

# Estrategia de Pruebas

## 1. Aplicación Bajo Pruebas

**1.1. Nombre Aplicación:** Ghost

**1.2. Versión:** 5.69.0

**1.3. Descripción:**

Ghost es una aplicación poderosa para que los creadores de nuevos medios publiquen, compartan y hagan crecer un negocio en torno a su contenido. Viene con herramientas modernas para crear un sitio web, crear blogs, enviar boletines y ofrecer suscripciones pagas a los miembros.

**1.4. Funcionalidades Core:**

- **Publicación.** Las publicaciones se crean dentro de Ghost-Admin utilizando el editor para determinar el contenido principal de su sitio. Dentro de ellos se encuentran todos los campos que puede esperar, como título, descripción, slug, metadatos, autores, etiquetas, etc.

Además, las publicaciones tienen campos de Inyección de código , lo que significa que puede registrar estilos adicionales, scripts u otro contenido para inyectar justo antes `</head>` o `</body>` en cualquier URL en particular donde lo desee.

✳ **Drafts.** Guarda los post editados sin publicar.

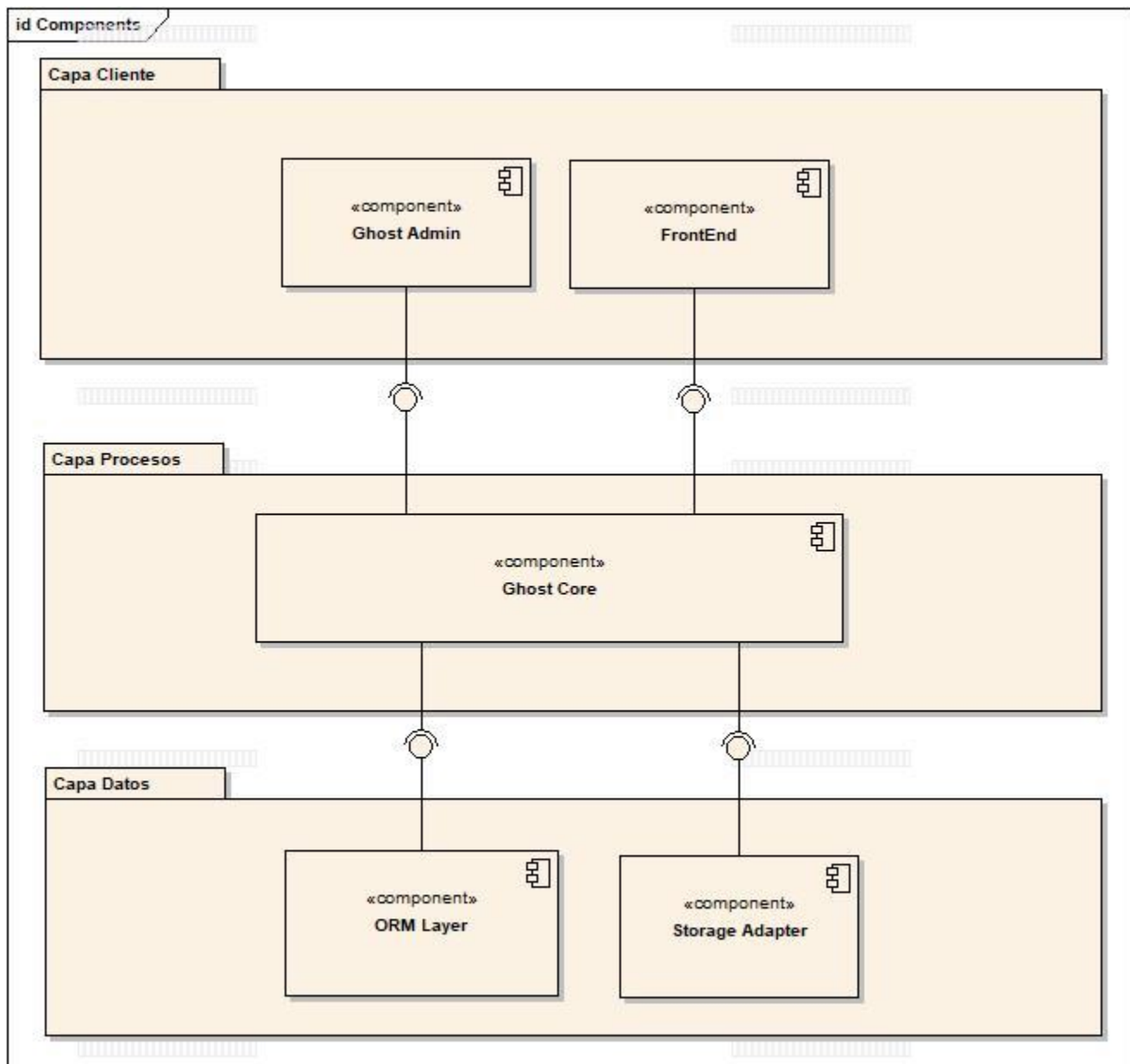
✳ **Scheduled.** Permite programar la publicación de los posts y a que privacidad tengan.

✳ **Membresías.** Cualquier usuario que desee ofrecer una forma para que su audiencia apoye su trabajo puede usar la función Miembros para compartir contenido, crear una audiencia y generar ingresos de un negocio de membresía.

✳ **Boletines por correo electrónico.** Los boletines por correo electrónico en Ghost se pueden programar y enviar a miembros gratuitos y de pago, o a un segmento de miembros gratuitos o de pago. Los boletines se entregan utilizando una hermosa plantilla HTML que está estandarizada para los clientes de correo electrónico más populares.

Los sitios Ghost tienen un solo boletín de forma predeterminada, pero se pueden crear y personalizar otros adicionales. Múltiples boletines le permiten personalizar el contenido para audiencias específicas y sus miembros pueden elegir qué contenido reciben.

## 1.5. Diagrama de Arquitectura:



- GHOST ADMIN.

Ghost-Admin es una interfaz de administración potente y opcional para editores. Facilita el flujo de trabajo editorial y ofrece un editor de alta calidad. Permite la gestión de contenido con acceso de lectura y escritura, brindando autenticación segura basada en roles.

- FRONTEND.

Ghost es independiente de cualquier interfaz en particular y su interfaz es opcional e intercambiable. Puede crear su propia interfaz completamente personalizada, especialmente útil si utiliza Ghost como servicio para aplicaciones móviles o nativas.

- CORE

Ghost es una API RESTful JSON que permite la creación, administración y recuperación de contenido de publicaciones de manera sencilla. Divide la API en Contenido y Administración, cada una con sus métodos de autenticación y herramientas para diversos casos de uso.

- ORM LAYER.

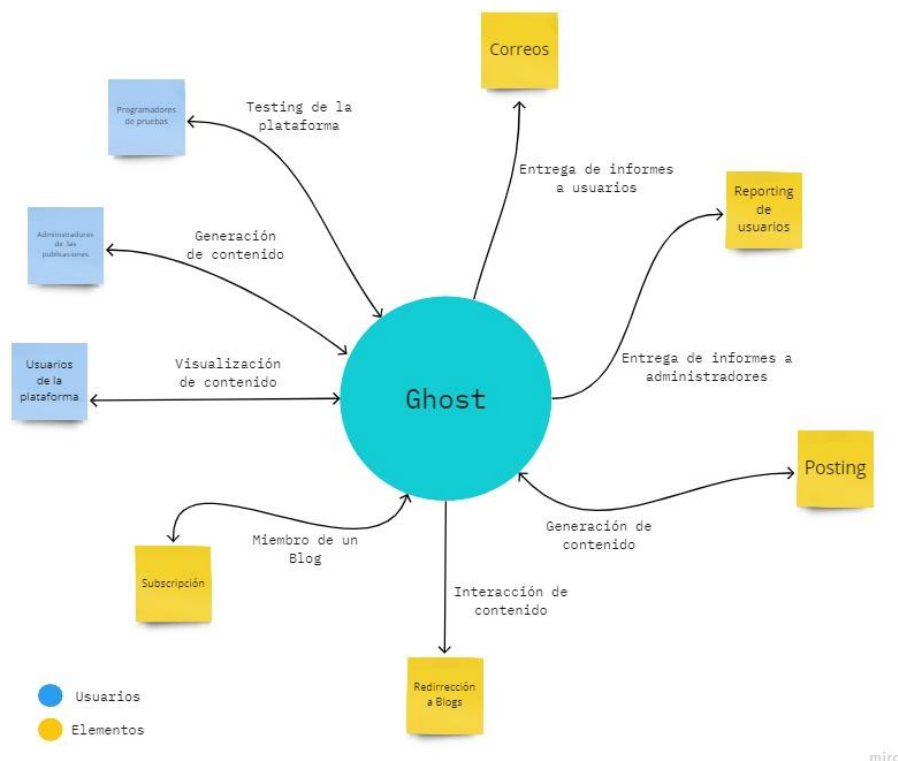
Ghost utiliza la capa ORM Bookshelf.js por defecto y es compatible con una variedad de bases de datos, siendo SQLite3 la predeterminada en desarrollo y MySQL recomendada para producción. STORAGE ADAPTER.

- Storage Adapter

Ghost usa almacenamiento local de manera predeterminada, pero los usuarios tienen la opción de emplear adaptadores de almacenamiento personalizados que permiten tener un sistema de archivos completamente externo. Además, existen numerosos adaptadores de almacenamiento prefabricados listos para ser utilizados en Ghost.

<https://ghost.org/docs/architecture/>

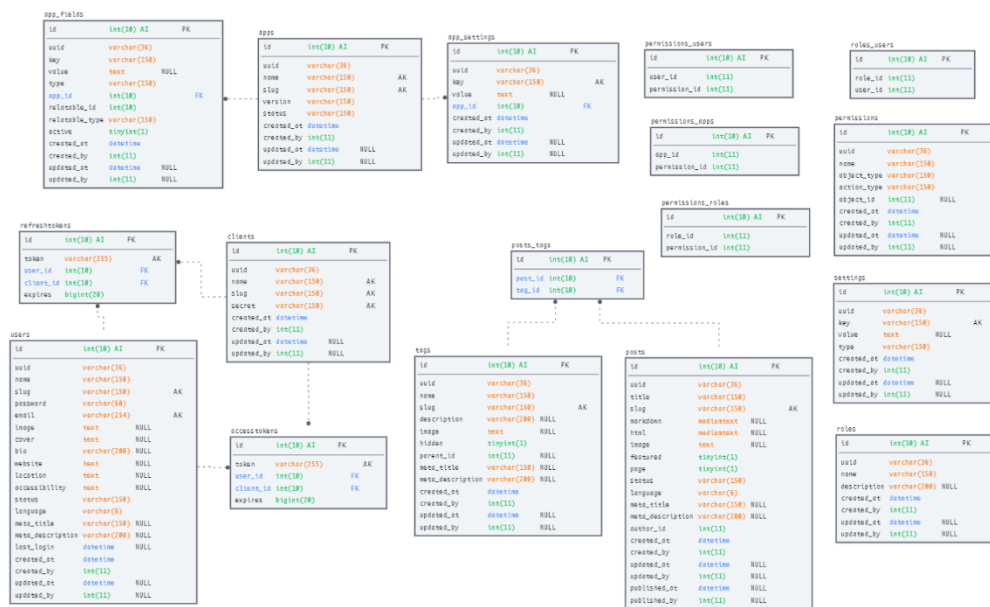
## 1.6. Diagrama de Contexto:



## 1.7. Modelo de Datos:

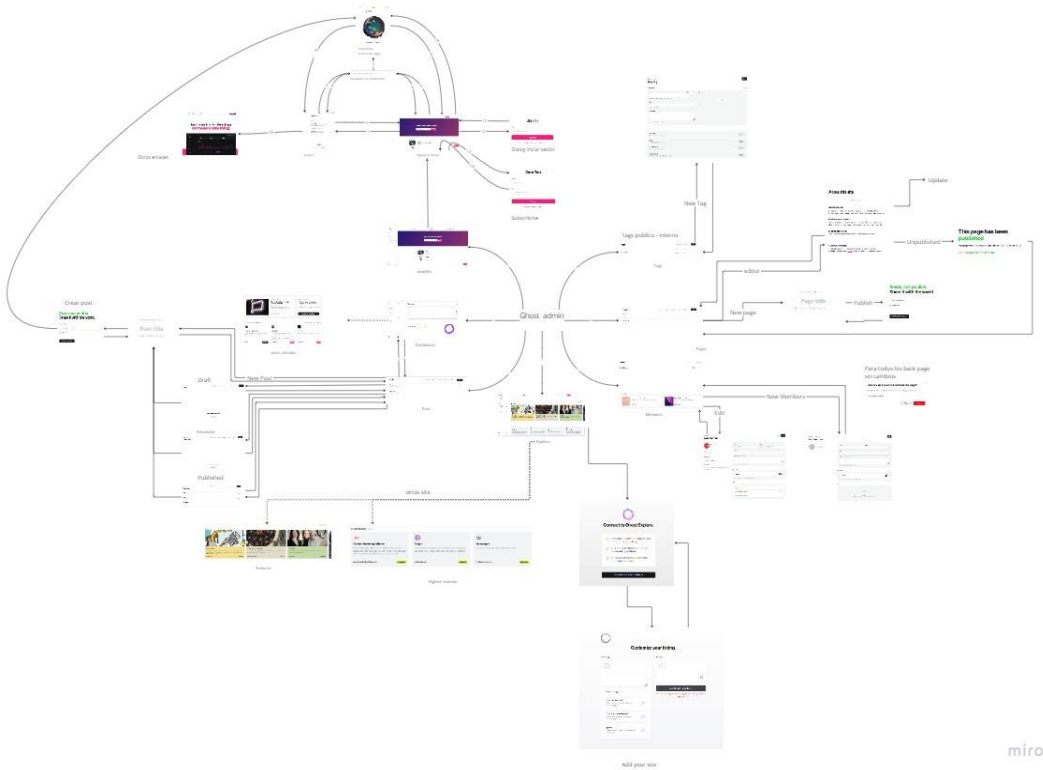
Fuente: <https://github.com/TryGhost/Ghost-Config/blob/master/ghost-versions/0.6.4/schema.sql>

Img: <https://ibb.co/XtrZitk>



## 1.8 Modelo de GUI:

[https://drive.google.com/file/d/1latEqXxOnG8CPAtQ5etrpddbg\\_2xN1Y\\_/view?usp=sharing](https://drive.google.com/file/d/1latEqXxOnG8CPAtQ5etrpddbg_2xN1Y_/view?usp=sharing)



## 1. Contexto de la estrategia de pruebas

### 1.1. Objetivos:

- Realizar pruebas funcionales para comprobar los atributos de calidad establecidos por la empresa TSDC en el producto GHOST
- Desarrollar pruebas con un enfoque que alcance a detectar los defectos que tiene el producto GHOST.

### 1.2. Duración de la iteración de pruebas:

Para la iteración de pruebas se propone realizar una labor de 7 días los cuales corresponden a 4 horas diarias por persona y 400 horas por maquinaria para realizar las pruebas manuales y automatizadas. Estas labores iniciaran a partir del lunes 30 de octubre y finalizaran el lunes 6 de noviembre para las pruebas manuales y la automatización de las pruebas, y la realización de las pruebas automatizadas iniciara el 1 de noviembre y finalizara el 6 de noviembre.

### 1.3. Presupuesto de pruebas:

### 1.3.1. Recursos Humanos

Se contrata dos ingenieros Senior por medio del outsourcing y cada uno de ellos tiene un costo por hora de 7 dólares y sus funciones serán configurar máquinas virtuales, realizar pruebas, estudio y análisis de resultados.

Contamos con un ingeniero Junior para labora 30 horas semanales para realizar pruebas manuales y automatizadas.

### 1.3.2. Recursos Computacionales

Máquina virtual para hacer las pruebas

Fuente: <https://aws.amazon.com/es/ec2/pricing/on-demand/>

Nombre de la Instancia: m6g.2xlarge

Sistema Operativo: Linux

VCPU: 8

Memoria: 32 Gb

Almacenamiento: Solo EBS

Rendimiento Red: Hasta 10 gigabits

Precio: \$ 0,308/Hora

Utilización: Se instalará el CMS GHOST conjuntamente con Cypress y Playwright para las pruebas de componentes, integración y e2e debido a que el equipo tiene conocimiento base de estas herramientas

Estación de trabajo para ingeniero Junior:

Sistema Operativo: Linux

Pantalla: 19"

CPU: Core i7 9na gen

Memoria: 16 GiB (2\*8gb)

Almacenamiento: Ssd Nvme M 2 de 250 GB

Utilización: Utilizaremos las estaciones de trabajo existentes para escribir los scripts de prueba y conectarnos con las máquinas virtuales en Amazon AWS

### 1.3.3. Recursos Económicos para la contratación de servicios/personal:

TRM= 1 dólar = \$4.108,50 Pesos

Ingeniero senior 7 dolares/horas

Fuente: <https://co.talent.com/salary?job=senior+qa>

400 horas/máquina en AWS

Costo del salario de automatizador senior en dolares \$ 1.277.83 mes / \$7 hora

Plantilla elaborada por

**THE SW DESIGN LAB**

Costo por horas laboradas de un ingeniero senior :28 hrs \* \$7 = \$196 dólares

Costo por dos ingenieros senior = \$196 \*2= \$392 dólares

Horas de utilización de máquina virtual en Amazon AWS = 336 hrs

Costo total Amazon AWS EC2: \$ 0,308 X 336= \$ 103.48 dólares

Costo Total de gastos = \$ 103.48 + \$392 = \$495.48 dólares

Fuente: <https://aws.amazon.com/es/ec2/pricing/on-demand/>

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

| Nivel       | Tipo        | Tecnica      | Objetivo  |
|-------------|-------------|--------------|---|
| componente  | funcionales | automatizada | <ul style="list-style-type: none"><li>- Comprobar el correcto funcionamiento de las funcionalidades.</li><li>- Detectar fallos o bugs en las funcionalidades implementadas.</li></ul> |
| componente  | caja negra  | automatizada | <ul style="list-style-type: none"><li>- Comprobar el correcto funcionamiento de las funcionalidades.</li><li>- Detectar fallos o bugs en las funcionalidades implementadas.</li></ul> |
| componente  | caja negra  | manual       | <ul style="list-style-type: none"><li>- Comprobar el correcto funcionamiento de las funcionalidades.</li><li>- Detectar fallos o bugs en las funcionalidades implementadas.</li></ul> |
| componente  | positivas   | automatizada | <ul style="list-style-type: none"><li>- Comprobar el correcto funcionamiento de las funcionalidades.</li><li>- Detectar fallos o bugs en las funcionalidades implementadas.</li></ul> |
| componente  | negativas   | automatizada | <ul style="list-style-type: none"><li>- Comprobar el correcto funcionamiento de las funcionalidades.</li><li>- Detectar fallos o bugs en las funcionalidades implementadas.</li></ul> |
| componente  | negativas   | automatizada | <ul style="list-style-type: none"><li>- Comprobar el correcto funcionamiento de las funcionalidades.</li><li>- Detectar fallos o bugs en las funcionalidades implementadas.</li></ul> |
| integracion | funcionales | automatizada | <ul style="list-style-type: none"><li>- Comprobar el correcto funcionamiento de las funcionalidades.</li><li>- Detectar fallos o bugs en las funcionalidades implementadas.</li></ul> |
| integracion | caja negra  | manual       | <ul style="list-style-type: none"><li>- Comprobar el correcto funcionamiento de las funcionalidades.</li></ul>  |

|               |             |              |  |
|---------------|-------------|--------------|--|
|               |             |              | - Detectar fallos o bugs en las funcionalidades implementadas.   |
| integracion   | caja negra  | automatizada | - Comprobar el correcto funcionamiento de las funcionalidades.<br>- Detectar fallos o bugs en las funcionalidades implementadas. |
| integracion   | positivas   | automatizada | - Comprobar el correcto funcionamiento de las funcionalidades.<br>- Detectar fallos o bugs en las funcionalidades implementadas. |
| integracion   | negativas   | automatizada | - Comprobar el correcto funcionamiento de las funcionalidades.<br>- Detectar fallos o bugs en las funcionalidades implementadas. |
| e2e           | funcionales | automatizada | - Comprobar el correcto funcionamiento de las funcionalidades.<br>- Detectar fallos o bugs en las funcionalidades implementadas. |
| e2e           | caja negra  | automatizada | - Comprobar el correcto funcionamiento de las funcionalidades.<br>- Detectar fallos o bugs en las funcionalidades implementadas. |
| e2e           | positivas   | automatizada | - Comprobar el correcto funcionamiento de las funcionalidades.<br>- Detectar fallos o bugs en las funcionalidades implementadas. |
| exploratorias | funcionales | manual       | - Comprobar el correcto funcionamiento de las funcionalidades.<br>- Detectar fallos o bugs en las funcionalidades implementadas. |
| exploratorias | caja negra  | manual       | - Comprobar el correcto funcionamiento de las funcionalidades.<br>- Detectar fallos o bugs en las funcionalidades implementadas. |
| exploratorias | positivas   | manual       | - Comprobar el correcto funcionamiento de las funcionalidades.<br>- Detectar fallos o bugs en las funcionalidades implementadas. |
| exploratorias | funcionales | automatizada | - Comprobar el correcto funcionamiento de las funcionalidades.<br>- Detectar fallos o bugs en las funcionalidades implementadas. |



|               |            |              |  |
|---------------|------------|--------------|--|
| exploratorias | caja negra | automatizada | - Comprobar el correcto funcionamiento de las funcionalidades.<br>- Detectar fallos o bugs en las funcionalidades implementadas. |
| exploratorias | positivas  | automatizada | - Comprobar el correcto funcionamiento de las funcionalidades.<br>- Detectar fallos o bugs en las funcionalidades implementadas. |
| exploratorias | negativas  | automatizada | - Comprobar el correcto funcionamiento de las funcionalidades.<br>- Detectar fallos o bugs en las funcionalidades implementadas. |

\* Pruebas de Unidad: Se van a obviar debido a que estamos probando un producto terminado, y las pruebas unitarias ya se realizaron en la fase de desarrollo del software.

\*\* Se deciden realizar algunas pruebas de forma manual por la complejidad de implementar su automatización, esto debido a la poca experiencia del automatizador junior.

\* \* \* La distribución de pruebas utilizada es la de piramide de automatización esto debido a la cantidad de pruebas utilizadas en cada nivel y para obtener un mayor rendimiento de los recursos que tenemos a nuestra disposición (automatizadores junior y equipos en Amazon AWS):

|               |    |
|---------------|----|
| exploratorias | 3  |
| e2e           | 2  |
| integración   | 4  |
| componente    | 11 |

## 1.5. Distribución de Esfuerzo

| ingenieros/dias | 1      | 2      | 3      | 4      | 5      | 6      | 7      | total   | Del primer dia  |   | De los dias 2 al 7 |  |          |
|-----------------|--------|--------|--------|--------|--------|--------|--------|---------|---|---|--------------------|--|----------|
| Ing. Senior 1   | 4 hrs  | 4 hrs  | 4 hrs  | 4 hrs  | 4 hrs  | 4 hrs  | 4 hrs  | 28 hrs  | *1 hora analizando Pruebas ingeniero Junior<br>*3h configuracion de maquina | *2h configuracion maquina<br>*1h Analizando resultados de maquina | Automatizadas      |  |          |
| Ing. Senior 2   | 4 hrs  | 4 hrs  | 4 hrs  | 4 hrs  | 4 hrs  | 4 hrs  | 4 hrs  | 28 hrs  |   |   |                    |  |          |
| Ing. Junior     | 4 hrs  | 4 hrs  | 4 hrs  | 4 hrs  | 4 hrs  | 4 hrs  | 4 hrs  | 28 hrs  |   |   |                    |  |          |
|                 |        |        |        |        |        |        |        |         | Pruebas componentes   |   |                    |  | Manuales |
| Machine/dias    | 1      | 2      | 3      | 4      | 5      | 6      | 7      | total   |   |   |                    |  |          |
| machine 1       | 24 hrs | 24 hrs | 24 hrs | 24 hrs | 24 hrs | 24 hrs | 24 hrs | 168 hrs |   |   |                    |  |          |
| machine 2       | 24 hrs | 24 hrs | 24 hrs | 24 hrs | 24 hrs | 24 hrs | 24 hrs | 168 hrs |   |   |                    |  |          |
| machine 3       | 4 hrs  | 4 hrs  | 4 hrs  | 4 hrs  | 4 hrs  | 4 hrs  | 4 hrs  | 28 hrs  |   |   |                    |  |          |

Listado de Pruebas:

| ID | Descripcion de la prueba | Tipo | Tecnica |
|----|--------------------------|------|---------|
|----|--------------------------|------|---------|

|       |  |              |              |
|-------|--|--------------|--------------|
| PC1   | suscribirse al sitio                                   | componente   | automatizada |
| PC2   | crear post   | componente   | automatizada |
| PC3   | publicar post  | componente   | automatizada |
| PC4   | cronogramar publicacion de post                        | componente   | automatizada |
| PC5   | listar posts   | componente   | automatizada |
| PC6   | listar pages   | componente   | automatizada |
| PC7   | crear page   | componente   | automatizada |
| PC8   | login de usuario administrador                         | componente   | automatizada |
| PC9   | modificar titulo del sitio                             | componente   | automatizada |
| PC10  | modificar tema del sitio                               | componente   | automatizada |
| PC11  | modificar barra de navegacion                          | componente   | automatizada |
| PI1   | Logear y suscribirse                                   | integracion  | automatizada |
| PI2   | crear y publicar post                                  | integracion  | automatizada |
| PI3   | crear y cronogramar publicacion post                   | integracion  | automatizada |
| PI4   | Crear usuario y loguearse                              | integracion  | automatizada |
| PE2E1 | crear usuario admin, crear post, publicar post, logout | e2e          | automatizada |
| PE2E2 | crear usuario admin, cambiar tema del sitio, logout    | e2e          | automatizada |
| PEX1  | Creacion y publicacion de posts.                       | exploratoria | manual       |
| PEX2  | exploracion con ripper                                 | exploratoria | automatizada |
| PEX3  | exploracion con monkey                                 | exploratoria | automatizada |

| Equipo    | 30 Oct. | 31 Oct.  | 1 Nov.  | 2 Nov. | 3 Nov. | 4 Nov.  | 5 Nov.   |
|-----------|---------|----------|---------|--------|--------|---------|----------|
| S1        | CM+VR   | PE2E1+VR | PC1+VR  | PC4+VR | PC7+VR | PC10+VR | PI3+PC19 |
| S2        | CM+VR   | PE2E2+VR | PC2+VR  | PC5+VR | PC8+VR | PC11+VR | PI4+VR   |
| J1        | PC3     | PC6+PI1  | PC9+PI2 | PEX1   | PEX2   | PEX3    | VR       |
| Maquina 1 | CM      | EAP      | EAP     | EAP    | EAP    | EAP     | EAP      |
| Maquina 2 | CM      | EAP      | EAP     | EAP    | EAP    | EAP     | EAP      |
| Maquina 3 | CM      | EAP      | EAP     | EAP    | EAP    | EAP     | EAP      |

\* Convenciones:

|     |                                 |
|-----|---------------------------------|
| CM  | Configuración Maquina           |
| PM  | Pruebas Manuales                |
| EAP | Ejecución Pruebas Automatizadas |
| VR  | Validar Resultados              |

VIDEO: <https://www.youtube.com/watch?v=luEl2YOXOpY>

Joining from Daniela Báez Rincón, Emerson Chaparro Ampa, Oscar Iván Manco Racines

**Pruebas Automatizadas Presupuesto 2**

Pruebas automatizadas de software

Cristian Arnulfo Arias Vargas  
Daniela Báez Rincón  
Emerson Chaparro Ampa  
Oscar Iván Manco Racines

**MISO**  
Maestría en Ingeniería de Software

4 Attendees in total

Salir