

Estrategia de Pruebas

1. Aplicación Bajo Pruebas

1.1. Nombre Aplicación: Ghost

1.2. Versión: 5.18.0

1.3. Descripción:

Ghost es una plataforma de publicación profesional de código abierto basada en una pila de tecnología moderna de Node.js, diseñada para equipos que necesitan potencia, flexibilidad y rendimiento.

Ghost es un CMS (por sus siglas en ingles Content Management System) en la nube, el cual te permite crear, organizar, publicar y eliminar contenidos de tu sitio web. Las CMS proveen 3 funciones principales:

- Creación de webs.
- Gestión y mantenimiento del sitio web.
- Administración de páginas web y del propio CMS.

Otras plataformas (en comparación con Ghost) abiertas son generalmente antiguas, lentas e infladas, mientras que otras plataformas cerradas no le otorgan absolutamente ningún control o propiedad de su contenido. Ghost ofrece lo mejor de ambos mundos y más.

1.4. Enlace al video de Explicación:

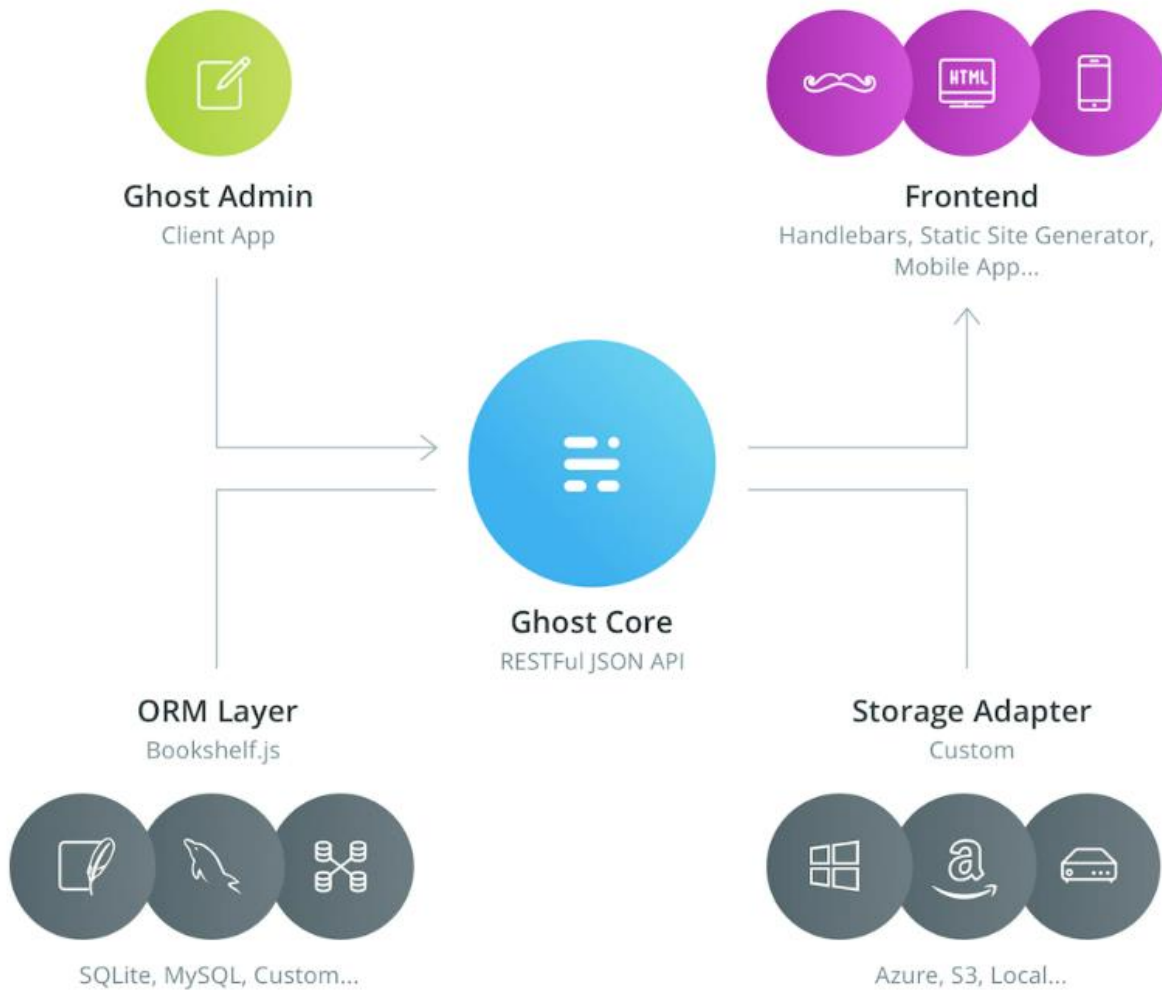
<https://youtu.be/ku6u860dE1M>

1.5. Funcionalidades Core:

Ghost provee una serie de funcionalidades entre las cuales encontramos:

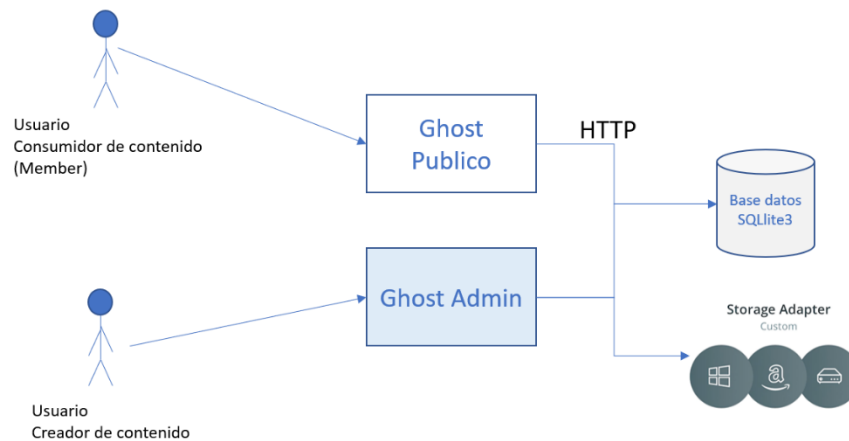
- Editar, crear y publicar un post.
- Borrar, despublicar y programar un post.
- Editar, crear, y publicar una página.
- Borrar, y despublicar una página.
- Editar, crear, y guardar un tag.
- Borrar, y despublicar un tag.
- Administrar el blog con distintos usuarios y roles específicos.
- Publicar en una fecha específica.

1.6. Diagrama de Arquitectura:



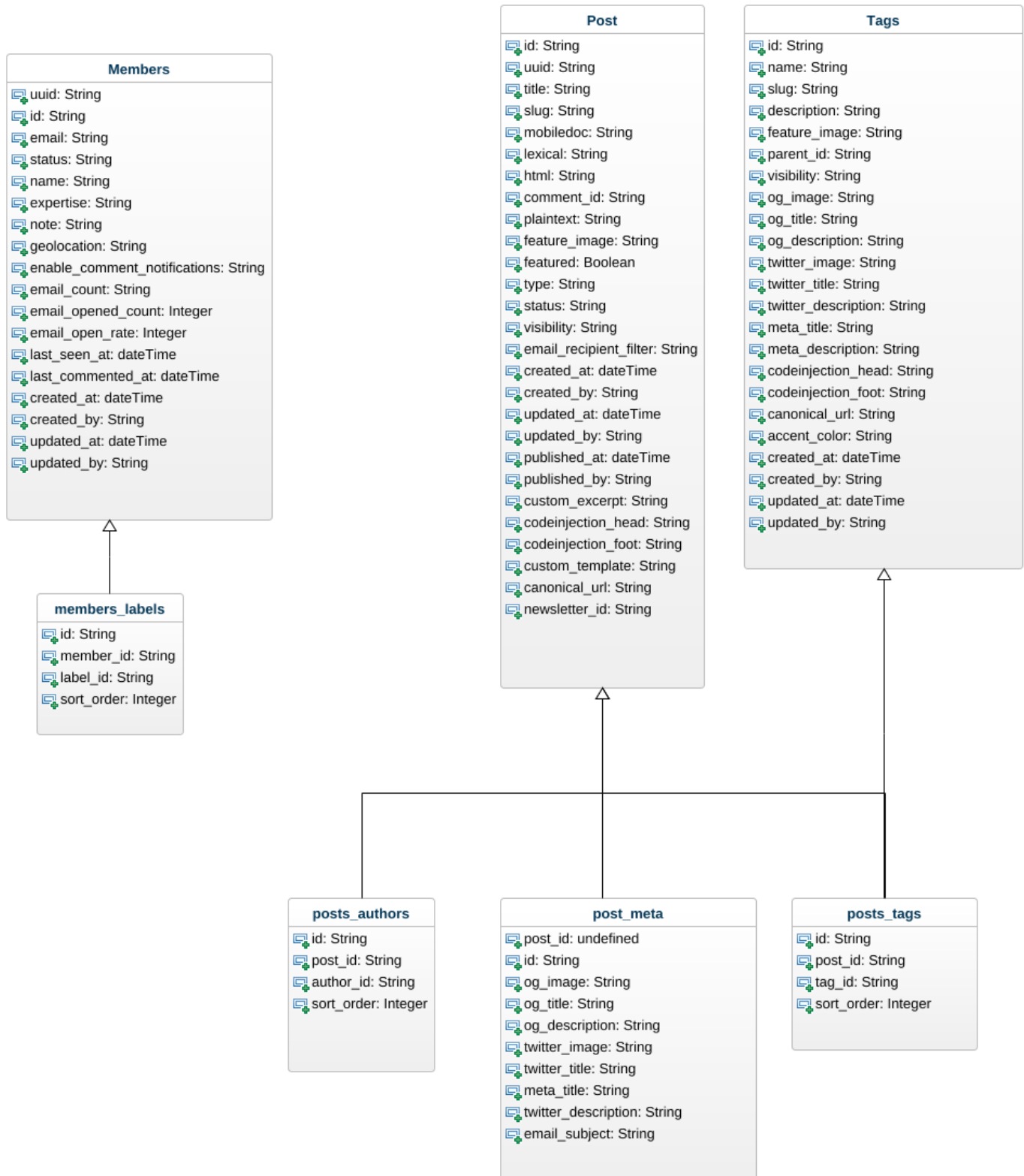
Si la imagen no se aprecia del todo una copia se encuentra dentro del zip de entrega o en el siguiente enlace:
https://1drv.ms/u/s!As_QMF2T0qpGgbpEXH1I_ksDyV2GcQ?e=llg8Vf

1.7. Diagrama de Contexto:



Si la imagen no se aprecia del todo una copia se encuentra dentro del zip de entrega o en el siguiente enlace:
https://1drv.ms/u/s!As_QMF2T0qpGgbpCd1lh6P5wSTafg?e=fWpYof

1.8. Modelo de Datos:



Si la imagen no se aprecia del todo una copia se encuentra dentro del zip de entrega o en el siguiente enlace:

https://1drv.ms/u/s!As_QMF2T0qpGgbpBYUsPm1EPM9ojyA?e=ZucZ2N

Plantilla elaborada por

THE SW DESIGN LAB

1.9. Modelo de GUI:



Si la imagen no se aprecia del todo una copia se encuentra dentro del zip de entrega o en el siguiente enlace:

https://1drv.ms/u/s!As_QMF2T0qpGgbpAaZqL_GglUdPoVw?e=yINb41

2. Contexto de la estrategia de pruebas

2.1. Objetivos:

El objetivo de esta estrategia es la detección de la mayor cantidad de errores posibles en el periodo de la iteración, es por esto por lo que se enfocara más que nada en la parte funcional de la ABP, pero como el recurso local (Ingeniero Automatizador) es de poca experiencia, se espera que este haga y documente pruebas manuales exploratorias que realice durante parte de la iteración, como parte de su proceso de entrada al proyecto.

Segundo objetivo es la entrega de pruebas automatizadas (tanto la proporcionadas por el servicio externo contratado como por el equipo interno), en donde se entreguen al menos 60 pruebas automatizadas (50 por parte del proveedor externo del tipo Caja Negra de GUI y 10 pruebas de caja blanca unitarias por parte del equipo interno).

Tomando en cuenta que en la actualidad no se cuenta ni con inventario de pruebas realizadas (ni manuales ni automatizadas) ni reportes de defectos, se espera con esta estrategia aumentar rápidamente el numero de pruebas en el catálogo y utilizando el menor número de recursos disponibles.

2.2. Duración de la iteración de pruebas:

Estas pruebas serán ejecutadas en un periodo de 2 semanas, esto con la finalidad de alinearse con los estándares y buenas prácticas de los equipos agiles. También nos permitirá evaluar la estrategia después de un periodo relativamente corto y determinar los cambios necesarios.

2.3. Presupuesto de pruebas:

Concepto	Costo Unitario	Cantidad	Monto Total
Hora de Trabajo Ingeniero Junior	\$USD 5	30 horas	\$USD 150
Hora maquina Amazon AWS	\$USD 96	1 instancia (400 horas)	\$USD 96
Hora de Trabajo Ingeniero Senior	\$USD 10	40 horas	\$USD 400
Renta de Equipo de Computo	\$USD 50	1 equipos	\$USD 50
contratación de Servicios Externos	\$USD 500	1 sprint	\$USD 500
Total			\$USD 1196

Para 60 pruebas a ejecutar en esta estrategia, tendríamos un valor por prueba de \$USD19.93

2.3.1. Recursos Humanos

Para esta estrategia se espera contar con el siguiente personal:

- 1 ingeniero automatizador junior (30 horas/persona), el cual deberá tener los conocimientos teóricos en Pruebas Automatizadas, así como conocimientos relevantes en el uso de AWS (proyectos escolares, tutoriales de al menos 20 horas).
- 4 ingenieros de Software Senior con dedicación limitada (10 horas/persona) que ya poseen experiencia con la ABP y pueden proveer guía y ayuda a la hora de ser necesario al equipo interno y supervisión del equipo externo.

2.3.2. Recursos Computacionales

Se espera para esta estrategia se utilicen los siguientes recursos tecnológicos:

- 400 horas/máquina en Amazon AWS, estas horas maquina serán utilizadas para correr las pruebas automatizadas entregadas tanto por el equipo interno (al Ingeniero automatizador junior), como las entregadas por el servicio externo contratado.
- 1 computador correspondiente al Ingeniero automatizador junior. Estos ordenadores son MacBook Air modelos 2020 en adelante.
- 4 computadores una correspondiente a cada Ingeniero Software Senior. Estos ordenadores son MacBook Air modelos 2020 en adelante.

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

En esta estrategia se considera la siguiente contratación de servicios:

- 500 USD para contratación de servicios de outsourcing de pruebas, en el contrato se estableció la cuota de 10 USD por prueba automatizada funcionando, con lo cual se espera recibir de la empresa 50 pruebas automatizadas sobre la ABP distribuidas entre los diferentes Dispositivos y Navegadores (Android, IOS, Firefox, Chrome y Safari).
 - Las pruebas automatizadas contratadas con el outsourcing sobre cada funcionalidad serán completas, ejemplo: Post, debe incluir desde crear y programar fecha futura con al menos 10 tipos diferentes de contenido (texto, imagen, video, sonido, código HTML, links, botón, lista desplegable, etc.), como también, listar, editar todos los campos utilizados y eliminar un Post, esto implica ejecutar 2 tipos de prueba por funcionalidad, una positiva y una negativa, lo que en total permitirá barrer 25 funcionalidades diferentes.

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

Nivel	Tipo	Técnica	Objetivo
Sistema	Funcional Caja Negra Positivas	Automatizado	Estas son las pruebas que se espera que los servicios contratados entreguen, con la experiencia que tienen los terceros se espera que se entreguen una de pruebas del sistema completo sin mucha revisión de las funciones internas del sistema. Se espera que se entreguen los scripts automatizados para ser ejecutados en los Servidores de AWS.
Unidad	Funcional Caja Blanca Positivas	Automatizada	Estas pruebas tienen la intención de empezar la validación desde el nivel más alto de granularidad para detectar errores desde esta etapa, se quiere que las pruebas sean positivas validando con los inputs esperados las salidas de funciones. Se espera un script que corra sobre el repositorio del código y se ejecute de manera automática cada nuevo Pull Request
Sistema	Funcional Caja Negra Positivas	Manual	Son las pruebas exploratorias del Ingeniero Junior, la intención es que se familiaricen con la ABP y se ubiquen en las vistas y elementos de esta. Se espera un reporte manual con pasos y resultados.

2.5. Distribución de Esfuerzo

En esta estimación se espera poder dividir el tiempo del equipo interno en dos bloques de la siguiente forma:

1. Pruebas exploratorias manuales 10 horas/persona, ya que el ingeniero Junior necesita tiempo para conocer la ABP.
2. Prueba Unitarias Automatizadas 20 horas/persona, ya que así se podrán encontrar aún más errores diseccionando el código de la ABP y se empezara la construcción de catalogo formal de pruebas.

La intención es que a medida que las iteraciones pasen se puedan agregar más pruebas Unitarias y de Cobertura de código y se vaya empujando al equipo a llegar a la pirámide de automatización Ideal.

En el siguiente diagrama se incluyen las pruebas de los servicios contratados, en este caso al ser un equipo externo y ajeno a nosotros se diagrama en proporción a la cantidad de pruebas entregadas en comparaciones a las pruebas ejecutadas por el equipo interno.



3. Ventajas y Desventajas de la Propuestas

3.1. Ventajas

- Menor Costo a comparación de la estrategia 1.
- Mayor número de pruebas End to End de GUI.
- Genera flexibilidad en el uso de recursos al poder conseguir siempre personas con experiencia en pruebas en el outsourcing, ya que les pagamos por prueba realizada y no tenemos que encargarnos de su curva de aprendizaje.
- Entrega de valor alto al principio.
- Podemos obtener benchmarking de estrategias de pruebas y otros conocimientos de la industria a través de un tercero que nos permee de metodologías nuevas.

3.2. Desventajas

- A largo plazo la entrega de valor no se evidenciaría, ya que internamente la organización cuenta con el conocimiento del código y del funcionamiento del sistema a través de la madurez de su equipo de pruebas.
- Menor número de pruebas en general.
- Menor número de pruebas unitarias.
- Necesidad de supervisión.

- Poco crecimiento del equipo interno.
- Detección tardía de errores que aumenta la complejidad de reparación.
- Riesgo de Calidad de los entregables por parte de los servicios externos dado el poco conocimiento de la ABP.
- Se presenta el riesgo de que las pruebas no tengan repetibilidad, ya que se pueden estar cambiando de personal que realiza las pruebas manuales.