Estrategia de Pruebas

1. Aplicación Bajo Pruebas
   1. **Nombre Aplicación:**

Ghost

* 1. **Versión:**

1.24.2

* 1. **Descripción:**

Ghost es plataforma de blogging de código abierto diseñada para simplificar la creación y publicación de contenido en línea. Con un enfoque en la escritura y la experiencia del autor, Ghost ofrece un editor de contenido minimalista que permite a los usuarios concentrarse en crear artículos y entradas de blog de manera efectiva. Además de su elegante editor, Ghost ofrece características como la organización de contenido en categorías y etiquetas, personalización de diseño mediante temas, programación de publica ciones, optimización SEO, integración con redes sociales y herramientas de análisis. También es posible habilitar opciones de membresía y suscripción para monetizar el contenido. Ghost se distingue por su velocidad, rendimiento y enfoque en la seguridad, lo que la convierte en una opción popular tanto para bloggers individuales como para sitios web más grandes.

* 1. **Funcionalidades Core**:

El trabajo se enfocará en las siguientes funcionalidades:

* + Funcionalidad de login
  + Funcionalidad de registrar un site
  + Funcionalidad de publicar, editar y borrar un post
  + Funcionalidad de programar el horario de publicación del post
  1. **Modelo de datos:**

En esta sección, se presenta el diagrama de base de datos de la aplicación Ghost. Este fue obtenido del archivo ghost\_local.db de la carpeta que contiene los archivos de instalación local de la aplicación. Fue diagramado en el IDE DataGrip con la herramienta JGraph diagrams.



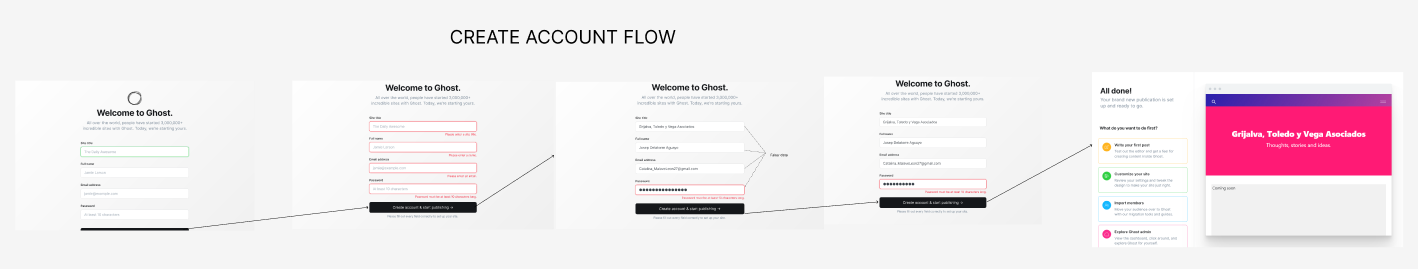
Link: <https://viewer.diagrams.net/?tags=%7B%7D&highlight=0000ff&edit=_blank&layers=1&nav=1&title=exported_from_idea.drawio#Uhttps%3A%2F%2Fdrive.google.com%2Fuc%3Fid%3D14ToEcujqWk6Qfj5GzFSATfOLvuiQmG1P%26export%3Ddownload>

|  |  |  |
| --- | --- | --- |
| Nombre del dato | Tipo | Cantidad de caracteres |
| email | varchar | 191 |
| password | varchar | 60 |
| name | varchar | 191 |
| website | varchar | 2000 |
| title | varchar | 2000 |
| plaintext | text |  |

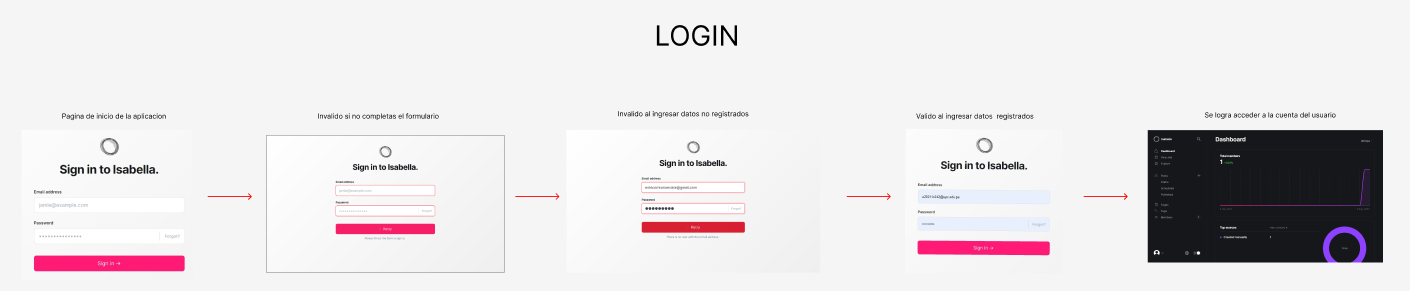
* 1. **Modelo de GUI:**

Un modelo GUI, o modelo de Interfaz Gráfica de Usuario es un sistema o representación visual de la estructura gráfica de una interfaz de usuario, útil para desarrolladores en el diseño de software. A continuación, se presenta el flujo de las funcionalidades principales escogidas en interacción con la interfaz gráfica.

Flujo de Create account Flow:



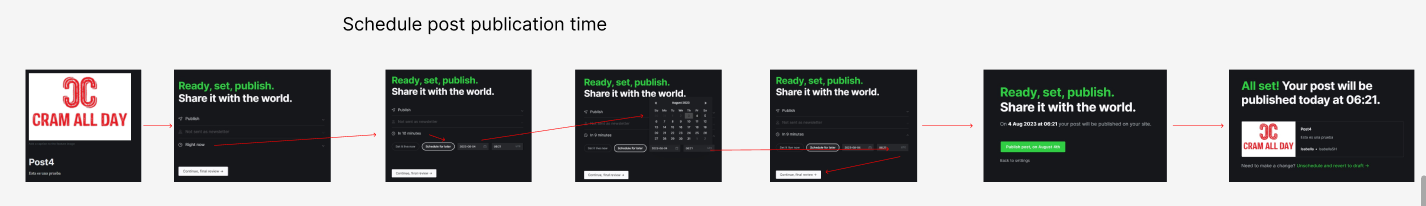
Flujo para iniciar sesión:



Flujo de crear post:



Fujo para programar la publicación de un post:



Link: <https://www.figma.com/file/s9s1zW8YBxp9P9Vi8UD3F2/DIAGRAMA-DE-GUI?type=design&node-id=0%3A1&mode=design&t=SATnE3pFtjbgkBFr-1>

1. **Contexto de la estrategia de pruebas**
   1. **Objetivos:**
   * Abordar pruebas de software a un sistema de gestión de contenido desde un enfoque funcional a partir de la validación de casos positivos y negativos.
   * Asegurar que las características esenciales del software funcionen de manera óptima y cumplan con los requerimientos establecidos
   1. **Duración de la iteración de pruebas:**

El equipo de trabajo se muestra conformado por cuatro integrantes. Estos estudiantes realizarán la iteración de pruebas en un intervalo de tres días. A continuación, se muestran los horarios de disponibilidad por día:

\* Ayasta Leonardo, Valery Ruth: 4h

\* Comettant Rubiños, Jessica Elizabeth:2h

\* Mendez Pastor, Brigitte Melody:2h

\* Soriano Huapaya, Isabella María:4h

En conclusión, se estima una duración de 36h persona para el total de pruebas y 30h maquina

* 1. **Presupuesto de pruebas:**

A continuación, se describirán los recursos con los que se dispone para realizar las pruebas y cómo se distribuirán.

* + 1. **Recursos Humanos**

El equipo está conformado por cuatro personas, a continuación, se detalla la experiencia previa de cada una:

**Ayasta Leonardo, Valery Ruth:** Tengo experiencia desarrollando aplicaciones web con Spring Boot y .NET Framework en el backend. También, domino frameworks front-end como Vue.js y Angular. En los proyectos he realizado pruebas unitarias como Junit y Cucumber. Además, he trabajado con varios gestores de base de datos como SQL Server, MySQL, Postgress y MongoDB. Mi tiempo disponible para el proceso de pruebas es de 20 horas a la semana (4h al día).

**Comettant Rubiños, Jessica Elizabeth:** Mi experiencia se basa en el uso de herramientas de diseño de aplicaciones web utilizadas en asignaturas exigidas por la malla curricular de mi carrera como frameworks de Angular y Vue, uso de lenguajes de enmarcado HTML, lenguaje de programación CSS, JavaScript, C# y Python. Respecto al manejo de gestores de bases de datos relacionales y no relacionales: uso de MongoDB y SQL Server. Finalmente, en relación a la documentación activa de casos de prueba, lenguaje Gherkin con la herramienta Cucumber en Visual Studio Code. La disponibilidad horaria otorgada al proceso de pruebas es de 10 horas semanales en segmentación de 2 horas diarias.

**Mendez Pastor, Brigitte Melody:** Tengo experiencia desarrollando aplicaciones web utilizando Spring Boot, .NET Framework, Laravel por la parte del backend; en cuando al frontend, domino los frameworks Vue.js y Angular. En cuanto a pruebas, tengo experiencia utilizando JUnit, Cucumber, XUnit, Specflow, PHP Unit y Selenium. Asimismo, he trabajado con gestores de bases de datos como SQL Server, MySQL y PostgreSQL. Mi tiempo disponible para el proceso de pruebas es de 10 horas a la semana (2h al día).

**Soriano Huapaya, Isabella María:** Mi experiencia previa la tengo en base a cursos llevados en la universidad en los cuales desarrolle aplicaciones web usando los frameworks de Vue y Angular para el enfoque Front-end y para el desarrollo del Back-end tengo experiencia usando Spring Boot y .NET Framework. Asimismo, he trabajo con MongoDB y SQL Server como gestores de base de datos. En base a estos proyectos pude aprender a redactar pruebas unitarias usando el lenguaje de Gherkin empleando Cucumber. El tiempo disponible con el que cuento para realizar las pruebas es de 4h diarias.

* + 1. **Recursos Computacionales**

La realización de pruebas fue realizada en el laboratorio de cómputo de la universidad. Esta contaba con las siguientes especificaciones técnicas:

|  |  |
| --- | --- |
| **Especificación** | **Valor** |
| Processor | Intel(R) Core(TM) i7-10700 CPU @ 2.90GHz 2.90 GHz |
| Installed RAM | 16.0 GB (15.8 GB usable) |
| Device ID | 1B10F39B-79BB-49EC-8ACD-41655F306AE5 |
| Product ID | 1B10F39B-79BB-49EC-8ACD-41655F306AE5 |
| System type | 64-bit operating system, x64-based processor |
| Edition | Windows 10 Enterprise |
| Version | 22H2 |
| OS build | 19045.2546 |
| Experience | Windows Feature Experience Pack 120.2212.4190.0 |

* + 1. **Recursos Económicos para la contratación de servicios/personal:**
* **Personal de pruebas:**

Se realiza la contratación de personal para ejecutar los casos de pruebas, generar los informes y resolver problemas:

* + - Ingeniero de Pruebas Junior (2 posiciones)

Duración: 1 semana

Costo estimado: S/. 2000

* + - Ingeniero de Calidad (2 posiciones)

Duración: 1 semana

Costo estimado: S/. 5000

Costo total estimado para los salarios del personal contratado: S/. 7000

* **Herramientas de pruebas y equipos**

Las APIs de automatización son de licencia gratuita y no se usarán servicios en la nube, pues se utilizará almacenamiento local.

Costo estimado por los equipos (hardware): S/. 1000

* **Contingencia:** para afrontar posibles imprevistos se incluirá un margen de contigencia del 10% sobre el presupuesto total:

Costo estimado del margen de contingencia: S/ 800

**Presupuesto Total para Recursos Económicos: S/8800**

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

Considerando los objetivos planteados para las pruebas, que incluyen la detección de defectos y la validación de las funcionalidades principales, se han seleccionado varias técnicas apropiadas. Se utilizarán tanto pruebas manuales como automatizadas para abordar diferentes aspectos de la calidad del software. Para las pruebas automatizadas, se emplearán las APIs de automatización Cypress, Playwright y Kraken, lo que permitirá una cobertura más amplia y eficiente de los casos de prueba.

En cuanto a los niveles de prueba, el enfoque se centrará en el nivel de aceptación. Aquí, el equipo de trabajo desempeñará el papel de usuarios finales en un contexto real, lo que nos brindará una perspectiva valiosa sobre cómo la aplicación se comporta desde la perspectiva del usuario y si cumple con las expectativas establecidas.

En relación con los tipos de pruebas seleccionados, se optará por las pruebas funcionales, ya que se enfocarán en probar las funcionalidades principales de la aplicación. Además, se aplicará el enfoque de caja negra, donde se evaluará el comportamiento de la aplicación sin conocer los detalles internos de su implementación. Se probarán casos positivos para validar que la aplicación funcione correctamente y produzca las salidas esperadas.

La combinación de pruebas manuales y automatizadas, junto con un enfoque en el nivel de aceptación y pruebas funcionales de caja negra con casos positivos, garantizará una evaluación integral de la calidad del software, detectando posibles defectos y asegurando que las funcionalidades esenciales estén correctamente implementadas. Todo esto contribuirá a la entrega de un producto confiable y robusto que cumpla con las expectativas del usuario final.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **DISTRIBUCIÓN DE ESFUERZO** | | | | |
| **Persona Encargada** | Jessica | Isabella | Valery | Brigitte |
| **Técnica** | Automatizada (Cypress) y manual | Automatizada (Playwright) y manual |  | Automatizada (Cypress) y manual |
| **Nivel** | Aceptación | Aceptación | Aceptación | Aceptación |
| **Tipo** | Funcional | Funcional | Funcional | Funcional |
| **Funcionalidad a probar** | Login  (Prueba automatizada)  Publicar un post  (Prueba manual) | Login  (Prueba automatizada)  Programar el horario de publicación del post  (Prueba manual) | Registro de usuarios (Prueba automatizada y prueba manual) | Crear un post (automatizada)  Eliminar o editar un post (Prueba Manual) |
| **Tiempo dedicado (h/día)** | 2 horas | 4 horas | 5 horas | 2 horas |
| **Presupuesto (S/.)** | S/250 | $250 | $333 | $200 |
| **Recursos computacionales individuales** | **Procesador:** Intel(R) Core(TM) i7-10700 CPU @ 2.90GHz 2.90 GHz  **RAM instalada:** 16.0 GB (15.0 GB usable)  **Tipo de sistema:**  Sistema operativo de 64 bits, procesador basado en x64  **Edición:**  Windows10 Enterprise  **Versión:** 22H2 | **Procesador:** Intel(R) Core(TM) i7-10700 CPU @ 2.90GHz 2.90 GHz  **RAM instalada:** 16.0 GB (15.0 GB usable)  **Tipo de sistema:**  Sistema operativo de 64 bits, procesador basado en x64  **Edición:**  Windows10 Enterprise  **Versión:** 22H2 | **Procesador:** Intel(R) Core(TM) i7-10700 CPU @ 2.90GHz 2.90 GHz  **RAM instalada:** 16.0 GB (15.0 GB usable)  **Tipo de sistema:**  Sistema operativo de 64 bits, procesador basado en x64  **Edición:**  Windows10 Enterprise  **Versión:** 22H2 | **Procesador:** Intel(R) Core(TM) i7-10700 CPU @ 2.90GHz 2.90 GHz  **RAM instalada:** 16.0 GB (15.0 GB usable)  **Tipo de sistema:**  Sistema operativo de 64 bits, procesador basado en x64  **Edición:**  Windows10 Enterprise  **Versión:** 22H2 |