Estrategia de Pruebas

1. Aplicación Bajo Pruebas
   1. **Nombre Aplicación:** Ghost
   2. **Versión:** (3.41.1), (3.3.0), (2.x)
   3. **Descripción**: Plataforma de blogs gratuita y de código abierto escrita en JavaScript y distribuida bajo la Licencia MIT, diseñada para simplificar el proceso de publicación en línea.
   4. **Funcionalidades Core**:  
      1. Listado de Funcionalidades de Contenido:
   * **View Site:** vista preliminar al sitio o portal.
   * **Posts:** Publicaciones funcionalidades de: Crear, Eliminar, Actualizar, Filtros (Autor, Nombre, Noticias, Etiquetas).
     + **Drafts:** Borradores sobre las publicaciones.
     + **Scheduled:** Publicaciones programadas.
     + **Published:** Publicados.
   * **Pages:** Crear, Eliminar, Actualizar páginas web para las Publicaciones.
   * **Tags:** Crear, Eliminar, Actualizar etiquetas, Internas, Publicas.
   * **Staff Users:** administración de usuarios, Invitación vía email.  
     1. Listado de Funcionalidades de Ajustes:

* **General:** Ajustes Generales
  + **Publication Info:** Información sobre las publicaciones
    - **Title & Description:** Detalles usados para identificar las publicaciones al rededor la web.
    - **Site Time Zone:** asignar el tiempo y fecha de la publicación, usadas por todas las publicaciones publicadas.
    - **Publication Language:** asignar el lenguaje usado en el sitio web.
  + **Publication Identity:** Identidad de la publicación.
    - **Publicaton Icon:** Icono usado en la interfaz gráfica de usuario su publicacion.
    - **Publication Logo:** Logotipo principal del sitio desplegado en el tema.
    - **Publication cover:** imagen sobre puesta en el sitio.
  + **Site Meta Settings:**
    - **Meta Data:** Contenido extra en la ingeniería de búsqueda.
    - **Twitter Card:** Personalizar la estructura de datos para el sitio en Twitter
    - **Facebook Card:** Personalizar estructura de datos del sitio.
  + **Social Acounts:** Links a cuentas de redes sociales.
  + **Make this site private:** Llevar a cabo la publicación completa del sitio sea privada.
* **Design:** Modificaciones de aspecto y navegabilidad del sitio web.
  + **Navigation:** tabla con las correspondientes Url creadas para los menús de navegación.
  + **Secondary Navigation:** menú secundario de navegación.
  + **Theme Directory:** Plantillas para modificación visual del sitio web.
  + **Installed Themes:** temas instalados en el sitio.
* **Code Injection:** Inyección de código fuente para modificar el cabezote y el pie del sitio web.
* **Integrations:** Integraciones con aplicaciones web ya establecidas o propias.
* **Labs:** Nuevas mejoras experimentales sobre las funcionalidades:
  + **Members:** Miembros y tomar pagos sobre sus suscripciones.
  + **Import Content:** Importar publicaciones desde otra instalación Ghost.
  + **Export Content:** Exportar el contenido de todo el sitio, en formato JSON.
  + **Delete Content:** Borrar el contenido de las publicaciones desde las bases de datos
  + **Beta Features:** Características en versión de prueba.  
    1. Listado de Funcionalidades de Usuario:
* **About Ghost:** Acerca de la aplicación.
* **What’s new:** Que hay de nuevo en la versión.
* **Your Profile:** Seccion de los datos completos de usuario.
* **Support Center:** Centro de soporte.
* **Tweet @Ghost:** Twitter de ghost.
* **How to use Ghost:** Guía de aprendizaje de uso.
* **Sign Out:** Salir.  
  + 1. Listado de Funcionalidades Web:
* **Home:** Landing del sitio.
  1. **Diagrama de Arquitectura:**

Se lleva a cabo las correspondientes vistas de arquitectura de la aplicación mencionada. Se publican en la plataforma padlet en el correspondiente link público [Arquitectura de Referencia](https://padlet.com/wsanchezp1/46sav01xw9pkzm5n).

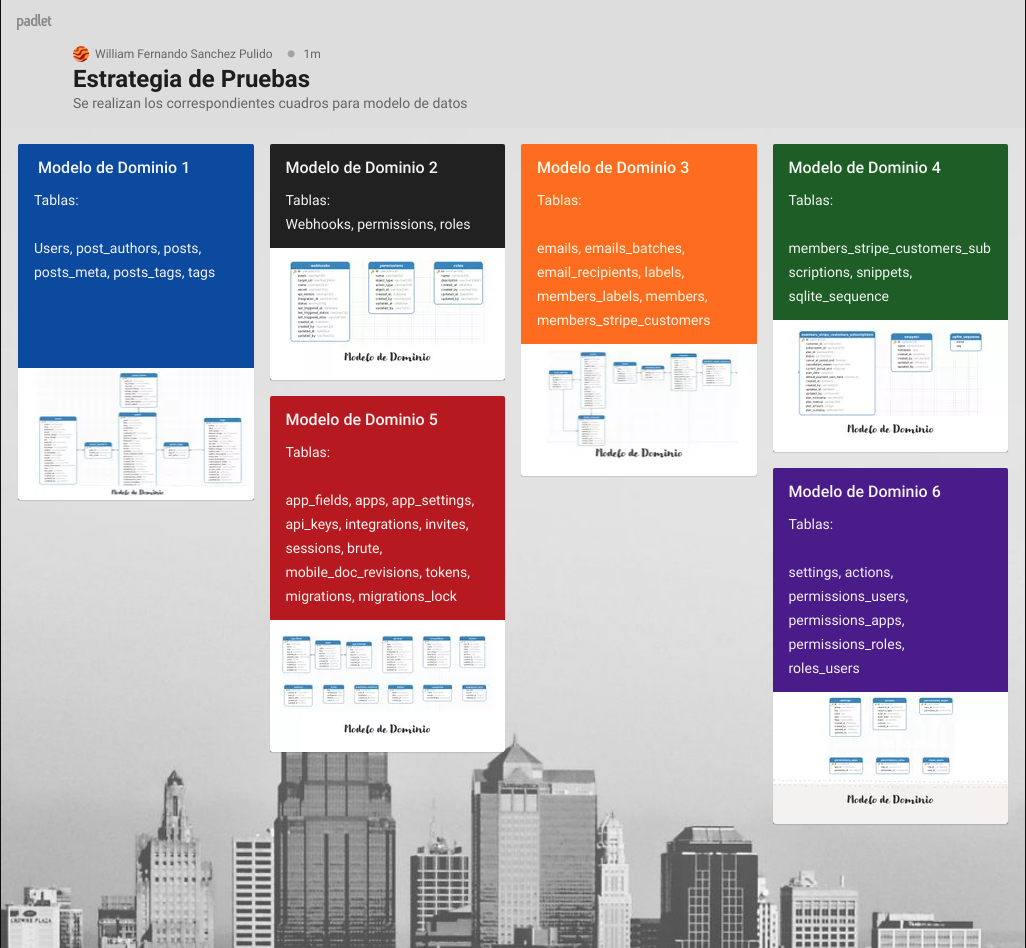


* 1. **Diagrama de Contexto:**

Se realiza el correspondiente diagrama de contexto por favor revisarlo en el siguiente link público: [Diagrama de Contexto](https://padlet.com/wsanchezp1/46sav01xw9pkzm5n)

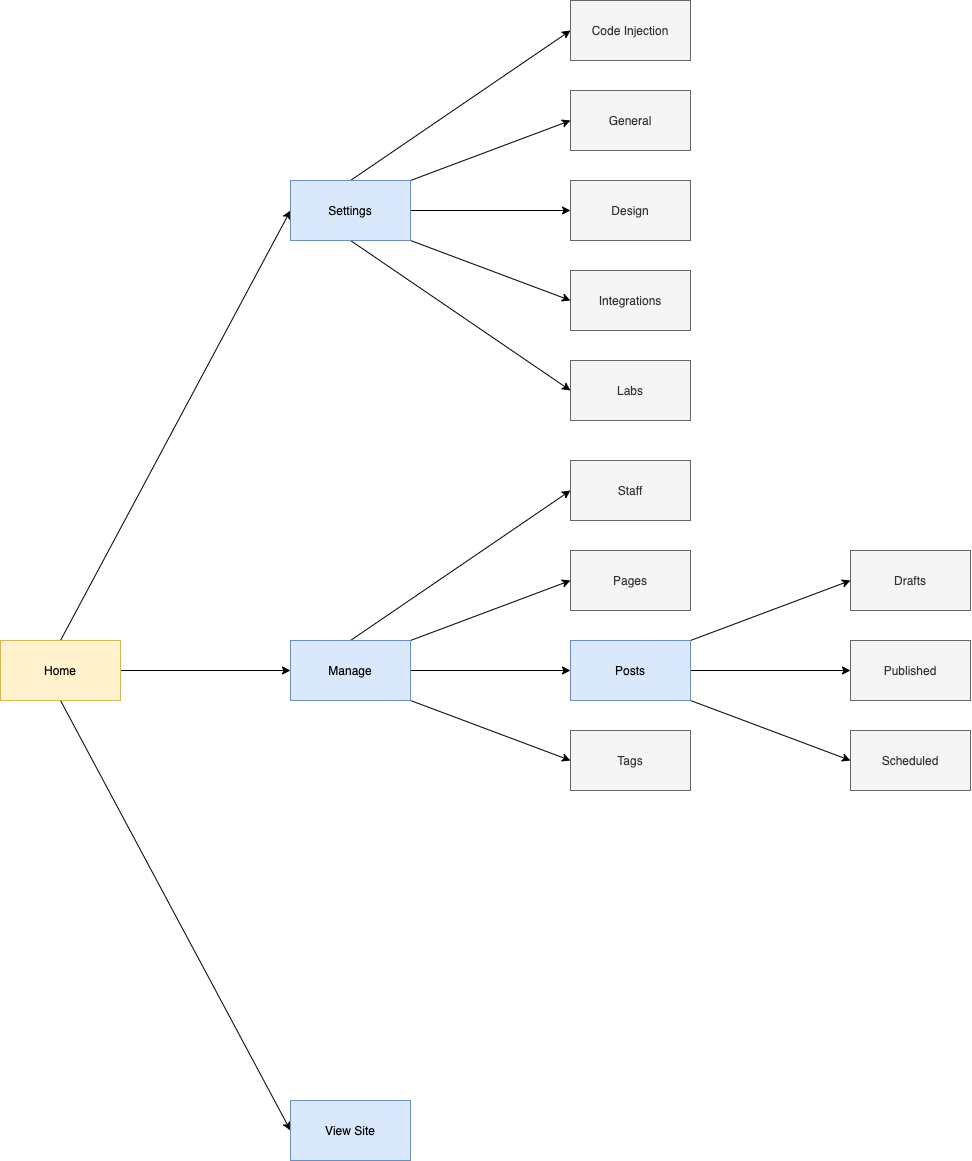
* 1. **Modelo de Datos:**

Se lleva a cabo el modelo de datos en la plataforma padlet, en el correspondiente link público: [Modelo de Datos](https://padlet.com/wsanchezp1/tujzk2r0yvqb4irm)



Tener en cuenta al momento de navegar en el hacer click en la correspondiente imagen contenida en la tarjeta para que esta se amplié y permita ver con mayor detalle el modelo.

* 1. **Modelo de GUI:**



1. **Contexto de la estrategia de pruebas**  
   1. **Objetivos:**

Antes de abarcar los objetivos queremos mencionar que el motivo por el cual tomamos la estrategia de pruebas mencionada por Kent C. Dodds, llamada Testing Trophy es dado a que esta se basa en el framework JavaScript el cual es la base de construcción de la aplicación a la cual se le está realizando el correspondiente proceso de pruebas en cada versión (ghost). Esta técnica tiene como pilar probar aplicaciones web e ir realizando los test automatizados nada más necesarios. Por lo cual se ajusta a la necesidad requerida y al uso de los insumos dados por el presupuesto obtenido.

Objetivo abarcar el 100% de cobertura de pruebas automatizadas por cada nivel mencionado en la estrategia

“Testing Trophy”.

|  |  |  |
| --- | --- | --- |
| Nivel | Porcentaje | Total, Porcentaje |
| Sistema | 100% | 100% |
| Integración | 100% |
| Unidad | 100% |
| Estáticas | 100% |
| Total, Cobertura | 100% |

* 1. **Presupuesto de pruebas:**

Para el correspondiente ejercicio de prueba se tendrá el presupuesto mencionado como:

* Tres ingenieros auto matizadores Senior, libertad en horas/máquina Nube.
  + 1. **Distribución de Esfuerzo:**

Para poder abarcar los objetivos mencionados se llevará la distribución de esfuerzo de la siguiente manera:

* Cada esfuerzo por ingeniero auto matizador estará orientado a un nivel y cada fase que se requiera realizar, para completar la tarea.
* Se describe el uso de la maquina correspondiente para la ambientación de pruebas y analizador de código estático.

|  |  |  |  |
| --- | --- | --- | --- |
| Recurso | Porcentaje  Dedicación | Fase | Nivel |
| IA-1 | 40% | Análisis y Diseño de Prueba | Sistema |
| IA-2 | 40% | Análisis y Diseño de Prueba | Integración |
| IA-3 | 40% | Análisis y Diseño de Prueba | Unidad |
| IA-4 | 40% | Gestor de la Configuración Exploratorias |  |
|  | | | |
| IA-1, IA-4 | 40% , 20% | Codificación Robot | Sistema |
| IA-2, IA-4 | 40% , 20% | Codificación Mock – Servidores Herméticos | Integración |
| IA-3 | 40% | Codificación Test | Unidad |
| MA | 5% | Analizador Código Estático | Estáticas |
|  | | | |
| IA-1, IA-4 | 20% , 10% | Pruebas Ejecución | Sistema |
| IA-2, IA-4 | 20% , 10% | Pruebas Ejecución | Integración |
| IA-3 | 20% | Pruebas Ejecución | Unidad |
| MA | 100% | Ambiente Pruebas | Completo |
|  | | | |
| TOTAL | 100% | | |

**Nota: La tabla descrita contiene las correspondientes convenciones: [IA - Ingeniero Auto matizador] [MA – Maquina Nube].**

Las correspondientes fases estarán condicionadas a los momentos de iteración planificadas de la siguiente manera:

|  |  |
| --- | --- |
| Fase | Momento |
| Análisis y Diseño de Prueba | Planeación |
| Codificación y Automatización | Iteración |

* + 1. **Recursos Humanos:**

Para poder cumplir con los objetivos se requiere que el debido recurso humano cuente con los correspondientes skills y tiempo de experiencia:

* Programación en Framework JavaScript con un tiempo de mínimo 5 años.

Se cuentan con los recursos auto matizadores los cuales son 4 y cada uno con una experiencia de programación en framework JavaScript, se requieren usar en un 100% del total del tiempo durante el transcurso del proyecto.

* + 1. **Recursos Computacionales:**

Se tiene como recursos computacionales, maquina en la nube con disposición libre durante el transcurso de 8 semanas. Las cuales estarán distribuidas en análisis de código estático en la fase de construcción y el restante en pruebas automatizadas. Se desconocen la capacidad a nivel computacional, por lo cual se sugiere tener la disposición a medida que se lleven a cabo las pruebas de extender sus recursos. Se usará el recurso con mayor expertise para abarcar transversalmente los niveles de pruebas; sistema e integración apoyando al equipo en cualquier inquietud o bloqueante que surja durante la fase de construcción de la prueba.

* + 1. **Recursos Económicos para la contratación de servicios:**

**N/A**

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

**A continuación, se listan las técnicas, niveles, tipos de pruebas por versión procesada del sistema ghost.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Técnicas | Niveles | Tipos | Objetivo | Versión Ghost    (3.41.1), (3.3.0), (2.x) |
| Automatisar como usuario, esto atraves de APIs(**Cypress, Puppeteer, Kraken, Cucumber, Faker, Resemble** ). | Sistema | Funcional, positivo | 90% cobertura detección de defectos. |
| Verificar que varias unidades trabajen en completa armonía Ejemplo: Admin Api con Client App. APIs(Cypress) | Integración | Funcional, positivo, caja negra | 70% cobertura detección de defectos. |
| Verificar que las unidades individuales y aisladas funcionen. API(Jest) | Unidad | Funcional, positivo, caja negra | 100% cobertura detección de defectos. |
| API(Sonarlint) | Estáticas | Caja blanca | 100% cobertura detección de defectos. |

**Recordemos conceptos**

**Tipos:** definen el propósito de las pruebas

* **Funcional:** Evaluar los requerimientos funcionales de la aplicación.
* **Positivo:** buscan validar que el sistema funciona en forma esperada, es decir, los escenarios normales de los casos de uso.
* **Caja Negra:** Se enfocan en las salidas del sistema.

**Niveles:** Granularidad de las partes correspondientes a un sistema

* **Sistema:** El objetivo es probar el sistema como un todo.
* **Integración:** validar el funcionamiento de varios componentes que al ser probados no presente estados o comportamientos inesperados.
* **Unidad:** validar que la funcionalidad individual funcione.
* **Estáticas:** Verificar que el código fuente no contenga errores.

**Técnicas:** Forma como los eventos y las pruebas son ejecutados.

* **APIs:** funciones y procedimientos que se brindan para ser utilizados en un nivel de abstracción.

Enunciamos los correspondientes usos de pruebas en la estrategia:

|  |  |
| --- | --- |
| Nombre | Repositorio |
| Manuales | <https://github.com/deyvidm18/pruebas-automatizadas/tree/manuales> |
| Reconocimiento | <https://github.com/deyvidm18/pruebas-automatizadas/tree/reconocimiento/Monkey-cypress> |
| E2E | <https://github.com/deyvidm18/pruebas-automatizadas/tree/main> |
| VRT | <https://github.com/deyvidm18/pruebas-automatizadas/tree/pruebas-vrt> |
| Generación de Datos | <https://github.com/deyvidm18/pruebas-automatizadas/tree/semana7> |

Cada una de estas herramientas fue usada en el transcurso de la ejecución del proyecto.

Relacionamos el registro de incidencias por pruebas:

|  |  |
| --- | --- |
| **BUGS** | **Repositorio** |
| **RIP-01, CYPRESS, MANUAL, VRT, GENERACIÓN DE DATOS** | <https://github.com/deyvidm18/pruebas-automatizadas/issues?page=1&q=is%3Aissue+is%3Aopen> |

**Enlace Vídeo**

Link: [Estrategia de Pruebas](https://youtu.be/3yBy-3_YQuc)