

Estrategia de Pruebas

1. Aplicación Bajo Pruebas

1.1. Nombre Aplicación: Ghost

1.2. Versión: 5.96.0

1.3. Descripción:

Ghost es una plataforma de publicación en línea centrada principalmente en la creación de blogs y sitios de contenido. Es un software de código abierto que ha ganado popularidad por su simplicidad y rapidez. La ventaja principal que ofrece la plataforma se halla en la publicación gratuita o muy económica de newsletters con gestión de suscripciones.

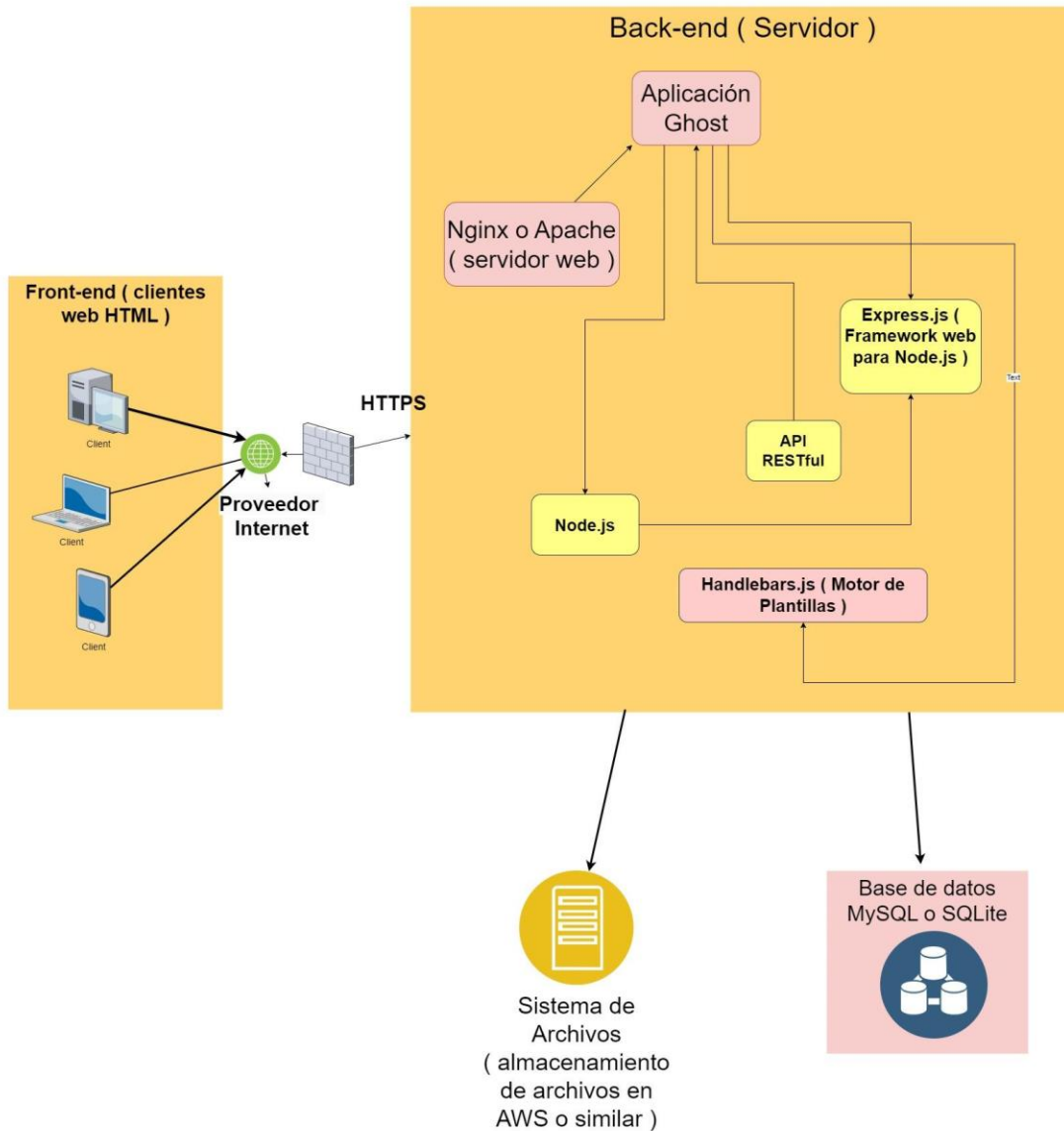
1.4. Funcionalidades Core:

Se ha identificado que, con base en los recursos disponibles, el alcance de pruebas del producto Ghost en esta estrategia planteada incluye los siguientes aspectos funcionales:

ID	NOMBRE	DESCRIPCIÓN
FU1	Administración de Posts	Permite crear, editar, eliminar y publicar posts en el blog
FU2	Administrador de Miembros - suscriptores	Permite la administración de los visitantes del sitio/blog que deseen recibir notificaciones de nuevas publicaciones
FU3	Administración de Pages	Permite crear nuevas páginas del sitio que serán accedidas de forma directa o utilizando un vínculo en la página principal
FU4	Administración de Tags	Cuenta con la capacidad de crear y asignar tags, con el objetivo de que se pueda organizar y clasificar el contenido publicado
FU5	Configuraciones generales, login y seguridad.	Permite modificar características generales del sitio como lo es el título, el tema de diseño, las conexiones a redes sociales. Adicionalmente se incluyen en este sección temas de seguridad y validación de login.

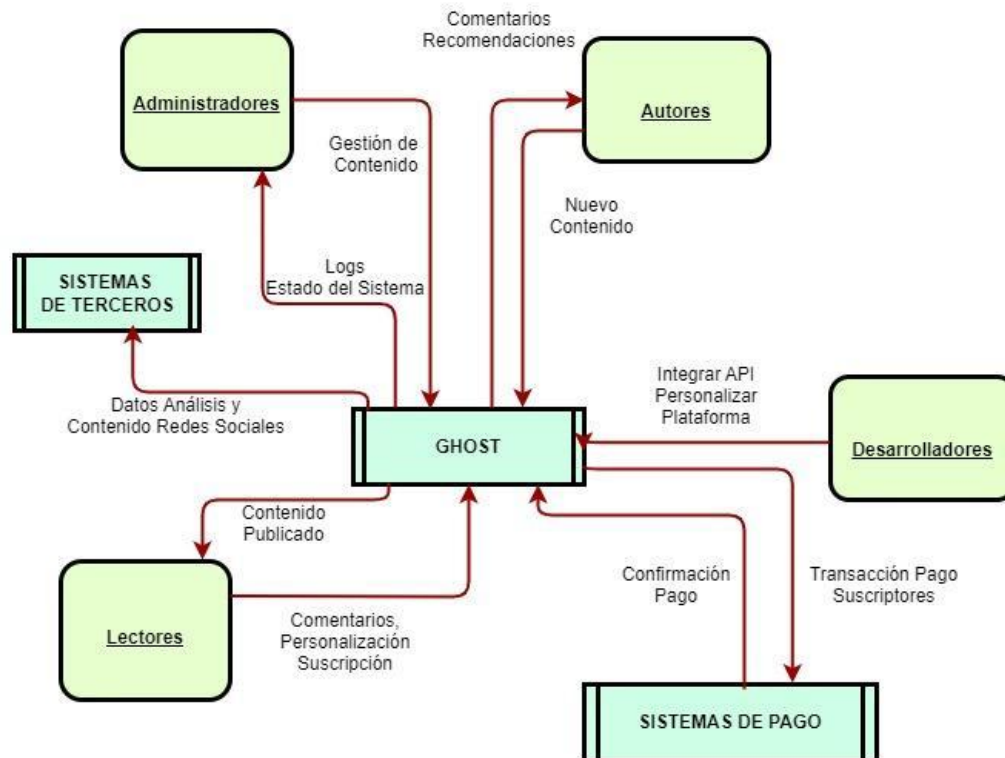
1.5. Diagrama de Arquitectura:

El Diagrama de Arquitectura del sistema fue construido a través de la herramienta Draw.io y está disponible en el archivo “Tarea8 – Modelo de Arquitectura.jpg” anexo a esta entrega. Se incorpora a continuación su imagen, pero se puede visualizar en más detalle abriendo el mencionado archivo:



1.6. Diagrama de Contexto:

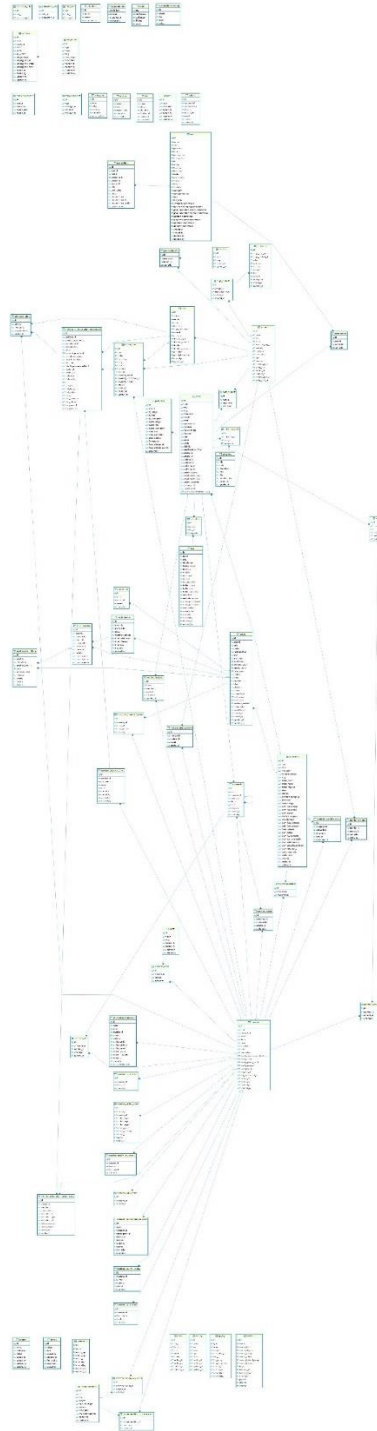
El Diagrama de Contexto del sistema fue construido a través de la herramienta Draw.io y está disponible en el archivo "Tarea8 – Modelo de Contexto.jpg" anexo a esta entrega. Se incorpora a continuación su imagen, pero se puede visualizar en más detalle abriendo el mencionado archivo:



1.7. Modelo de Datos:

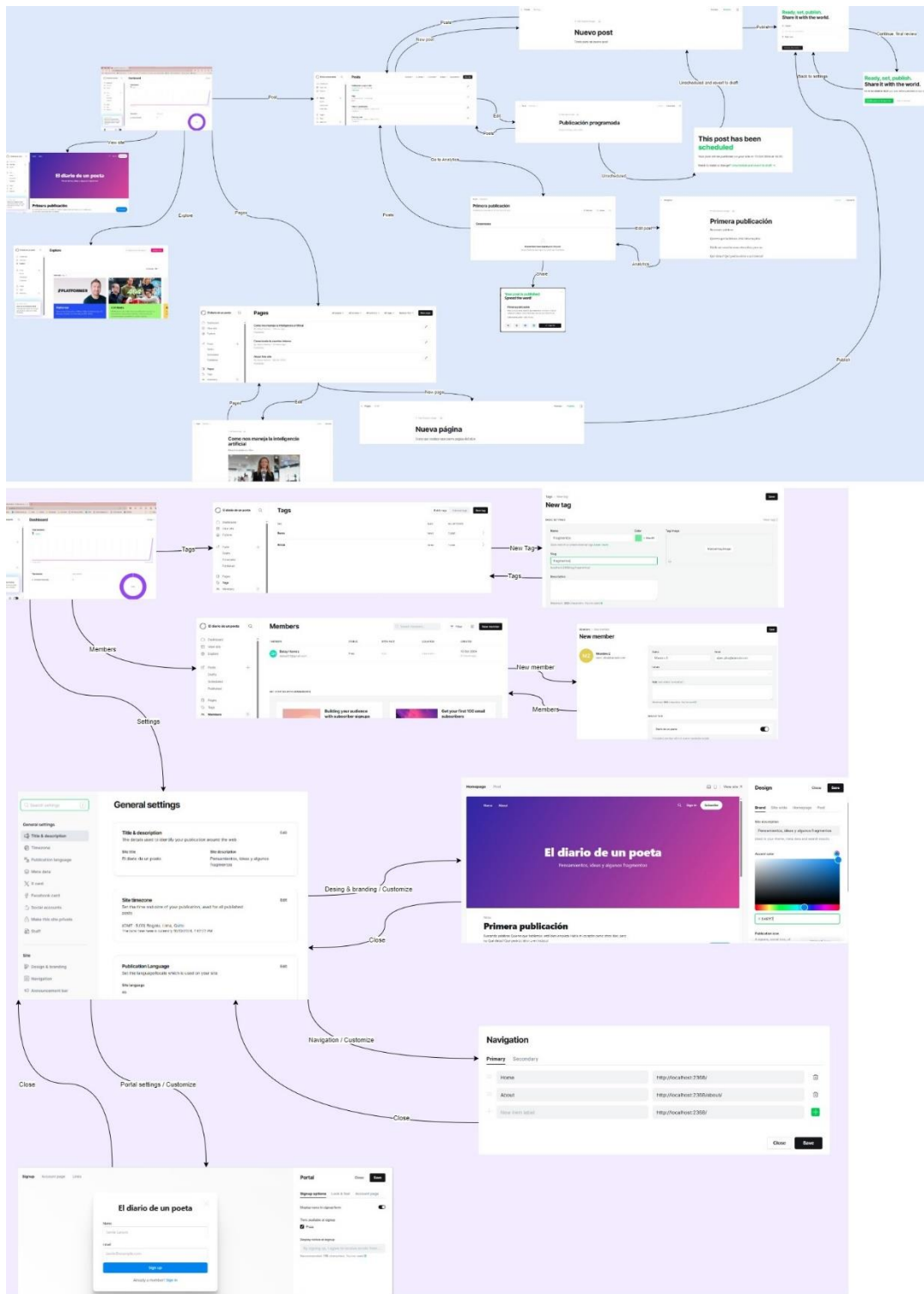
1.8. Modelo de Datos:

El Modelo de dominio del sistema fue construido a través de la herramienta DBeaver con base en el archivo ghost-local.db y está disponible en el archivo “Tarea8 – Modelo de Datos.jpg” anexo a esta entrega. Se incorpora a continuación su imagen, pero se puede visualizar en más detalle abriendo el mencionado archivo:



1.9. Modelo de GUI:

El Modelo GUI del sistema fue construido a través de la herramienta Draw.io y está disponible en el archivo “Tarea8 – Modelo GUI.jpg” anexo a esta entrega. Se incorpora a continuación su imagen, pero se puede visualizar en más detalle abriendo el mencionado archivo:



2. Contexto de la estrategia de pruebas

2.1. Objetivos:

Los objetivos principales de la estrategia de pruebas son:

- Objetivo 1: Detectar la mayor cantidad de errores en la funcionalidad del aplicativo Ghost 5.96.0 con criticidad alta y que afecten sensiblemente la operación.
- Objetivo 2: Validar el 90% de la funcionalidad con criticidad media
- Objetivo 3: Certificar las funcionalidades de los módulos identificados como “Core” del producto.
- Objetivo 4: Identificar escenarios críticos con las pruebas exploratorias para enfocar los esfuerzos de automatización sobre dichos escenarios.
- Objetivo 5: Verificar la correcta ejecución de las principales funcionalidades mediante el uso de pruebas de extremo a extremo.
- Objetivo 6: Identificar cambios en la interfaz gráfica que puedan ser desfavorables para la ejecución normal de la aplicación.
- Objetivo 7: Explorar manualmente algunas funcionalidades que no pueden automatizarse.

2.2. Duración de la iteración de pruebas:

De acuerdo a lo planteado disponemos de 8 semanas para implementar y desarrollar la estrategia de pruebas propuesta y entregar resultados consolidados. Durante estas semanas contamos con cuatro probadores (testers) que se detallan en la siguiente sección.

2.3. Presupuesto de pruebas:

2.3.1. Recursos Humanos

El presupuesto en recursos humanos de pruebas para esta nueva estrategia son cuatro testers senior, durante 8 semanas, en una dedicación de 8 horas por persona semanalmente. Esto quiere decir que las cuatro personas trabajarán en conjunto 32 horas semanales, y en conjunto 256 horas a lo largo de las ocho semanas. Se tiene previsto que los esfuerzos sean

equivalentes y que las cuatro personas ocupen sus ocho horas semanales disponibles en las tareas asignadas en la estrategia.

2.3.2. Recursos Computacionales

Para la presente estrategia se cuenta con las siguientes máquinas:

	Sistema Operativo	Tipo de Procesador	Memoria RAM	Espacio en Disco duro (Disponible)
PC 1	Windows 10 Home	Intel Core i5	8 GB	500 GB
PC 2	Windows 10 Home	Intel Core i5	8 GB	500 GB
PC 3	Windows 10 Home	Intel Core i5	8 GB	500 GB
PC 4	Macbook Pro M1	Apple M1	12 GB	500 GB

Adicionalmente, se requiere el siguiente software instalado en cada máquina o permisos para accederlo vía web:

- Ghost 5.96.0
- Node JS 20.18.0
- SQL Lite 3 (Base de datos que hace parte de la instalación de Ghost)
- Github – disponible en web - 2.47.1.
- Cypress 13.15.2
- Npm 10.8.2
- Faker 9.2.0
- Mockaroo – Ultima versión disponible en línea - internet.
- Visual Studio Code 1.95 o Cualquier Editor para edición de textos
- Resemble JS – 5.0.0

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

Se asume que la Empresa cuenta con la disponibilidad de los cuatro probadores senior y que no dispone de momento de recursos económicas para contratar adicionales. Por otra parte, en esta etapa no se cuenta con recursos de servidor en nube, razón por la cual la versión de la ABP se probará localmente en los equipos de cómputo de los testers o en algún servir que esté disponible en la plataforma de la Empresa.

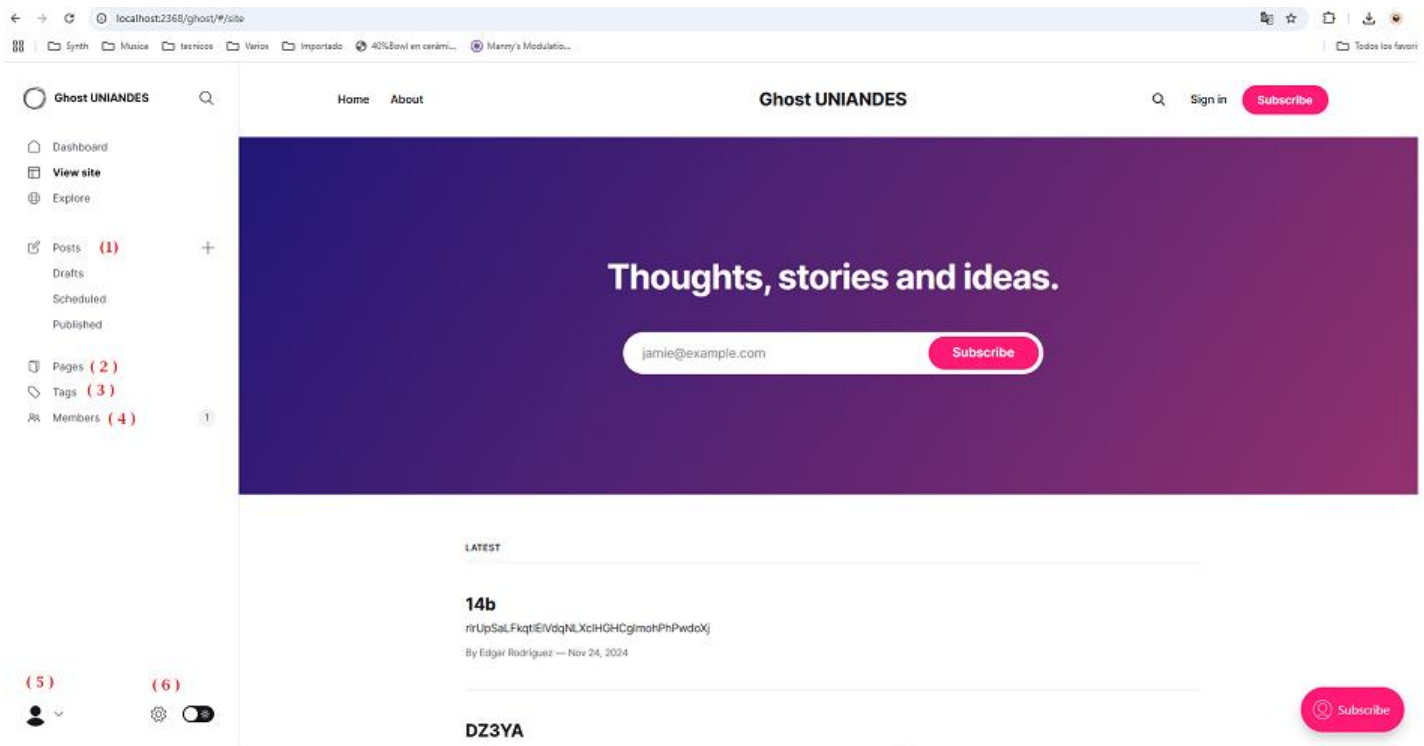
2.4. TNT (Técnicas, Niveles y Tipos) de pruebas:

Nivel	Tipo	Técnica	Objetivo
Pruebas de sistema	Pruebas funcionales Pruebas de caja negra Pruebas positivas y negativas	Pruebas manuales / exploratorias	Objetivo 7
Pruebas de sistema	Pruebas funcionales. Pruebas de caja negra. Pruebas positivas	Pruebas automatizadas E2E con API de Cypress	Objetivos 2, 3 y 5
Pruebas de integración	Pruebas funcionales Pruebas de caja negra Pruebas positivas y negativas	Pruebas automatizadas de regresión visual con ResembleJS	Objetivo 6
Pruebas de integración	Pruebas funcionales. Pruebas de caja negra. Pruebas positivas y negativas	Pruebas automatizadas de validación de datos con Cypress y Faker	Objetivos 1, 2, 3, 4

2.5. Distribución de Esfuerzo y otras Consideraciones en la Estrategia

Luego de un cuidadoso análisis de las funcionalidades ofrecidas por el ECM Ghost y con base en los conceptos y técnicas aprendidos en las últimas semanas, el equipo ha identificado que más del 90% de la funcionalidad Core del sistema se encuentra ubicada en seis (6) opciones principales que se muestran en la siguiente pantalla. Estas funcionalidades son:

- (1) Funcionalidad Posts
- (2) Funcionalidad Pages
- (3) Funcionalidad Members
- (4) Funcionalidad Tags
- (5) Administración Profiles
- (6) Funcionalidad General Settings + Opciones de Login y Seguridad



Respecto a las funcionalidades (5) y (6) – en rojo - se detectó en ejercicios previos una gran dificultad en lograr automatización de dichas funciones en razón a que la identificación de los controles gráficos no es trivial. En la mayoría de los casos, los nombres de los campos, etiquetas, cajas de texto, botones, listas desplegables, etc. se generan con nombres dinámicos cambiantes (exploración HTML y DOM de la página), lo que impide adelantar un proceso consistente de automatización.

Con base en lo anterior, la estrategia propuesta establece que asignaremos **uno** de los cuatro recursos (Tester 4) para trabajar específicamente y durante cinco semanas en pruebas manuales sobre estas dos funcionalidades (5) y (6) que incluye temas de Administración, perfil de usuario, incorporación de archivos anexos (fotos, banners) y otros. El cálculo de las cinco semanas para el Tester4 en pruebas manuales no es caprichoso, y corresponde a cuarenta horas que sobre el total de 256 horas de todo el equipo equivalente al 15.6%. Esto permite que, siguiendo los conceptos básicos de la pirámide de automatización, se asigne un porcentaje menor a las pruebas manuales y se invierta el resto de las horas en procesos de automatización de pruebas con ayuda de los otros testers.

Otra decisión que se tomó para estructurar la distribución de esfuerzo es que no se utilizarán técnicas de navegación por Monkeys y Rippers. Lo anterior porque el equipo ha podido constatar que este tipo de pruebas no son del todo efectivas, impiden un enfoque real de las funcionalidades y módulos, no contamos con recursos de máquina adicionales, se requieren muchas horas de procesamiento, en algunos casos se repiten innecesariamente rutas de pruebas ya exploradas (caso monkeys) y no existe certeza exacta de los posibles resultados.

Respecto a la asignación de tareas para los Tester senior 1 a 3 hemos decidido **especializarlos** y enfocarlos a cada uno en funcionalidades de Ghost. Como se ve en el cuadro de distribución de esfuerzo más adelante, los Tester 1 a 3 trabajarán durante las primeras cuatro (4) semanas y en paralelo las funcionalidades de Post (Tester 1), Members (Tester 2) y Tags (Tester 3) respectivamente. Para entender mejor la distribución se recomienda tener en cuenta los colores de la tabla de actividades porque indican el enfoque y campo de acción que tendrá cada tester durante las ocho semanas.

Respecto a la funcionalidad de Pages se ha observado que es **idéntica** a la de Posts, tiene las mismas opciones, botones desplegables, etc. Por esta razón, hemos decidido que el Tester 1, una vez terminadas sus pruebas en Posts el final de la semana 4, adelante a continuación pruebas automatizadas sobre la funcionalidad de Pages durante las semanas 5, 6 y 7 (una semana menos que lo invertido en Posts) (pruebas similares a las de Post de las cuatro primeras semanas). El conocimiento previo en Posts facilitará al Tester 1 configurar las pruebas de Pages.

Las pruebas de Tester 1, 2 y 3 estarán enfocadas en la utilización de Cypress durante las primera mitad de las cinco semanas y a continuación trabajarán en la generación de scripts similares, pero esta vez enfocadas en pruebas Cypress complementadas con estrategias de generación de datos con Faker y Mockaroo.

Se solicitará en cada caso que los testers utilicen patrones como WHEN-GIVE-THEN y pageObjects en todas implementaciones que desarrollen.

Se escogió Cypress sobre otra opción como Kkaken en vista de que existe muy buena documentación, en general los métodos de la librería son mucho más fáciles de utilizar, es más sencilla la ubicación de los elementos del HTML, la ejecución es sencilla desde la consola y el dashboard es bastante claro e intuitivo para seguimiento de errores y eventos del sistema, y adicionalmente facilita la escogencia de diversos browsers en las pruebas.

Por otra parte, los testers 2 y 3 a partir de la quinta semana estarán disponibles, habrán terminado lo correspondientes a funcionalidades de Members y Tags y se dedicarán durante tres semanas adicionales (5, 6 y 7) a implementar pruebas VRT (Visual Regression Testing) sobre lo ya avanzado por todos en las primeras cuatro semanas. La idea es que estos dos testers utilicen una semana y media cada uno para desarrollar pruebas de reconocimiento visual para Members - Posts (tester 2) y Tags - Pages (Tester 3). Recordemos que el Tester 1 ya tendrá insumos de Pages desde finales de la semana 5 para entregar al Tester 3 e implementar VRT en Pages.

Para las pruebas de regresión visual VRT se propone la utilización de la herramienta ResembleJS. En este caso no se utilizarán versiones diferentes de Ghost, sino versiones alternativas del navegador web. En particular, se propone ejecutar comparaciones sobre Chrome, Edge y Firefox. Se descartan otros navegadores porque no son muy conocidos.

Respecto al tester 4, este recurso estará disponible desde la semana 6 para adelantar por su cuenta pruebas automatizadas diferentes a las manuales que venía desarrollando hasta la semana 5, razón por la cual, hemos decidido asignarle la tarea de desarrollar scripts Cypress y ejecutar pruebas automatizadas sobre aspectos de Seguridad y Login que no habíamos considerado hasta el momento. Este tester es el único que finalmente combinará pruebas manuales y automatizadas.

Durante la semana 8, de cierre de tareas del proyecto, el equipo completo adelantará un proceso de finalización que incluye revisión de la documentación, registro de incidentes que hagan falta y donde sea necesario, retroalimentación con el equipo de trabajo, retrospectivas y pruebas cruzadas cuando se requiera. En este último caso, por ejemplo, se podrían proponer pruebas tipo cruzado como: Un Post se la asignan Tags creados en el otro módulo.

Además de la última semana que será de cierre, definimos que la distribución del equipo en pruebas desde el punto de vista funcional a lo largo de las ocho semanas es la siguiente, como se ve con enfoques puntuales y temáticos para cada tester:

Tester 1	Funcionalidad Posts	Funcionalidad Pages
Tester 2	Funcionalidad Members	Funcionalidades VRT Post y Members
Tester 3	Funcionalidad Tags	Funcionalidad VRT Tags y Pages
Tester 4	Pruebas Manuales en Funcionalidad Administración, configuración, anexos y otros	Automatización Seguridad - Login

Teniendo en cuenta todo lo anotado anteriormente y una perspectiva global de ocho semanas, la distribución de actividades para cada tester semana a semana se observa en el siguiente plan detallado de trabajo:

SEMANA	TESTER 1	TESTER 2	TESTER 3	TESTER 4	TOTAL HORAS
SEMANA 1	Preparación de Scripts, Primeras Ejecuciones Pruebas E2E Posts - Cypress	Preparación de Scripts, Primeras Ejecuciones Pruebas E2E Members - Cypress	Preparación de Scripts, Primeras Ejecuciones Pruebas E2E Tags - Cypress	Pruebas Manuales en funcionalidades de Administración, configuración, anexos y otros	
	8	8	8	8	32
SEMANA 2	Preparación de Scripts, Primeras Ejecuciones Pruebas E2E Posts - Cypress	Preparación de Scripts, Primeras Ejecuciones Pruebas E2E Members - Cypress	Preparación de Scripts, Primeras Ejecuciones Pruebas E2E Tags - Cypress	Pruebas Manuales en funcionalidades de Administración, configuración, anexos y otros	
	8	8	8	8	32
SEMANA 3	Preparación de Scripts, Primeras Ejecuciones Pruebas E2E Posts Cypress + Validación de Datos Faker-Mockaroo	Preparación de Scripts, Primeras Ejecuciones Pruebas E2E Members Cypress + Validación de Datos Faker-Mockaroo	Preparación de Scripts, Primeras Ejecuciones Pruebas E2E Tags Cypress + Validación de Datos Faker-Mockaroo	Pruebas Manuales en funcionalidades de Administración, configuración, anexos y otros	
	8	8	8	8	32
SEMANA 4	Preparación de Scripts, Primeras Ejecuciones Pruebas E2E Posts Cypress + Validación de Datos Faker-Mockaroo	Preparación de Scripts, Primeras Ejecuciones Pruebas E2E Members + Validación de Datos Faker-Mockaroo	Preparación de Scripts, Primeras Ejecuciones Pruebas E2E Tags + Validación de Datos Faker-Mockaroo	Pruebas Manuales en funcionalidades de Administración, configuración, anexos y otros	
	8	8	8	8	32

SEMANA 5	Preparación de Scripts, Primeras Ejecuciones Pruebas E2E Pages Cypress	Desarrollo Scripts Pruebas VRT Members	Desarrollo Scripts Pruebas VRT Tags	Pruebas Manuales en funcionalidades de Administración, configuración, anexos y otros	
	8	8	8	8	32
SEMANA 6	Preparación de Scripts, Primeras Ejecuciones Pruebas E2E Pages Cypress + Faker + Mockaroo	Desarrollo Scripts Pruebas VRT Members y Posts Cypress + Resemble JS	Desarrollo Scripts Pruebas VRT Tags y Pages Cypress + Resemble JS	Preparación de Scripts, Primeras Ejecuciones Seguridad - Login, etc + Validación de Datos - Cypress	
	8	8	8	8	32
SEMANA 7	Preparación de Scripts, Primeras Ejecuciones Pruebas E2E Pages Cypress + Faker + Mockaroo	Desarrollo Scripts Pruebas VRT Post Cypress + Resemble JS	Desarrollo Scripts Pruebas VRT Pages Cypress + Resemble JS	Preparación de Scripts, Primeras Ejecuciones Seguridad - Login, etc + Validación de Datos - Cypress	
	8	8	8	8	32
SEMANA 8	Documentación y Análisis de Resultados - Herramientas que se requieran por demanda	Documentación y Análisis de Resultados - Herramientas que se requieran por demanda	Documentación y Análisis de Resultados - Herramientas que se requieran por demanda	Documentación y Análisis de Resultados - Herramientas que se requieran por demanda	
	8	8	8	8	32

**TOTAL 8
SEMANAS: 256**

La pirámide de automatización que refleja esta propuesta es similar a la siguiente:

