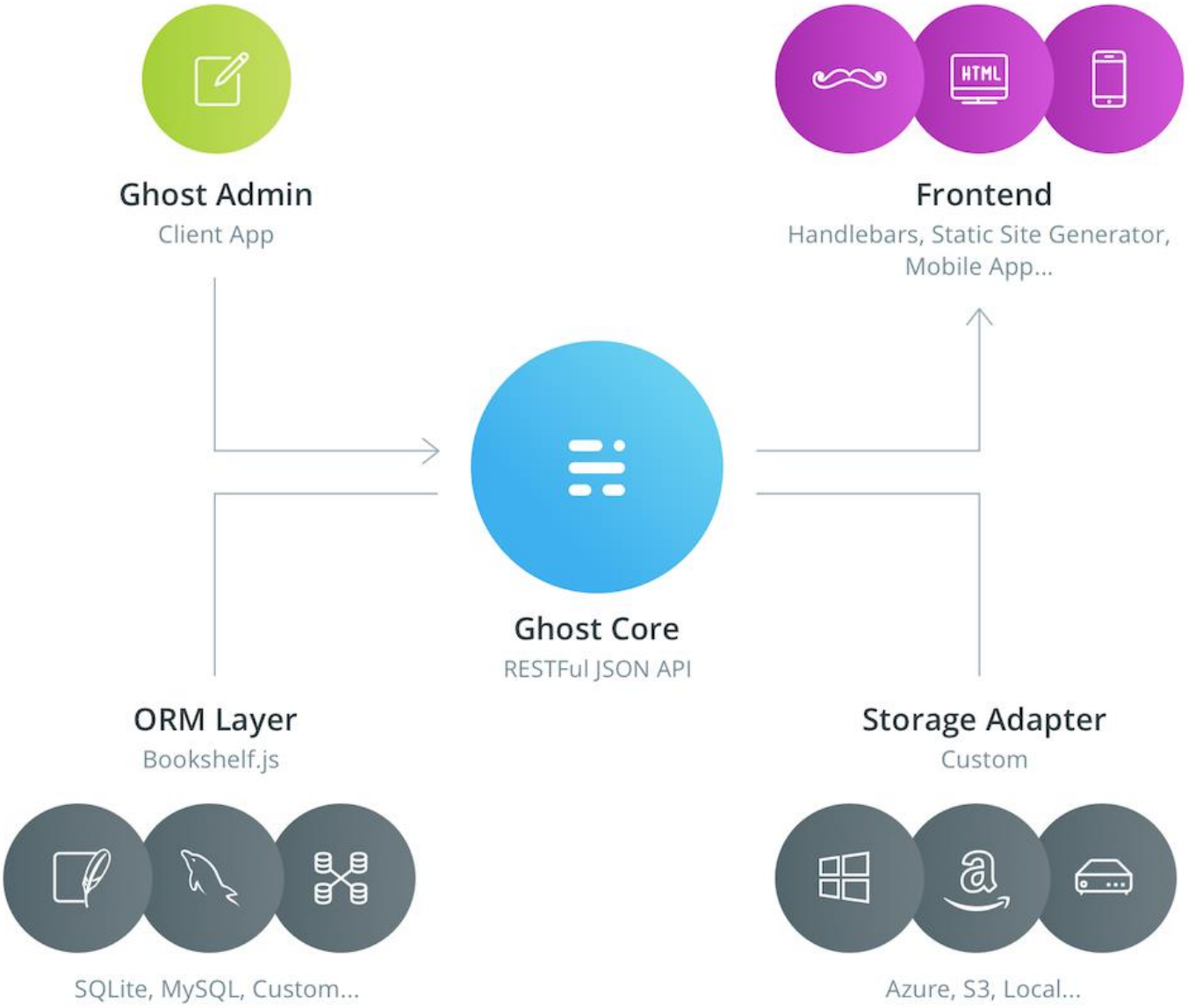
Estrategia de Pruebas

1. Aplicación Bajo Pruebas

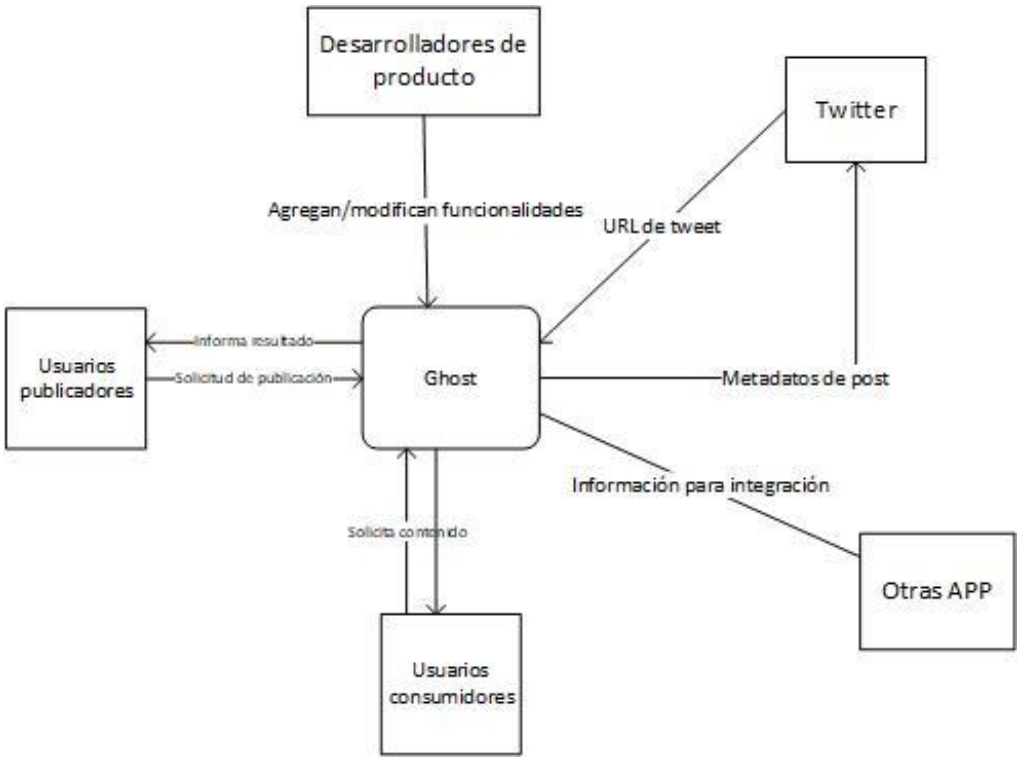
1.1. **Nombre Aplicación:** Ghost
1.2. **Versión:** 3.42.5
1.3. **Descripción**: Ghost es un sistema gestor de contenidos para la creación de publicaciones en línea de código libre.
1.4. **Funcionalidades Core**:

- Agregar post: permite al usuario agregar un post a su catálogo de posts. Entre los elementos de un post están su título, su contenido en diferentes formatos (imágenes, videos), tags asociados y su autor entre otros.
- Editar post: permite al usuario modificar un post cambiando los valores de los elementos mencionados en la funcionalidad 1.
- Publicar post: permite al usuario hacer público su post para que pueda ser consultado por otros usuarios en su blog.
- Revertir publicación de post: permite al usuario devolver la publicación de un post para poder hacer cambios sobre el mismo. Con esto el post ya no puede ser visto por otros usuarios en el blog.
- Borrar post: permite al usuario borrar definitivamente un post que ya no va a necesitar.
- Crear tags: permite al usuario crear tags que pueden asociarse en un futuro a los posts. Entre la información que representa a un tag está su nombre, slug, una descripción, los metadatos, entre otros.
- Editar tags: permite editar tags modificando alguno de los elementos mencionados en la funcionalidad en la funcionalidad anterior.
- Borrar tags: permite al usuario borrar definitivamente un tag que ya no va a usar.
- Editar usuario: permite al usuario editar información de usuario como el nombre, slug, el correo, la contraseña, entre otros.
- Invitar usuarios: permite invitar a usuarios para que participen con alguno de los roles que se manejan en el sistema en la administración o construcción del blog.
- Suspender usuario: permite suspender a un usuario para no pueda iniciar sesión y con ello no pueda crear contenido en el blog.

1.5. Diagrama de Arquitectura:



1.6. Diagrama de Contexto:



1.7. Modelo de Datos:

El modelo de datos se puede ver en el siguiente enlace:

https://github.com/wilson-bg/entrega-final/blob/main/6.%20Soportes/Imagen/ghost-local.png

1.8. Modelo de GUI:

El modelo de datos se puede ver en el siguiente enlace:

https://github.com/wilson-bg/entrega-final/blob/main/6.%20Soportes/Imagen/ModelGUI_Ghost.jpg

2. Contexto de la estrategia de pruebas

2.1. Objetivos:

2.1.1. Comprobar el correcto funcionamiento de las funcionalidades del sistema en diferentes navegadores.
2.1.2. Evaluar la capacidad que tiene el sistema para evitar fallas
2.1.3. Verificar la capacidad del sistema ante un error
2.1.4. Implementar pruebas para probar el correcto funcionamiento del código creado
2.1.5. Encontrar e identificar el mayor número de bug que pueda tener la aplicación
2.1.6. Demostrar que la aplicación cumple con las especificaciones y requerimientos establecidos
2.1.7. Asegurar el correcto funcionamiento de toda la aplicación, frente a un cambio de cualquiera de sus funcionalidades.
2.1.8. Detectar problemas de rendimiento

2.2. Duración de la iteración de pruebas:

La presente estrategia contempla **(4)** cuatro iteraciones de **(2)** dos semanas cada una, continuación, se detallan las actividades a ejecutar en cada iteración.

Iteración 1
En esta iteración se realizarán pruebas de exploración y el equipo podrá crear los escenarios prueba que serán insumos de las iteraciones contempladas. Igualmente se reportarán los hallazgos encontrados

Iteración 2
En esta iteración, desarrollaremos un script bajo la modalidad monkey en la herramienta Cypress con el objetivo que ejecute pruebas aleatorias que permitan identificar escenarios no contemplados en iteración 1. Estas pruebas pueden realizarse en horarios de no oficina para que los equipos puedan ser usados durante la jornada laboral, se generará un reporte de hallazgos del resultado de las pruebas de tipo monkey.

Iteración 3
En esta iteración, se aplicarían 2 tipos de pruebas: pruebas e2e automatizadas y pruebas automatizadas de regresión,
implementación de pruebas automáticas e2e utilizando como herramienta Cypress.
Con el objetivo de validar la versión de Cypress inicialmente entrega al equipo (versión X), la y su comparación (versión X +1), la construcción de pruebas automáticas de regresión utilizando las Cypress y Resembler.JS. se generará un reporte de hallazgos del resultado de las pruebas teniendo en cuenta la comparación de imágenes tomadas durante la iteracion de las versiones de ghost.

Iteración 4
En esta iteración se presumen contar ya con un producto estable y de calidad los esfuerzos se orientarán hacia la ejecución de pruebas de sistema y de aceptación.

Iteración	Actividad	Duración	Resultado
1	• Diseñar casos de prueba • Pruebas manuales (exploratorias)	Semana 1 y 2	• Identificar sus funcionalidades • Escenarios de prueba • Reporte de hallazgos
2	• Pruebas Aleatorias (Monkey)	Semana 3 y 4	• Identificar escenarios que no fueron encontrados en la iteración 1 • Reporte de hallazgos • Resultados inesperados
3	• Pruebas E2E automatizadas (Cypress) • Pruebas automatizadas de regresión (Cypress - Resembler.JS)	Semana 5 y 6	• Identificar escenarios que no fueron encontrados en la iteración 2 • Reporte de hallazgos • Reporte Visual de diferencias entre versiones
4	• Pruebas de Aceptación y Sistema (Cypress)	Semana 7 y 8	• Identificar escenarios que no fueron encontrados en la iteración 3 • Reporte de hallazgos • Reporte de Gestion • Software Estable y de Calidad

2.3. Presupuesto de pruebas:

2.3.1. Recursos Humanos

Se cuenta con tres ingenieros automatizadores con una disponibilidad de 8 horas semanales por persona y con experiencia en diferentes herramientas para la implementación de pruebas automatizadas mediante técnicas como: APIs de automatización, Record & Replay y pruebas de reconocimiento.

2.3.2. Recursos Computacionales

Cantidad	Tipo	Disponibilidad	Características
2	Servidores	100 horas / Maquina	CPU Core i7 32 GB de RAM 1 TB SD Windows Server Ghost Cypress Npm

3	Computador de escritorio	24/7	CPU Core i5 16 GB de RAM 500 GB SSD Windows 10

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

Recurso	Cantidad	Valor unitario estimado/mes	Valor total estimado/mes
Ingeniero de pruebas	3	USD $800	USD $2400
Servidores	2	$0	$0
Computador personal	2	$0	$0
Total:	**6**	**USD $800**	**USD $2400**

No se cuenta con recursos para tercerizar ninguna de las tareas relacionadas con las pruebas del software

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

Nivel	Tipo	Técnica	Objetivo
Sistema	E2E	GUI ripping	2.1.1
Sistema	Negativa	Exploratoria	2.1.2
Sistema	Negativa	Exploratoria	2.1.3
Unidad	Caja blanca	APIs de automatización	2.1.4
Sistema	Caja negra	Monkey Testing	2.1.5
Aceptación	Positivas	Pruebas manuales Alfa	2.1.6
Integración	Funcional	Record & Replay	2.1.7
Sistema	No funcional	Pruebas de stress Pruebas de carga	2.1.8

2.5. Distribución de Esfuerzo

Prueba	Recurso	Duración
GUI ripping, Monkey Testing, Record & Replay, pruebas de stress y de carga	Servidores	20 horas por día / maquina
Pruebas de APIs de automatización	Equipo ingeniero automatizador	3 horas por día
Pruebas exploratorias	Equipo ingeniero automatizador	5 horas por día