Estrategia de Pruebas

1. Aplicación Bajo Pruebas

1.1. Nombre Aplicación: Ghost
1.2. Versión: 3.42.5
1.3. Descripción:

Ghost es una plataforma de código abierto de publicaciones basada en Node.js. Se encuentra orientada a periodistas y escritores. La vista de administrador está orientada a ofrecer todas las herramientas necesarias para que un usuario individual o una organización cree y administre su contenido, así como realizar distintas modificaciones relacionadas a estilos y layout.

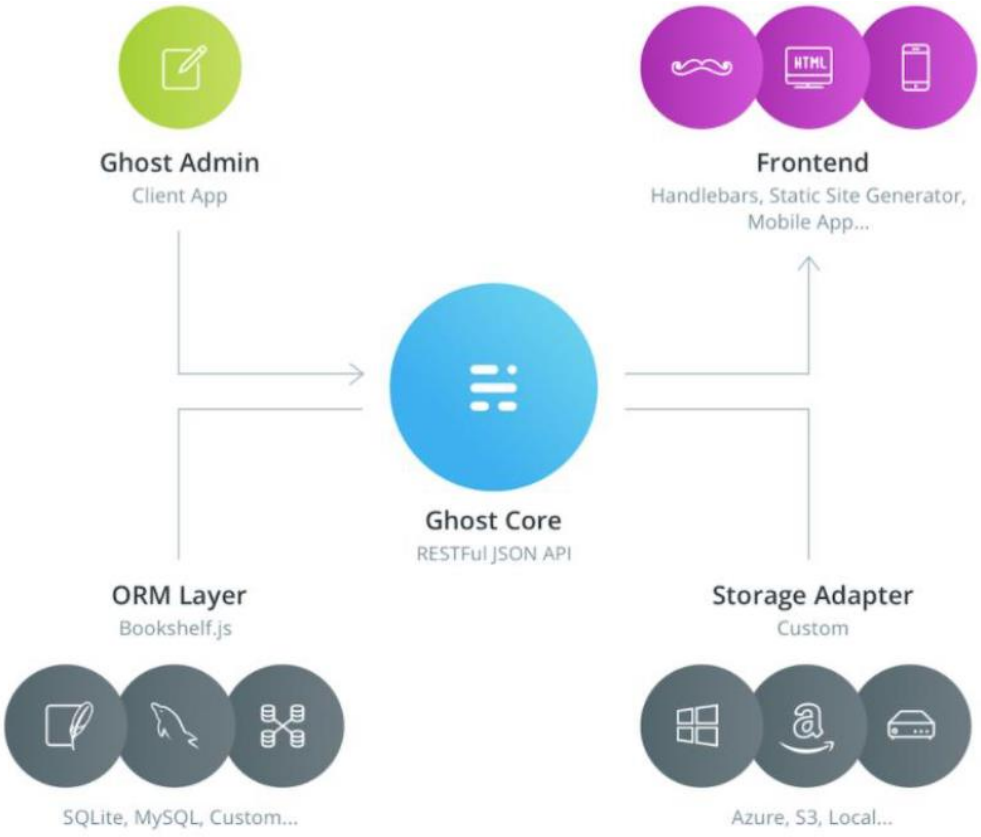
1.4. Funcionalidades Core:

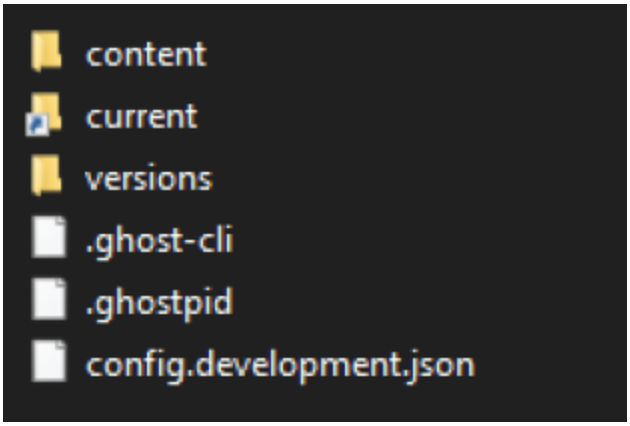
Funcionalidades	Descripción
Seguridad	Autenticación de usuario: Permite realizar el proceso de validación de credenciales para el acceso al menu administración. Cierre de sesión: finaliza la sesión
Publicaciones: post	Crear post Filtrar post Publicar post Modificar post Eliminar post
Publicaciones: páginas	Creación de paginas Publicar Páginas Modificar Páginas Filtrar Páginas por estado (Draft o Published) Filtrar páginas por fecha de creación Eliminar pagina
Etiquetas	Creación de etiquetas Modificar Etiquetas Eliminar Etiquetas
Staff	Creacion de usuarios Staff Modificar perfil de Staff

Funcionalidades	Descripción
Configuración	Editar titulo y descripción de Página Configurar zona horaria Cambiar el icono de publicación Cambiar logo de publicación Cambiar cover de publicación Cambiar meta data para search engines Cambiar el Twitter Card Cambiar el Facebook Card Cambiar cuentas de redes sociales Hacer sitio privado Crear ruta de navegacion Elimnar ruta de navegacion Crear ruta de navegacion secundaria Elimnar ruta de navegacion secundaria Cambiar tema Insertar codigo en header Insertar codigo en footer

1.5. Diagrama de Arquitectura:

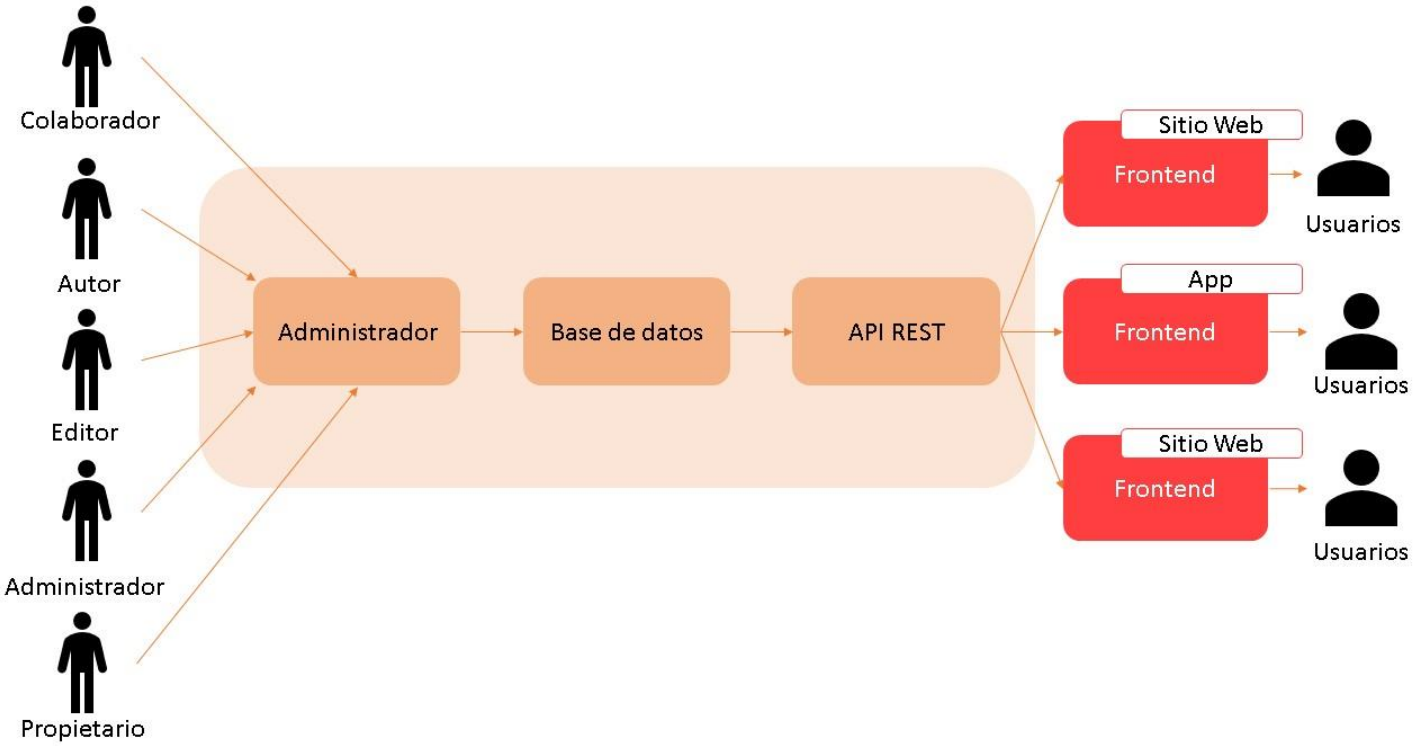
Al consultar la arquitectura de Ghost de su página oficial, se relaciona con una arquitectura *microkernel* el cual se compone de un sistema core y sus módulos plugin. Esta arquitectura permite que sea un producto extensible y flexible, el cual se adecua a una necesidad en particular al permitir agregar o modificar los plugins. Adicionalmente, el sistema organizado por componentes especializados promueve la alta cohesión y bajo acoplamiento de los mismo. Patrones como Facade son aplicados al exponer APIS en su core y plugins.



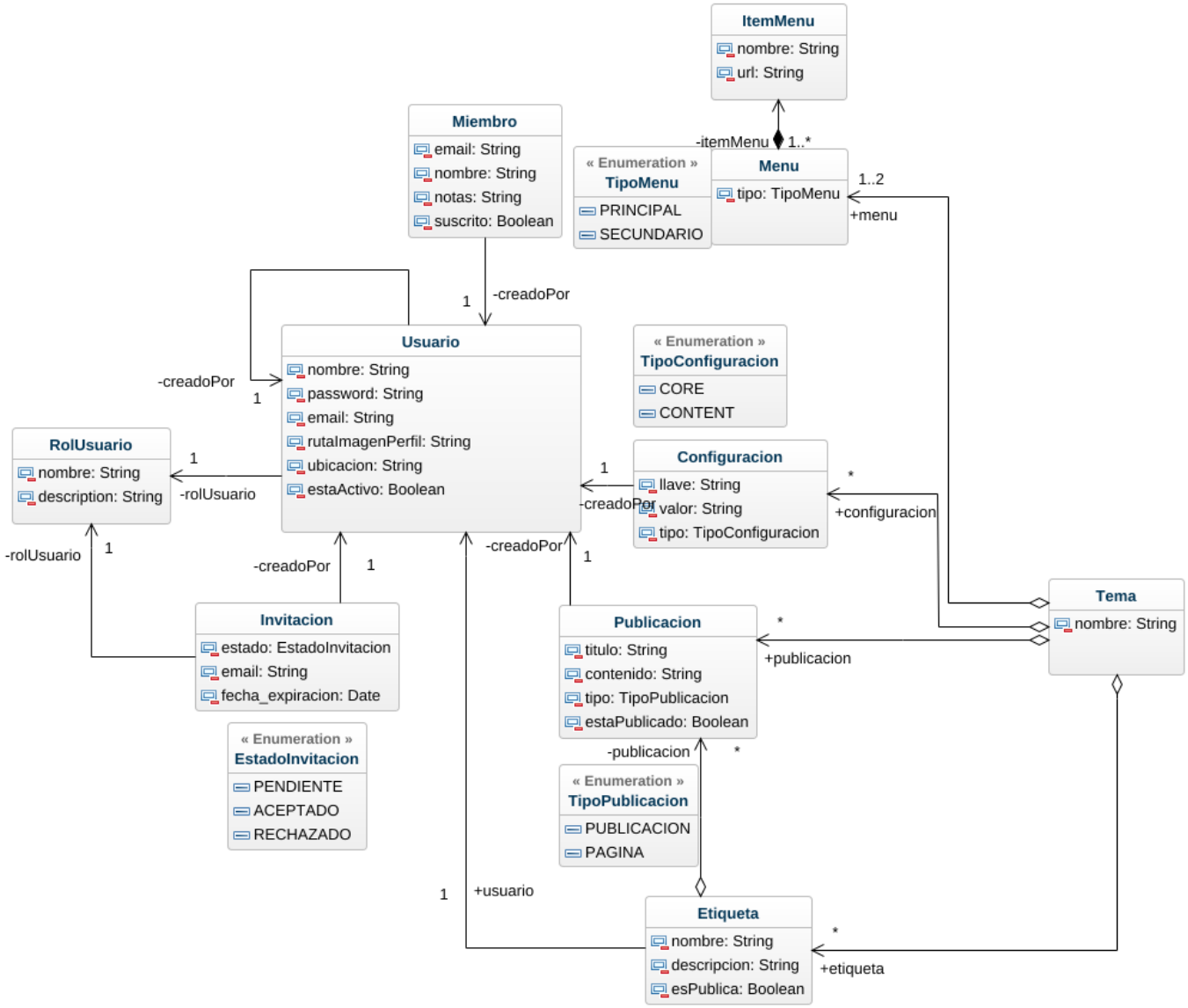


A nivel general se describen los directorios de Ghost:
Core: Contiene los archivos principales que componen Ghost
Content: Contiene los archivos que el usuario puede agregar o cambiar, como temas e imágenes.

1.6. Diagrama de Contexto:



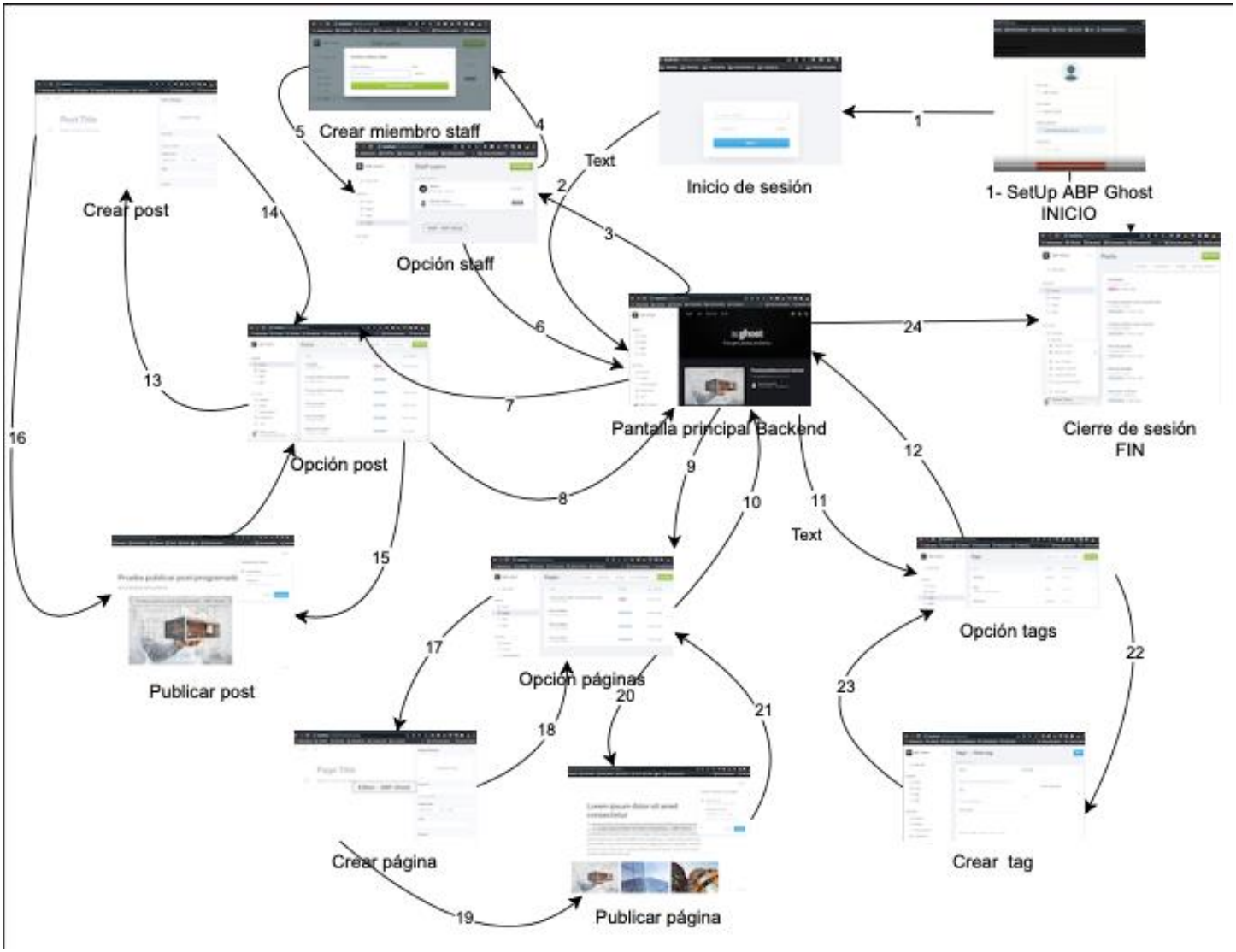
Modelo de Datos:



Entidad	Descripción
Usuario	Representación de la persona que realiza acciones sobre el sistema. Algunas de estas acciones incluyen crear miembros de la comunidad del sitio, crear invitaciones para acceso y contribución en la configuración y contenido del sitio, configurar los roles de usuario y gestionar las publicaciones y etiquetas.
RolUsuario	Perfil de usuarios del sistema que definen la forma en que éste interactúa con el sistema. Estos roles a nivel de usuario final pueden ser: • **Colaboradores**: puede ingresar al sistema y crear publicaciones, pero no publicarlas (hacerlas visibles en el frontend). • **Autores**: pueden crear y publicar nuevas publicaciones y etiquetas. • **Editores**: pueden invitar, gestionar y editar autores y colaboradores. • **Administradores**: tienen todos los privilegios para editar todos los datos y configuraciones. • **Propietario**: un administrador que no puede ser eliminado

Miembro	Persona que ha optado por suscribirse al sitio del frontend y ha confirmado su suscripción mediante la confirmación enviada por el sitio vía correo electrónico.
Invitacion	Solicitud realizada por usuarios editores, administradores o el propietario a una persona a través de correo electrónico para ser un usuario del sitio jugando un rol de usuario específico.
EstadoInvitacion	Estado de solicitud enviada a la persona: • **PENDIENTE**: la persona no ha contestado la solicitud enviada • **ACEPTADO**: la persona ha aceptado la solicitud para ser parte del sitio como usuario. Al aceptar esta solicitud, la persona se dará de alta en el sitio como usuario según el tipo de rol de usuario que se haya relacionado en la invitación. • **RECHAZADO**: la persona ha rechazado la solicitud de participación en el sitio
Menu	Contiene los ítems de navegación del sitio en el frontend que determinan la navegación por los contenidos de éste.
TipoMenu	• **PRINCIPAL**: para navegar por las opciones principales de contenido del frontend • **SECUNDARIO**: menú secundario para navegación por contenidos adicionales al contenido principal.
ItemMenu	Ítem de navegación que enruta a un contenido del sitio, sea una publicación o una página.
Configuracion	Rango de opciones tipo llave-valor que permiten al usuario modificar el comportamiento de las funcionalidades de la aplicación.
TipoConfiguracion	• **CORE**: aplicable al API REST JSON de la aplicación para administrador y frontend. • **CONTENT**: aplicable al sitio del administrador y frontend
Publicacion	Son creadas desde al administrador del sitio de la aplicación y representan en contenido que será visible en el frontend.
TipoPublicacion	• **PUBLICACION**: agrupa contenido que es creado, publicado y actualizado frecuentemente en el tiempo, como blogs o podcasts. • **PAGINA**: contenido independiente y estático que no cambia en el tiempo, como la información de la organización dueña de la página o un formulario de contacto.
Etiqueta	Entidad que se emplea para filtrar y organizar la relación entre las entidades de contenido (publicaciones).
Tema	Capa de diseño visual del sitio visible al público que se construye de manera flexible por desarrolladores y que incluye el contenido configurado en el sitio del administrador para ser visible en el frontend. La configuración involucrada incluye

1.7. Modelo de GUI:



2. Contexto de la estrategia de pruebas

2.1. Objetivos:

El objetivo de este documento es definir la estrategia de pruebas para la aplicación Ghost en su versión 3.42.5. A continuación, se definen los objetivos específicos que se esperan alcanzar:

- Definir una estrategia de pruebas con una duración de ocho (8) semanas con cuatro testers senior que permita validar las siguientes funcionalidades: seguridad(login), post, páginas y etiquetas.
- Definir las técnicas, tipos y niveles de pruebas así como las herramientas tecnológicas a utilizar para el desarrollo de las pruebas.
- Documentar y ejecutar 51 escenarios para realizar pruebas exploratorias manuales para el grupo de funcionalidades: seguridad(login), post, páginas y etiquetas.

- Realizar pruebas de reconocimiento que permitan detectar posibles excepciones o crashes en la aplicación bajo pruebas.
- Documentar y ejecutar 40 escenarios para realizar pruebas de extremo a extremo para el grupo de funcionalidades: (login), post, páginas y etiquetas.
- Documentar y ejecutar 10 escenarios para realizar pruebas de regresión visual VRT para el grupo de funcionalidades: (login), post, páginas y etiquetas.
- Documentar y ejecutar 120 escenarios de validación mediante las estrategias de generación de datos (i) pool de datos a-priori, (ii) pool de datos (pseudo) aleatorio dinámico y (iii) escenario aleatorio.
- Detectar posibles defectos sobre las funcionalidades core con base en los escenarios de prueba.
- Registrar los defectos encontrados en un sistema de registro de incidencias y asignar responsable con prioridad.

2.2. Duración de la iteración de pruebas:

Las pruebas tendrán una duración de 8 semanas y cada tester dedicara 8 horas por semana.

Fecha de inicio: 24 de mayo del 2021. Fecha fin: 18 de julio del 2021.

- Las pruebas de exploratorias tendrán una duración de 7 días, y serán ejecutadas iniciando el 24 de mayo del 2021 y finalizan el 30 de mayo del 2021.
- Las pruebas de reconocimiento tendrán una duración de 7 días, y serán ejecutadas iniciando el 31 de mayo del 2021 y finalizan el 6 de junio del 2021.
- Las pruebas de extremo a extremo (E2E) tendrán una duración de 14 días, y serán ejecutadas iniciando el 7 de junio del 2021 y finalizan el 20 de junio del 2021.
- Las pruebas de regresión visual (VRT) tendrán una duración de 7 días, y serán ejecutadas iniciando el 21 de junio del 2021 y finalizan el 27 de junio del 2021.
- La ejecución de los escenarios de validación de datos de tendrán una duración de 14 días, y serán ejecutadas iniciando el 28 de junio del 2021 y finalizan el 11 de julio del 2021.
- La realización de los análisis y reportes de resultados de la estrategia de pruebas tendrá una duración de 7 días iniciando el 12 de julio de 2021 y finalizando el 18 de julio de 2021.

2.3. Presupuesto de pruebas:

2.3.1. Recursos Humanos

Recurso	Experiencia	Disponibilidad
Tester Senior.	• 3 años de experiencia en automatización de pruebas, con amplio manejo de SOAPUI, JMETER, Fiddler, JUNIT(Selenium Web Driver) Automatización con Jenkins, TFS, Test Studio (Telerik), y de aplicaciones desarrolladas en .NET AJAX (Telerik), Angular (Framework), manejo	8 horas/semana

	avanzado de Github. Conocimientos en Cucumber y ResembleJS.	
Tester Senior.	• 3 años de experiencia en automatización de pruebas, con manejo de JMETER, automatización con Jenkins y AWS. • Conocimientos en Web Services (SOAP y REST), Bases de Datos (MySQL, Oracle o DB2) y conocimientos en SCRUM y Cypress	8 horas/semana
Tester Senior.	• 3 años de experiencia en automatización de pruebas con AWS, Travis y Jenkins. • Desarrollo de web services, certificada en SCRUM Master, Oracle Java Developer Master y Oracle Certified Professional (OCP). Conocimientos en herramientas de generación automática de datos.	8 horas/semana
Tester Senior.	• 3años de experiencia en automatización de pruebas con AWS y Jenkins. • Certificado en SCRUM Master, Conocimientos en Playwright.	8 horas/semana

2.3.2. Recursos Computacionales

Cantidad	Recurso	Especificaciones	Disponibilidad
4	Computadoras personales	• Lenovo intel core i7 decima generación, memoria RAM 16GB, almacenamiento SSD 1TB	Si
4	Línea base de software de ofimática y desarrollo de software	• Office 365 • Visual Studio Code • VirtualBox	Si
4	Software para automatización de pruebas	• Cypress-monkey • RIPuppet • Cypress • Kraken • ResembleJS • BackstopJS • Faker	Si

		• Mockaroo	

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

No se cuentan con recursos económicos para contratar servicios externos.

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

NIVEL	TIPO	TECNICA	Detalle	OBJETIVO
Sistema	Funcionales, caja negra, positiva/negativa	Manual	Realizar pruebas manuales para conocer las funcionalidades del sistema e identificar posibles errores de manera preliminar.	Documentar y ejecutar 20 escenarios para realizar pruebas exploratorias manuales para el grupo de funcionalidades: seguridad(login), post, páginas y etiquetas.
Sistema	No funcionales, caja negra, positiva/negativa	Automatizada	Se utilizará pruebas exploratorias mediante el uso de Monkeys y Rippers para detectar errores mediante interacciones poco usuales.	Realizar pruebas de reconocimiento que permitan detectar posibles excepciones o crashes en la aplicación bajo pruebas.
Aceptación/ Sistema	Funcionales, E2E, positiva/negativa	Automatizada	Se propone hacer uso de Cypress y Kraken para probar las funcionalidades que más se usan dentro del sitio empezando por el componente/flujo de login.	Documentar y ejecutar 40 escenarios para realizar pruebas de extremo a extremo para el grupo de funcionalidades: (login), post, páginas y etiquetas.
Sistema	No funcional, VRT	Automatizada	Se realizarían pruebas con ResembleJS y BackstopJS para validar que los defectos encontrados y reportados en la versión anterior de la aplicación (3.3.0) no se introducen en la versión 3.42.5 y que afecten la GUI.	Documentar y ejecutar 10 escenarios para realizar pruebas de regresión visual VRT para el grupo de funcionalidades: (login), post, páginas y etiquetas.
Integración	Funcional, frontera y robustez, positiva/negativa	Automatizada	Se propone la realización de pruebas de validación de formularios generando datos con Mockaroo y Faker para validar las robustez y límites de los campos.	Documentar y ejecutar 120 escenarios de validación mediante las estrategias de

			Estas pruebas resultan más económicas de realizar en escenarios de prueba independientes que incluirlas dentro de las pruebas E2E y ayudan a localizar defectos que se encuentren en un componente y que no dependan de un flujo completo de funcionalidad.	generación de datos (i) pool de datos a-priori, (ii) pool de datos (pseudo) aleatorio dinámico y (iii) escenario aleatorio.

2.5. Distribución de Esfuerzo

Iteración	Objetivo	Responsable	Entregable	Esfuerzo
1 24/05/2021 - 30/05/2021	Realizar pruebas exploratorias manuales sobre la aplicación bajo pruebas. Identificar posibles defectos, excepciones y crashes. Identificar escenarios de prueba iniciales.	Testers seniors	Inventario pruebas exploratorias. Incidencias reportadas con evidencia.	32 horas
2 31/05/2021 - 06/06/ 2021	Realizar pruebas exploratorias automatizadas de reconocimiento sobre la aplicación bajo pruebas. Identificar posibles defectos, excepciones y crashes. Identificar escenarios de prueba iniciales. Identificar estados adicionales de la aplicación.	Testers seniors	Reporte de pruebas de reconocimiento con RIPuppet y Monkey-cypress.	32 horas
3 07/06/2021 - 20/06/2021	Diseñar pruebas automatizadas E2E Instalar y configurar Cypress y Kraken como preparación de ambientes para la ejecución de las pruebas Implementar scripts de pruebas automatizadas E2E	Testers seniors	Scripts de pruebas automatizadas. Reporte de resultados de ejecución de las pruebas E2E.	64 horas
4 21/06/2021 - 27/06/2021	Incluir dentro de las pruebas E2E desarrolladas en la iteración anterior la captura automática de	Testers seniors	Reporte de diferencias visuales de Ghost para la regresión.	32 horas

	screenshots para cada paso de las pruebas integrando ResembleJS y BackstopJS. Ejecutar los scripts de pruebas E2E modificados sobre las versiones 3.3.0 y 3.42.5 de Ghost para obtener los screenshots.			
5 28/06/2021 - 11/07/2021	Incluir mecanismos de generación de datos en formularios dentro de los scripts de pruebas E2E para generar nuevos escenarios de pruebas que permitan validar valores inválidos y de frontera.	Testers seniors	Scripts de pruebas automatizadas Resultados de pruebas	64 horas
6 12/07/2021 - 18/07/2021	Reportar incidencias encontradas y documentar el proceso de prueba. Socializar los resultados dentro del equipo.	Testers seniors	Documento final de resultados y proceso de pruebas para el cliente. Incidencias reportadas. Repositorio con scripts de automatización de pruebas.	32 horas