

Estrategia de Pruebas No. 1

Fecha:	17 de noviembre de 2024
Autores:	Calletana López Juan Tapia Raúl Ramos Sergio Andrés Gelves
Empresa:	The Software Design Company (TSDC)

Tabla de Contenido

1	Aplicación bajo prueba	2
1.1	Nombre de la Aplicación:	2
1.2	Versión de la aplicación:	2
1.3	Descripción:	2
1.4	Listado de funcionalidades	2
2	Contexto de la estrategia de pruebas	8
2.1	Objetivos:	8
2.2	Duración de la iteración de pruebas:	8
2.3	Presupuesto de pruebas:	9
2.4	TNT (Técnicas, Niveles y Tipos) de pruebas:	10
2.5	Distribución de Esfuerzo:	11
2.6	Justificación uso de herramientas:	12
2.7	Análisis de herramientas:	12

1. Aplicación bajo prueba

1.1 Nombre de la Aplicación:

Ghost

1.2 Versión de la aplicación:

5.96.0

1.3 Descripción:

Ghost es un sistema de gestión de contenidos (CMS) que se usa principalmente para crear blogs y sitios web enfocados en publicaciones. Es conocido por ser fácil de usar y por permitir que las personas publiquen y gestionen contenido de manera sencilla

1.4 Listado de funcionalidades

Inicio

1. Creación de cuenta primera vez: cuando se inicia por primera vez el módulo administrativo se muestra un formulario para crear una cuenta nueva.
2. Iniciar sesión: inicia sesión en el módulo administrativo y muestra el dashboard de la aplicación.

Configuración General.

3. Configurar título y descripción: permite cambiar el título y la descripción del sistema
4. Configurar zona horaria: permite cambiar la zona horaria del sistema
5. Configurar idioma: permite cambiar el idioma principal del sistema
6. Configurar meta data: permite configurar los meta datos para los buscadores web

7. Configurar cuentas sociales: permite modificar los vínculos a las redes sociales Facebook y X

Staff:

8. Invitar persona: permite invitar otras personas para participar en el sistema
9. Modificar persona: permite editar los datos de un colaborador

Personalizar el diseño:

10. Personalizar marca: Permite cambiar descripción, color de botones y de fondo de la página
11. Personalizar ancho del sitio: permite cambiar el tipo de letra y ubicación del logotipo
12. Personalizar Home Page: permite cambiar la estructura de la página principal
13. Personalizar Post: permite activar opciones de personalización en la vista de los posts
14. Personalizar Navegación: Permite configurar los vínculos de navegación del menú principal
15. Modificar perfil: permite cambiar el correo y la contraseña entre otros datos del usuario que inicio sesión.

Post

16. Agregar post: permite crear una nueva publicación en estado borrador
17. Editar post: permite modificar los datos de un post existente
18. Eliminar post: permite eliminar un post publicado
19. Configurar post: permite modificar la configuración de la publicación del post
20. Publicar post: permite publicar el post en la página
21. Despublicar post: permite ocultar un post publicado, el cual no se verá en la página, regresa a estado de borrador.

Páginas

- 22. Nueva página: permite agregar una nueva página en modo borrador.
- 23. Editar página: permite modificar la información de una página existente
- 24. Configurar página; permite modificar los datos como la url, fecha de publicación, tags, tipo de acceso, entre otros.
- 25. Eliminar página: permite eliminar una página existente
- 26. Publica página: permite publicar una página en el sistema
- 27. Despublicar página: permite ocultar una página y volver al estado en borrador.

Elementos web

- 28. Agregar imagen: permite agregar una imagen a una página o post.
- 29. Agregar video: permite agregar un video a una página o post.
- 30. Agregar producto: permite agregar una imagen con un título, botón y estrellas para calificación del producto.
- 31. Agregar galería: permite agregar varias imágenes organizadas, máximo 9.

Tags

- 32. Agregar tag: permite agregar tag para agrupar los posts publicados
- 33. Editar tag: modifica un tag existente.

Suscriptores

- 34. Agregar suscriptor: permite agregar un suscriptor por medio del correo, este es una persona que tiene acceso a contenido exclusivo.
- 35. Editar suscriptor: Editar un suscriptor existente.

1.5 Diagramas de la arquitectura.

Diagrama de Componentes

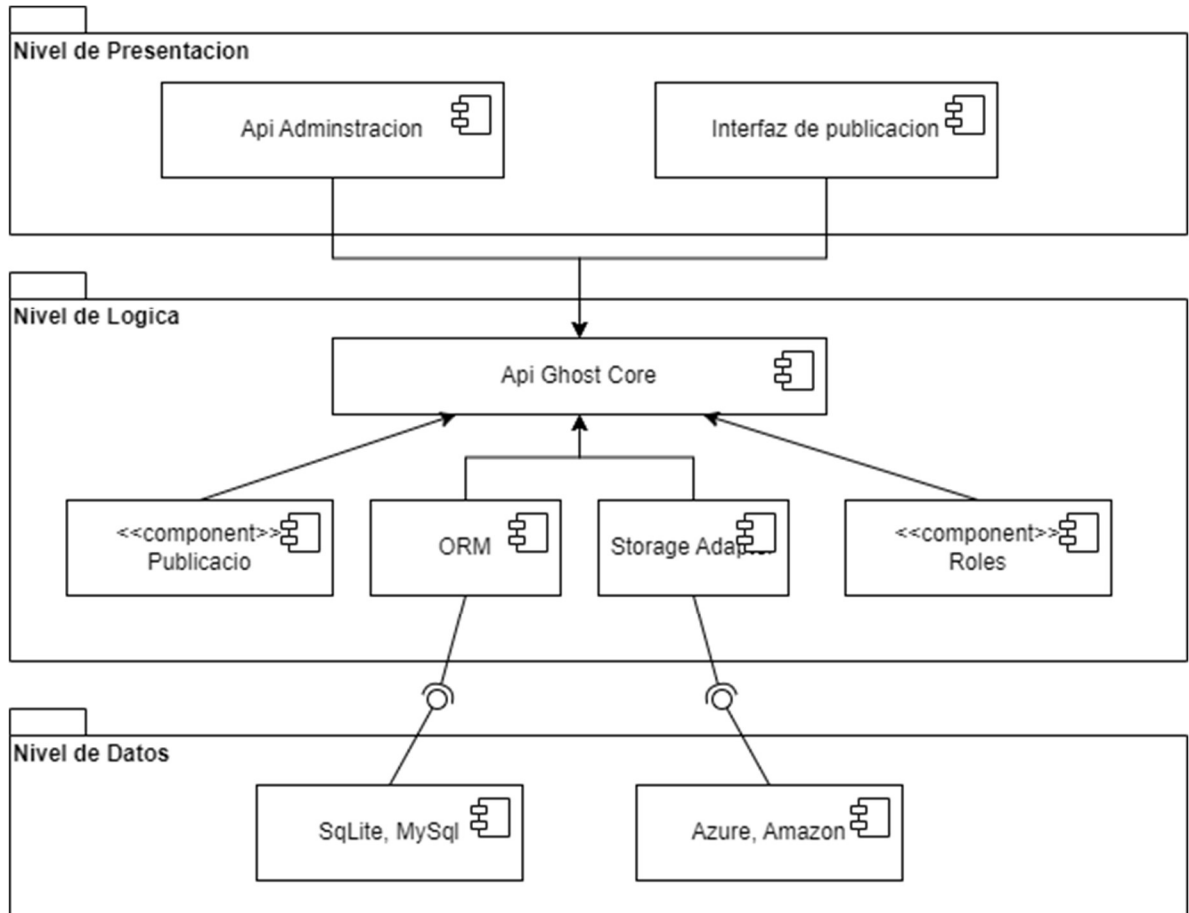
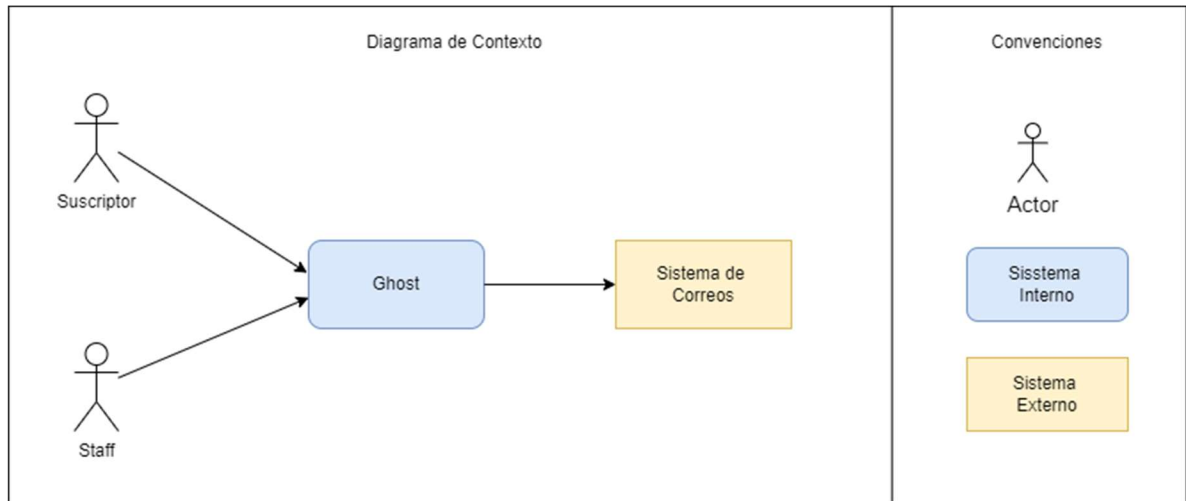
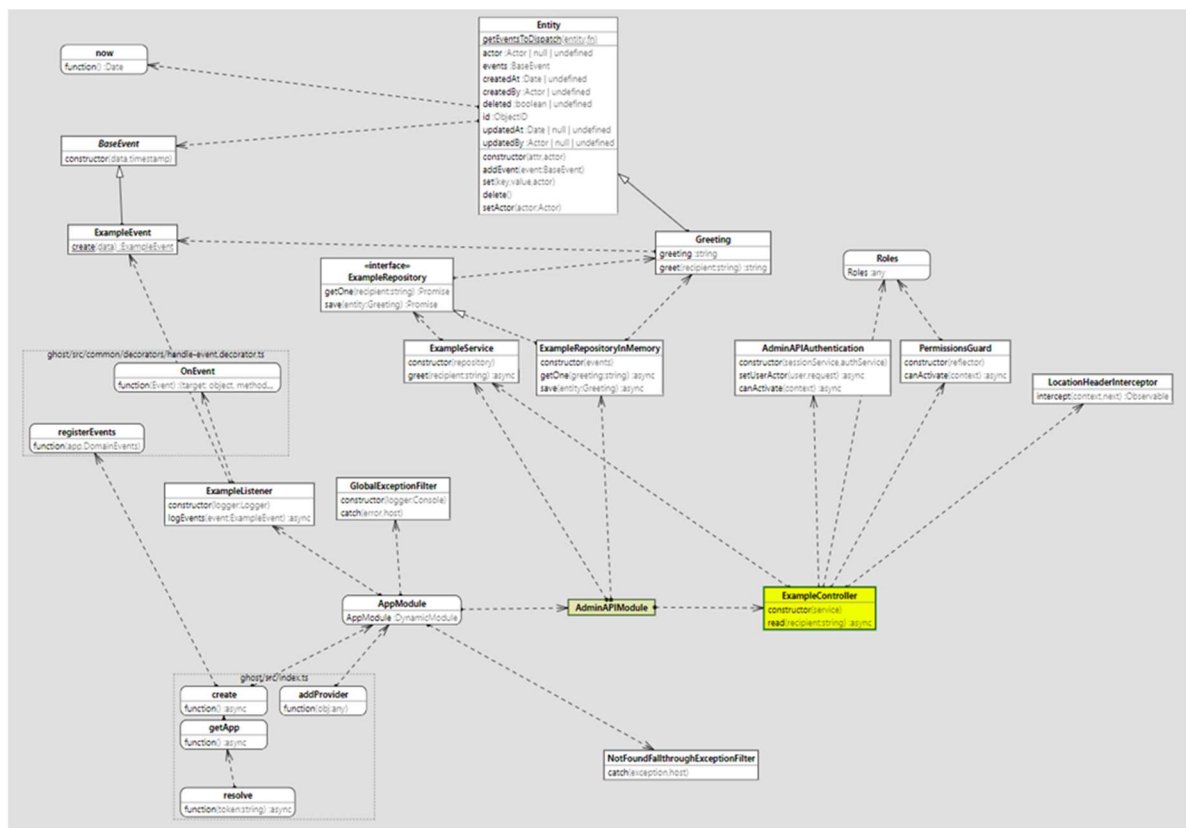


Diagrama de Contexto



Modelo de datos



2 Contexto de la estrategia de pruebas

2.1 Objetivos:

- Automatizar pruebas de regresión y funcionales claves para garantizar que las funcionalidades principales del sistema no se vean afectadas por nuevas implementaciones. Esto incluye la creación y mantenimiento de scripts automatizados para ejecución en paralelo en AWS.
- Ejecutar pruebas exploratorias y de integración manuales para detectar defectos en las interacciones entre módulos y en áreas críticas que puedan no estar completamente cubiertas por la automatización.
- Realizar pruebas de carga y rendimiento en módulos críticos para asegurar la estabilidad y desempeño del sistema bajo condiciones controladas de carga.
- Maximizar el uso eficiente de los recursos asignados (recursos humanos y computacionales) para completar las pruebas dentro del tiempo disponible, aprovechando la ejecución en paralelo en AWS.

Funcionalidades a probar:

- Creación de cuenta (primer uso)
- Iniciar sesión
- Configuración de título, idioma y zona horaria
- Invitación de usuarios
- Agregar y publicar posts
- Agregar y publicar páginas
- Adición de elementos web

2.2 Duración de la iteración de pruebas:

La duración total de la iteración de pruebas será de **1 semana**.

Dado el tiempo limitado, se implementará la ejecución en paralelo de pruebas automatizadas para maximizar el uso de las 200 horas de máquina en Amazon AWS.

2.3 Presupuesto de pruebas:

2.3.1 Recursos Humanos:

Ingenieros Automatizadores Junior:

50 horas asignadas a cada uno (3 ingenieros en total), dedicadas a la creación, mantenimiento, y ejecución de pruebas automatizadas. Utilizarán herramientas como Monkey-Cypress para pruebas exploratorias y RIPuppet para rutas guiadas de interacción.

Ingenieros de Soporte Adicional:

Cada uno tiene 10 horas asignadas para colaborar en pruebas manuales, exploratorias y asistencia en configuración de entornos y análisis de resultados, sumando 40 horas en total.

Total de Horas Disponibles:

- Ingenieros Automatizadores Junior: $3 \times 50 \text{ horas} = 150 \text{ horas}$
- Equipo Adicional de Soporte: $4 \times 10 \text{ horas} = 40 \text{ horas}$

2.3.2 Recursos Computacionales:

- 200 horas/máquina en Amazon AWS: Debido a la limitación de tiempo (1 semana), las horas se distribuirán para maximizar la ejecución en paralelo de pruebas automatizadas.

Distribución de Horas en AWS:

- 45 horas - Creación, mantenimiento y ejecución de pruebas automatizadas de integración, regresión y validación funcional mediante scripts optimizados.
- 15 horas - Pruebas de rendimiento y carga en áreas críticas del sistema, asegurando estabilidad y capacidad de respuesta bajo carga.
- 120 horas - Pruebas exploratorias automatizadas y validación de integraciones mediante Monkey-Cypress para interacciones aleatorias y RIPuppet para rutas guiadas en flujos de usuario.
- 20 horas - Pruebas manuales ejecutadas por el equipo adicional, enfocadas en áreas clave del sistema para asegurar una revisión exhaustiva de los flujos críticos.

2.3.3 Recursos Económicos para la Contratación de Servicios/Personal:

- No se prevé contratación adicional de personal.
- Los servicios en AWS estarán limitados a las 200 horas asignadas. Es crucial monitorizar y ajustar el uso para evitar sobre costos y asegurar el uso eficiente del tiempo.

2.4 TNT (Técnicas, Niveles y Tipos) de Pruebas:

Técnicas:

- Pruebas automatizadas: Principalmente pruebas de regresión, funcionales y de integración, utilizando herramientas como Selenium, JUnit o frameworks equivalentes.
- Pruebas manuales: Se enfocarán en pruebas exploratorias, validando funcionalidades clave no cubiertas por los scripts automatizados.
- Pruebas de carga y rendimiento: Ejecución de pruebas en AWS para evaluar el rendimiento bajo condiciones de estrés controlado.

Niveles:

- Pruebas unitarias: Automatizadas y ejecutadas continuamente como parte del ciclo de integración continua (realizadas por el equipo de desarrollo).
- Pruebas de integración: Validación de la interacción entre módulos clave.
- Pruebas de sistema: Pruebas automatizadas y manuales para asegurar que el sistema cumpla con los requisitos generales.
- Pruebas de aceptación: Son las pruebas realizadas en los ambientes reales de producción para la aceptación del cliente.

Tipos:

- Pruebas funcionales: Evaluarán el comportamiento del sistema según los requisitos.
- Pruebas no funcionales: Evalúan aspectos no funcionales como el rendimiento
- Prueba de caja blanca: Evalúa la lógica interna del código, verificando su estructura y funcionamiento sin considerar las entradas y salidas.
- Prueba de caja negra: Se centra en las entradas y salidas del sistema, validando que funcione según los requisitos sin necesidad de conocer su implementación interna.
- Pruebas de rendimiento/carga: validarán tiempos de respuesta y estabilidad bajo carga en puntos críticos.
- Pruebas exploratorias: Se realizarán para detectar defectos en áreas del sistema no cubiertas por pruebas automatizadas.

Nivel	Tipo	Técnica	Objetivo
Pruebas Unitarias	Funcionales De caja blanca	Automatización continua	Verificar cada unidad de código individual para asegurar funcionalidad y calidad básica del código
Pruebas de Integración	Funcionales De caja negra	Automatización y manual	Validar la correcta interacción entre módulos clave en el sistema
Pruebas de Sistema	Funcionales De Caja negra	Automatización y exploración	Garantizar que el sistema cumpla con los requisitos de usuario mediante pruebas funcionales
Sistema	Funcional	Pruebas Automatizadas de Regresión Visual	Identificar defectos tras cambios en funcionalidades mediante regresión visual en las áreas críticas.
Pruebas de Sistema	Exploratorias De Caja negra	Exploración manual	Detectar problemas en áreas no cubiertas por automatización
Pruebas de Sistema	Rendimiento/Carga De Caja negra	Simulación de carga en AWS	Asegurar la estabilidad del sistema bajo condiciones de alta demanda

2.5 Distribución de Esfuerzo:

Siguiendo el patrón de la pirámide de pruebas, la estrategia se enfoca en una base sólida de pruebas automatizadas de niveles inferiores (unitarias y de integración), con menos esfuerzo en pruebas manuales y de sistema para una validación más rápida y confiable.

Pruebas Unitarias Automatizadas (50%):

- La base de la pirámide está compuesta principalmente por pruebas unitarias automatizadas, que aseguran la funcionalidad de cada unidad de código en un entorno controlado.
- Estas pruebas están integradas en el ciclo de integración continua y permiten identificar defectos rápidamente antes de pasar a niveles superiores.

Pruebas de Integración Automatizadas (20%):

- En el siguiente nivel, las pruebas de integración validan la interacción entre los módulos clave, garantizando que cada componente funcione en conjunto según lo previsto.
- Estas pruebas también se automatizan y se ejecutan en paralelo para maximizar el uso de las 200 horas de AWS.

Pruebas de Sistema Automatizadas y de Regresión Visual (20%):

- Este nivel incluye pruebas funcionales y de regresión automatizadas para asegurar que el sistema completo cumple con los requisitos y que nuevas funcionalidades no afecten las existentes.
- Dado el tiempo limitado, estas pruebas se ejecutarán en paralelo en AWS para maximizar la eficiencia.

Pruebas Exploratorias Manuales (10%):

- En la punta de la pirámide se ubican las pruebas manuales exploratorias, que son importantes para detectar defectos en áreas críticas que no se pueden cubrir completamente con automatización.
- El equipo adicional se enfoca en estas pruebas para revisar escenarios específicos y garantizar que el sistema tenga un comportamiento consistente en casos únicos.

2.6 Justificación del Uso de Monkey-Cypress y RIPuppet

Ambas herramientas se utilizarán para complementar las pruebas automatizadas:

Monkey-Cypress: Ideal para pruebas exploratorias, generando interacciones aleatorias que permiten identificar errores inesperados en áreas no cubiertas por pruebas específicas.

RIPuppet: Permite definir algunas rutas de interacción, lo que proporciona una exploración guiada y ayuda a descubrir errores en secuencias de usuario más comunes.

2.7 Análisis de Herramientas

Herramienta	Pros	Contras
Monkey-Cypress	Configuración rápida y sin scripts, ideal para pruebas aleatorias y exploratorias.	Puede generar errores irrelevantes debido a la aleatoriedad, lo que requiere análisis extra.
RIPuppet	Exploración guiada en rutas específicas, útil para identificar fallos en flujos de usuario.	Configuración inicial más compleja; puede limitar la aleatoriedad si no se ajusta adecuadamente.

AnexoS

- Manual básico de cypress.pdf
- Precios de referencia de AWS
- Link de video presentación: <https://youtu.be/c8PeyOzRXOw>
- Link de prueba monkey: https://youtu.be/cdeP_LdMUms