	2
ejecución del monkey en firefox	0	4	2
Reporte de pruebas	1	1	1
Reporte de incidencias	1	1	1
Total esfuerzo	8	8	8

- Semana 3

Esta semana se destinará para la ejecución de pruebas exploratorias automatizadas con generación y ejecución de eventos aleatorios utilizando una herramienta llamada Playwright en su versión 1.14.1 para la ejecución de las pruebas en los navegadores Chromium, Firefox y Gnome Web (Webkit).

Se iniciará la semana con la definición de los objetivos y hoja de ruta de trabajo, de esto surge la necesidad de configurar la herramienta Playwright para que realice las pruebas de exploración con 1 nivel de profundidad, debido al diseño de interfaz del administrador de Ghost la cual cuenta un dashboard como nodo principal y varias secciones como subnodos. Luego se procederá a configuración del ambiente de pruebas donde los tres ingenieros participaran y después se realizará la codificación de los scripts para Playwright y finalmente se llevará a cabo la ejecución de las pruebas y el análisis y reporte de resultados.

Semana 3 - Ripper			
Actividad	Jose Garay	Juan Becerra	Alexander Mora
Definir hoja de ruta y escenarios de pruebas	1	1	1
Configurar herramientas y ambiente de pruebas	1	1	1
Desarrollo de script	2	2	2
Ejecución de pruebas	3	3	3
Análisis y reporte de resultados	1	1	1
Total esfuerzo	8	8	8



- Semana 4

Utilizando el API de automatización Cypress se planifica ejecutar las pruebas en la semana 4, las cuales con de extremo a extremo (E2E), para esto es necesario hacer una definición inicial de los escenarios de cada una de las funcionalidades existentes que se requieren probar.

Al definir los escenarios es importante conocer como es la maquetación de las funcionalidades, se procede a codificar las pruebas en Cypress, abordando diferentes caminos en los que se puede usar la aplicación de esta manera aumenta las probabilidades de encontrar issues que puedan estar presentes.

Para dar un mejor manejo a la creación de pruebas se usan patrones de implementación como PageObject y/o GivenThenWhen para dar un lineamiento en la construcción de las pruebas y su ejecución se haga de una forma eficiente respecto a lo necesario a probar.

Semana 4 - E2E Cypress			
Actividad	Jose Garay	Juan Becerra	Alexander Mora
Definición de escenarios de acuerdo con funcionalidades existentes	1	1	1
Codificación de escenarios	4	4	4
Ejecución de scripts en navegador Chrome	1	1	1
Ejecución de scripts en navegador Firefox	1	1	1
Reporte de incidencias	1	1	1
Total esfuerzo	8	8	8

- Semana 5

En la semana 5 se abordan las pruebas E2E con la API de automatización Kraken, siguiendo un proceso que inicia con la definición de los escenarios que se van a codificar, continúa con la codificación de los escenarios en Kraken, siguiendo buenas prácticas como el uso de la metodología Given When Then para asegurar la calidad, mantenibilidad y optimizar el tiempo de codificación.

Cada uno de los 3 ingenieros tendrá asignados unas funcionalidades del aplicativo de las cuáles será responsable de definir los escenarios, construirlos y reportar los hallazgos encontrados.

Posteriormente se destina un tiempo para la ejecución de los scripts y la solución de defectos de código para finalizar con el reporte de las incidencias encontradas en el aplicativo durante la ejecución de las pruebas.

Semana 5 - E2E Kraken			
Actividad	Jose Garay	Juan Becerra	Alexander Mora
Definición de escenarios de acuerdo con funcionalidades existentes	1	1	1
Codificación de escenarios	4	4	4
Ejecución de scripts en navegador Chrome	1	1	1
Ejecución de scripts en navegador Firefox	1	1	1
Reporte de incidencias	1	1	1
Total esfuerzo	8	8	8



- Semana 6

Esta semana se destinará a la ejecución de pruebas de regresión visual con el objetivo de comparar dos versiones diferentes de la aplicación Ghost, la versión 3.41.1 y la versión 4.44.0. Se hizo uso de la librería de Javascript llamada BackstopJS en su versión 6.1.4.

Se iniciará con la definición de los objetivos donde se establecieron las dos versiones de Ghost que se deseaba comparar y también se definió un mínimo de 6% como umbral de diferencia en la comparación de las imágenes de cada escenario. Para ejecutar las pruebas se configuraron los escenarios y las herramientas de Kraken para la v3.41.1 y Cypress para la v4.44.0, usadas en las semanas 4 y 5 respectivamente, para tomar screenshot de cada escenario y luego poder compararlos con BackstopJS.

Como resultado del análisis de un total de 97 escenarios se obtuvo que solo 22 escenarios estuvieron bajo el 6% de límite diferencia establecido

Semana 6 - Regresión visual			
Actividad	Jose Garay	Juan Becerra	Alexander Mora
Definición de objetivos para la ejecución de pruebas VRT	1	1	1
Configuración de ambiente de pruebas y scripts	3	3	3
Ejecución de pruebas	3	3	3
Análisis y reporte de resultados	1	1	1
Total esfuerzo	8	8	8

- Semana 7

En la semana 7 se aborda una nueva estrategia de pruebas que complementa las pruebas E2E que ya se tienen implementadas, esta nueva estrategia consiste en utilizar 2 métodos distintos de generación de datos, como lo son A Priori y Pseudo Aleatorio.

Para ambos métodos de generación de datos, se utilizará la herramienta Mockaroo la cual permite definir una entidad con los atributos y a su vez definir las características deseadas para cada atributo, y luego generar múltiples copias de esa entidad para su consulta de manera estática a partir de un JSON o dinámica a través de un endpoint que es llamado desde la API de automatización.

La semana iniciará con la definición de los escenarios en donde se implementarán las estrategias de generación de datos, posteriormente se llevará a cabo la implementación de las estrategias y la ejecución para verificar y solucionar cualquier problema de código que se presente, finalmente del ejercicio se reportarán todas las incidencias que se presenten durante la ejecución de las pruebas.

Semana 7 - Validación de datos a priori/ pseudo aleatorio			
Actividad	Jose Garay	Juan Becerra	Alexander Mora
Definición de escenarios de acuerdo con funcionalidades existentes	1	1	1
Codificación de escenarios de datos a priori	2	2	2
Codificación de escenarios de datos pseudo aleatorio	2	2	2
Ejecución de escenarios de validación de datos a priori	1	1	1

Ejecución de escenarios de validación de datos pseudoaleatorio	1	1	1
Reporte de Incidencias	1	1	1
Total Esfuerzo	8	8	8

- Semana 8

Para la última semana de la planificación de la estrategia de pruebas sobre la aplicación GHOST se implementa la validación de datos aleatorios, para lo cual es necesario hacer uso de la librería Faker en su versión 7.6.0 que es una librería que provee muchos datos de una forma aleatoria, de distinto tipos, sintaxis y estructura por lo cual nos es muy útil para generar un set de datos y probar la robustez de la aplicación.

El proceso consta de definir los escenarios, que pueden estar apoyados en las pruebas E2E que se trabajaron en las semanas previas y realizar ajustes o adiciones si se requiere para integrar los datos aleatorios en el momento de la codificación. Estas pruebas van a ser ejecutadas en Chrome y Firefox, que a su vez genera datos distintos en cada ejecución, lo cual es bien útil pues aumenta las posibilidades de encontrar issues en el aplicativo.

Se van a ejecutar en el API de automatización Cypress, y esta herramienta nos ayuda con un reporte de los resultados de las pruebas ejecutadas, adicionalmente se planifica el construir un reporte con los issues encontrados en la ejecución de pruebas con datos aleatorios.

Semana 8 - Validación de datos aleatorios			
Actividad	Jose Garay	Juan Becerra	Alexander Mora
Definición de escenarios de acuerdo con funcionalidades existentes	1	1	1
Codificación de escenarios de datos con datos aleatorios	3	3	3
Ejecución de escenarios de validación de datos aleatorios en chrome	1	1	1
Ejecución de escenarios de validación de datos aleatorios en firefox	1	1	1
Reporte de pruebas	1	1	1
Reporte de incidencias	1	1	1
Total esfuerzo	8	8	8