

Estrategia de Pruebas

Nombre Aplicación: Ghost

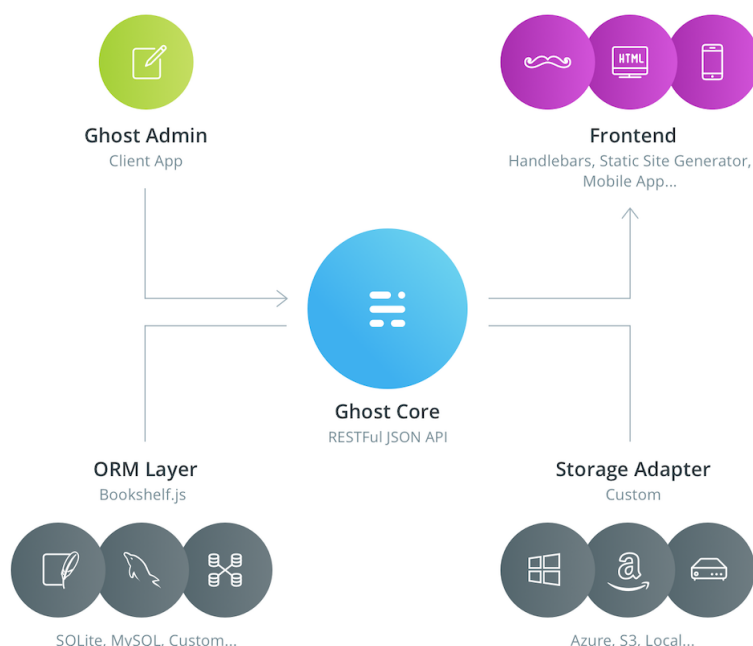
Versión: 3.41.1

Descripción: Es una plataforma open-source enfocada en periodistas independientes y escritores alrededor del mundo, esto se logra a través de la publicación y gestión de blogs. Fue escrita originalmente en javascript y lanzada en 2013.

Funcionalidades Core

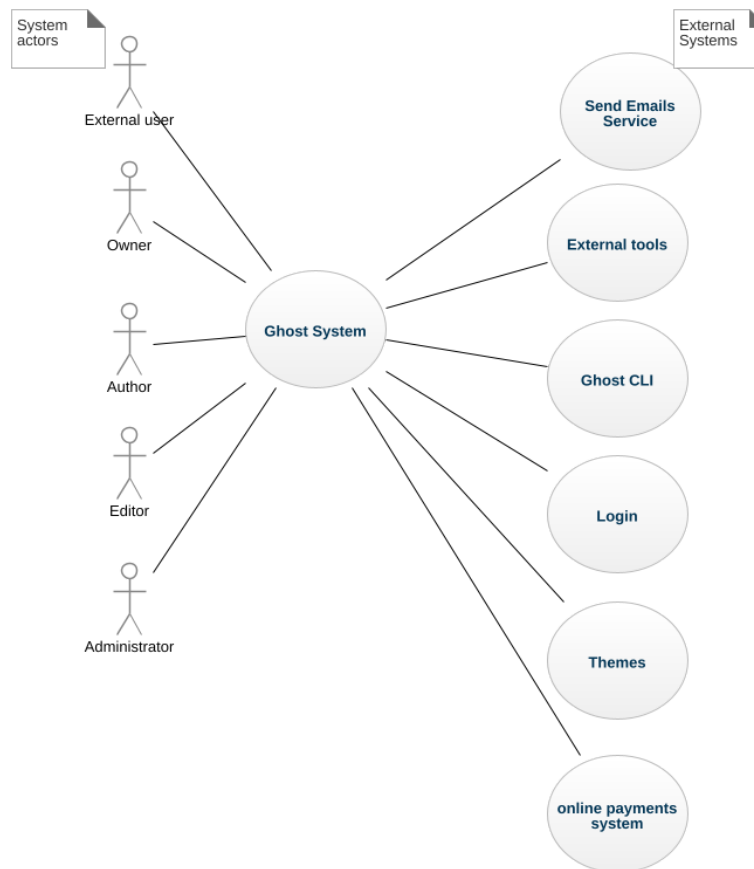
- Crear, editar, ver y filtrar nuevos post por campos específicos.
- Modificar la navegación del menú principal y personalizar el tema del blog.
- Creación y publicación de páginas para agregar al blog.
- Creación de Tags para asociarlos a los posts.
- Ajustar la información general del blog como título, url de redes sociales, tags, etc.
- Autenticación y autorización de usuarios.
- Ver sitio y navegar post los posts publicados.

Diagrama de Arquitectura

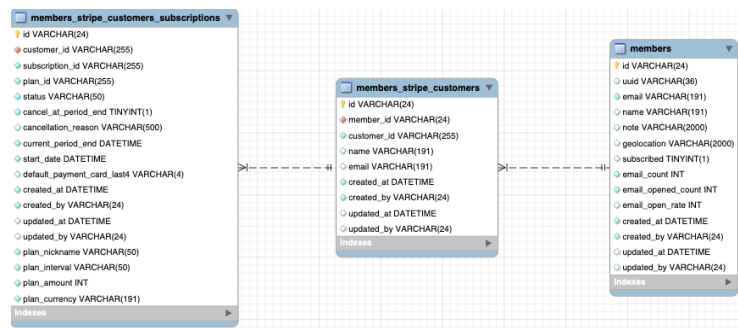


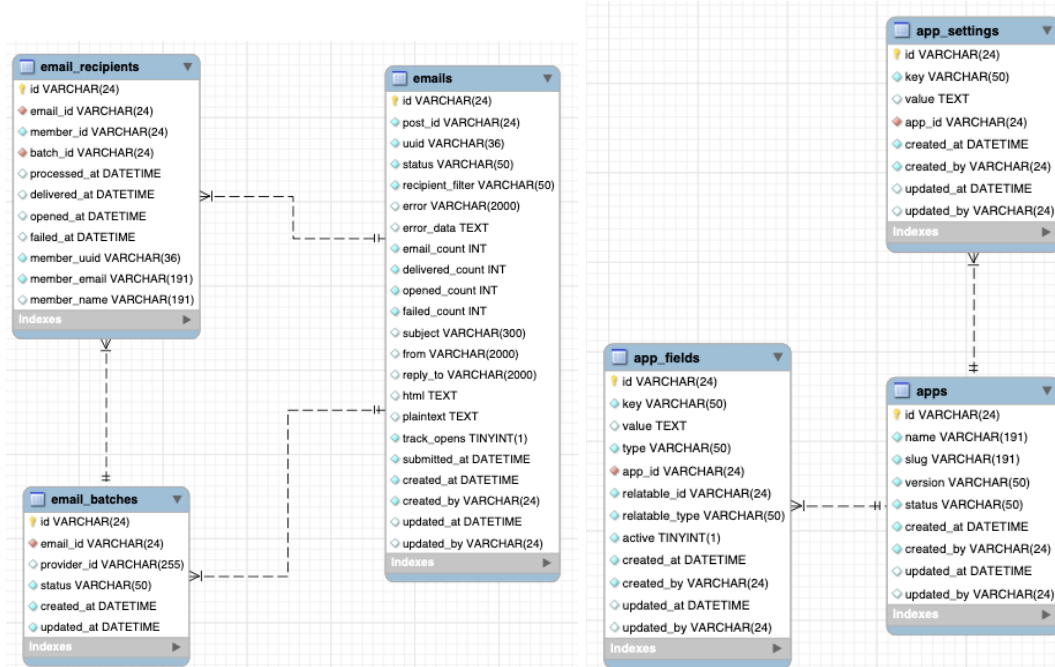
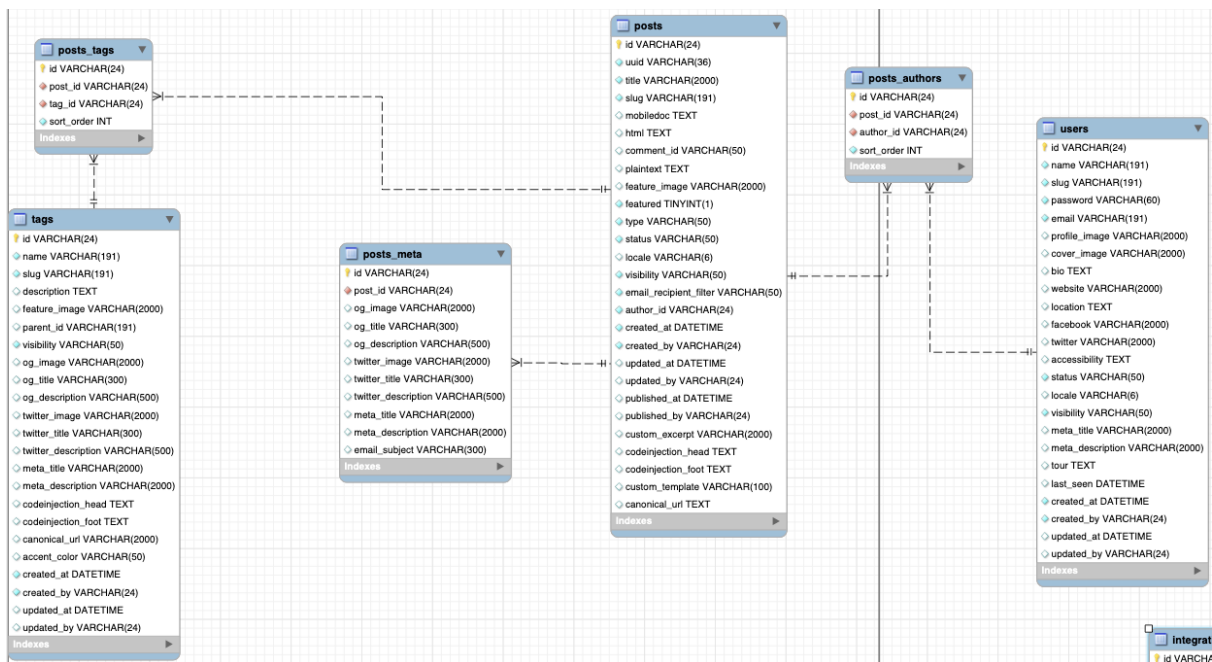
Se observa en el diagrama que la aplicación ghost cuenta con un Core que es un restful API que es utilizada en el admin (client app) y en el front-end. también cuenta con una capa ORM que puede ser personalizada o con tecnologías como SQLite o MySQL, además de esto se puede adaptar para utilizar almacenamiento personalizado(azure,amazon,etc).

Diagrama de Contexto

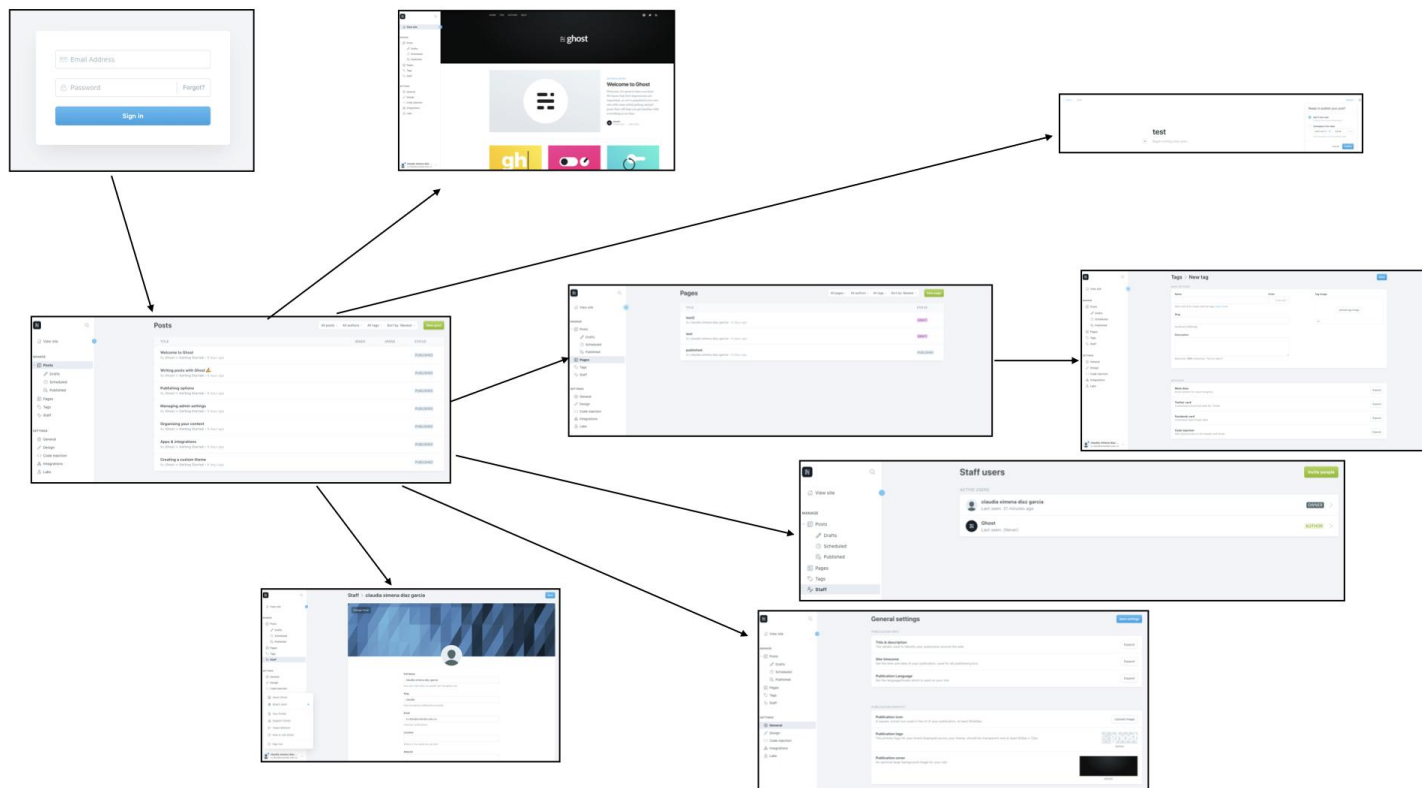


Modelo de Datos





Modelo de GUI



Contexto de la estrategia de pruebas

Objetivos

- Probar las funcionalidades core del aplicativo Ghost.
- Encontrar defectos de usabilidad y accesibilidad.
- Conocer a priori las funcionalidades del aplicativo
- Explorar posibles escenarios de prueba
- Implementar pruebas automatizadas aleatorias
- Encontrar eventos aleatorios y casos raros.
- Explorar sistemáticamente la interfaz
- Generar entradas para formularios.
- Inducir errores en la interfaz gráfica
- Encontrar diferencias entre distintas versiones del aplicativo.
- Comparar diferentes versiones del aplicativo para hallar posibles fallos.
- Probar bajo extensa data diferentes formularios.
- Validar formatos y tipos de campo en formularios.

Presupuesto de pruebas

Para el presupuesto de esta estrategia se tienen dos ingenieros senior (64 horas/ persona a lo largo de 8 semanas)
- 50 horas/máquina en los equipos locales.

Distribución de Esfuerzo

Para la distribución de esfuerzo se establecen sprint de una semana.

Ingeniera Claudia Diaz - Senior

Sprint	Actividad	Esfuerzo
Sprint 1	Realización de pruebas exploratorias manuales	6 horas /persona
Sprint 1	Generación de reporte	2 horas /persona
Sprint 2	Análisis de estrategias de pruebas automatizadas	8 horas / persona
Sprint 3	Creación de la estrategia de pruebas y planeación.	8 horas /persona
Sprint 4	Ejecución de pruebas aleatorias (Cypress)	6 horas /persona
Sprint 4	Generación de reporte e incidencias	2 horas/persona
Sprint 4	Ejecución de pruebas aleatorias	5 horas / maquina
Sprint 5	Ejecución de pruebas E2E (Cypress)	6 horas /persona
Sprint 5	Generación de reporte e incidencias	2 horas/persona
Sprint 5	Ejecución de pruebas aleatorias	5 horas / maquina
Sprint 6	Ejecución de pruebas de regresión visual (VRT)	7 horas / persona
Sprint 6	Ejecución de pruebas de regresión visual (VRT)	7 horas / máquina
Sprint 6	Reporte de pruebas e incidencias	1 hora / persona
Sprint 7	Ejecución de pruebas de volumen - Inyección de datos	7 horas / persona
Sprint 7	Ejecución de pruebas de volumen - Inyección de datos	7 horas / máquina
Sprint 7	Generación de reporte e incidencias	1 hora / maquina
Sprint 8	Consolidación de resultados en reporte	4 horas / persona
Sprint 8	Ajustes de incidencias en tablero	2 horas / persona

Sprint 8	Presentación de resultados al equipo	2 horas/ persona
Total Horas del Ingeniero automatizador		64 Horas
Total Horas máquina		25 horas

Ingeniera Enrique Gamba - Senior

Sprint	Actividad	Esfuerzo
Sprint 1	Realización de pruebas exploratorias manuales	6 horas /persona
Sprint 1	Definición de pruebas exploratorias .	2 horas /persona
Sprint 2	Análisis de estrategias de pruebas automatizadas	8 horas /persona
Sprint 3	Creación de la estrategia de pruebas y planeación.	8 horas /persona
Sprint 4	Ejecución de pruebas aleatorias (RIPuppet)	6 horas /persona
Sprint 4	Generación de reporte e incidencias	2 horas/persona
Sprint 4	Ejecución de pruebas aleatorias	5 horas / maquina
Sprint 4	Ejecución de pruebas E2E (Kraken)	6 horas /persona
Sprint 5	Generación de reporte e incidencias	2 horas/persona
Sprint 5	Ejecución de pruebas aleatorias	5 horas / maquina
Sprint 6	Ejecución de pruebas de regresión visual (VRT)	7 horas / persona
Sprint 6	Ejecución de pruebas de regresión visual (VRT)	7 horas / máquina
Sprint 6	Reporte de pruebas e incidencias	1 hora / persona
Sprint 7	Ejecución de pruebas de volumen - Inyección de datos	7 horas / persona
Sprint 7	Ejecución de pruebas de volumen - Inyección de datos	7 horas / máquina
Sprint 7	Generación de reporte e incidencias	1 hora / maquina
Sprint 8	Consolidación de resultados en reporte	4 horas / persona
Sprint 8	Ajustes de incidencias en tablero	2 horas / personas
Sprint 8	Presentación de resultados al equipo	2 horas/ persona
Total Horas del Ingeniero automatizador		64 Horas
Total Horas máquina		25 horas

Recursos Humanos

Para lograr los objetivos del proyecto se espera que los 2 automatizadores senior tengan como mínimo tres años de experiencia en pruebas automatizadas y conocimientos básicos en pruebas manuales, además de esto es deseable que su experiencia previa sea en aplicaciones web. Por lo menos uno de los dos automatizadores debe tener capacidades de liderazgo y comunicación pues este será el encargado de la comunicación con el PL.

La herramienta para el registro de incidencias será Jira, por esto mismo los dos deben tener conocimientos en esta herramienta.

Tiempo disponible:

- Ingeniero automatizador 1 (encargado) - 64 horas
- Ingeniero automatizador 2 (con mayor experiencia) - 64 horas

Recursos Computacionales

- Se contará con dos computadores en donde cada tester ejecutará sus pruebas
- No se utilizarán recursos móviles ni de grabación. Se espera que las pruebas sobre dispositivos móviles sean emulados desde el navegador.

Características de los computadores::

- Procesador Intel i7 (Novena generación))
- Memoria (GiB): 16
- Almacenamiento: 1TB
- Disco de estado sólido
- Ancho de banda de EBS dedicado (Mbps): 450
- Rendimiento de red: Moderado

Recursos Económicos para la contratación de servicios:

No se necesitan recursos económicos para la contratación de servicios externos.

TNT (Técnicas, Niveles y Tipos) de pruebas

Tipo	Nivel	Técnica	Objetivo
Funcionales Positivas Negativas	Sistema	E2E Pruebas Manuales Pruebas Exploratorias Monkeys Rippers VRT Generación automática de datos	Probar las funcionalidades core del aplicativo Ghost.
Funcionales	Sistema	E2E Pruebas Manuales Pruebas Exploratorias	Encontrar defectos de usabilidad y accesibilidad.
Funcionales Positivas Negativas	Sistema	Pruebas Manuales	Conocer a priori las funcionalidades del aplicativo
Funcionales	Sistema	Pruebas Exploratorias	Conocer a priori las funcionalidades del aplicativo
Funcionales Positivas Negativas	Sistema	Pruebas Manuales	Explorar posibles escenarios de prueba
Funcionales	Sistema	Pruebas Exploratorias	Explorar posibles escenarios de prueba
Funcionales Positivas Negativas	Sistema	Monkeys	Implementar pruebas automatizadas aleatorias
Funcionales Positivas Negativas	Sistema	Rippers	Implementar pruebas automatizadas aleatorias
Funcionales Positivas Negativas	Sistema	Monkeys	Encontrar eventos aleatorios y casos raros.

Funcionales Positivas Negativas	Sistema	Rippers	Explorar sistemáticamente la interfaz
Funcionales Positivas Negativas	Sistema	E2E	Explorar sistemáticamente la interfaz
Funcionales Positivas Negativas	Sistema	E2E	Generar entradas para formularios.
Funcionales Positivas Negativas	Sistema	E2E	Inducir errores en la interfaz gráfica
Funcionales	Sistema	VRT	Encontrar diferencias entre distintas versiones del aplicativo.
Funcionales	Sistema	VRT	Comparar diferentes versiones del aplicativo para hallar posibles fallos.
Funcionales Positivas Negativas	Sistema	Generación automática de datos	Probar bajo extensa data diferentes formularios.
Funcionales Positivas Negativas	Sistema	Generación automática de datos	Validar formatos y tipos de campo en formularios.

Conclusión

Es interesante los diferentes usos que tienen las herramientas de pruebas automatizadas y cómo se puede implementar una u otra según lo requiera la aplicación a probar. También al mezclar herramientas se pueden hacer pruebas más robustas y complejas.

A continuación compartiremos una tabla comparativa con los pros y los contras que identificamos durante el curso:

Herramienta	Pros	Contras
Apis de automatización	<ul style="list-style-type: none">• Permite hacer pruebas más rápidas que las manuales• Permite encontrar casos raros en las pruebas aleatorias• Fácil de implementar, es solo descargar y agregar las funcionalidades• Es muy manual la forma como se debe ir y ver los nombres y clases de los elementos para que sean accedidos• Doble trabajo ya que hay probar dos versiones de la misma aplicación y cada una tiene diferentes elementos	<ul style="list-style-type: none">• Pruebas muy frágiles, si se cambia el id de un campo ya no funciona la prueba• Al encontrar un solo error se termina todo el proceso a probar• Hay que seleccionar específicamente el elemento y no siempre lo identifica fácilmente• Es interesante el tomar los screenshots para analizar las pantallas y que sean usadas para una regresión visual
E2E	<ul style="list-style-type: none">• Es más cercano a simular un comportamiento real de parte de un usuario y es más coherente en los pasos a seguir.• Permite encontrar objetos de acuerdo a los identificadores, nombres, etc de los elementos de la página, por lo que es más sencillo indicar en donde hacer el evento.• Permite crear escenarios de pruebas	<ul style="list-style-type: none">• Hay que separar las feature por archivo, lo que implica que si hay muchas pruebas van a haber muchos archivos.• Para las mismas pruebas algunas veces pasan y

	de forma más sencilla e intuitiva ya que usa lenguaje natural BDD.	otras no, por lo que no es confiable el resultado de las pruebas.
VRT	<ul style="list-style-type: none"> • Es muy rápido hacer las comparaciones para dos versiones de la misma herramienta y así enfocarse en las funcionalidades que cambiaron 	<ul style="list-style-type: none"> • Es muy milimétrico y si las pantallas son iguales pero hay una diferencia en la posición de un elemento ya lo toma como falla
Generación de datos	<ul style="list-style-type: none"> • Es muy ágil la manera como genera datos y tipos de datos aleatorios sin necesidad de crear pruebas una a una • Permite hacer gran volumen de pruebas y encontrar validaciones de campos por tipos de datos 	<ul style="list-style-type: none"> • Por la naturaleza de Ghost hay pruebas en donde se no se encuentran fallas y en otras no, ya que cambian los id de los componentes • Las validaciones de Ghost son bastante acertadas, valida tamaños de los campos, valida nulos, y tipos de datos