

# QA STATION

PRESENTACIÓN FINAL  
CAPSTONE

# INTEGRANTES DEL EQUIPO



**Esteban Díaz**

Desarrollador de  
Software



**Diego Gieminiani**

Desarrollador de  
Automatización



**Marian Moreno**

Scrum Master

# PROBLEMA

La falta de estandarización en los procesos de QA en equipos pequeños y medianos genera productos con errores y retrasos en el ciclo de desarrollo, dificultando la entrega eficiente de resultados. .



# PROPUESTA DE SOLUCIÓN

QA Station aborda este problema con una plataforma que simplifica la automatización de procesos de QA, permitiendo generar y ejecutar casos de prueba y pruebas funcionales sin necesidad de conocimientos avanzados en herramientas complejas.

# OBJETIVO GENERAL

Resolver la dificultad que enfrentan equipos pequeños y profesionales en formación al ejecutar y gestionar pruebas de calidad de software (QA) de manera eficiente y accesible, asegurando la calidad del software.

# OBJETIVOS ESPECIFICOS

01

Facilitar la creación de casos de prueba para usuarios con poca experiencia técnica, aprovechando la inteligencia artificial (IA) para generar automáticamente estos casos a partir de una URL.

02

Simplificar la ejecución de pruebas funcionales automatizando la ejecución de los tests, utilizando herramientas como Selenium.

03

Automatizar el flujo completo de pruebas funcionales desde la generación de casos de prueba hasta su ejecución, permitiendo que todo el proceso se realice sin intervención manual, a partir de una simple URL de la página web.

# ALCANCE DEL PROYECTO

01

Generación  
automática de casos  
de prueba con  
inteligencia artificial

02

Creación de pruebas  
funcionales con  
inteligencia artificial

03

Ejecución de pruebas  
funcionales sin  
programación

04

Resultados del estado  
de pruebas  
funcionales

# LIMITACIONES DEL PROYECTO

01

La aplicación está diseñada exclusivamente para realizar pruebas en la página web.

02

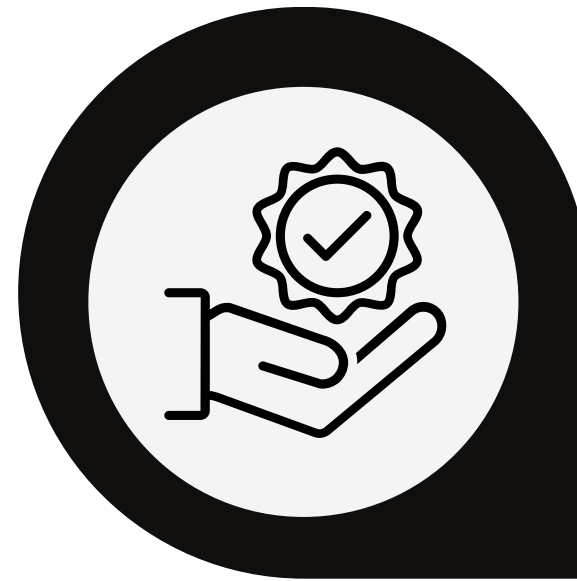
Actualmente, su funcionamiento se encuentra limitado al uso de Python con Selenium



# COMPETENCIAS DE LA CARRERA

## PRUEBAS DE CALIDAD

- Ejecución de pruebas.
- Validación de sistema.
- Garantizar cumplimientos.
- Buenas prácticas.



## MODELO DE DATOS

- Diseñar modelos de datos.
- Consistencia, integridad y escalabilidad.



## GESTIÓN DE PROYECTOS

- Asegurar entrega a tiempo.
- Planificación de actividades.
- Decisiones estratégicas.
- Cumplimiento de estándares de la empresa.



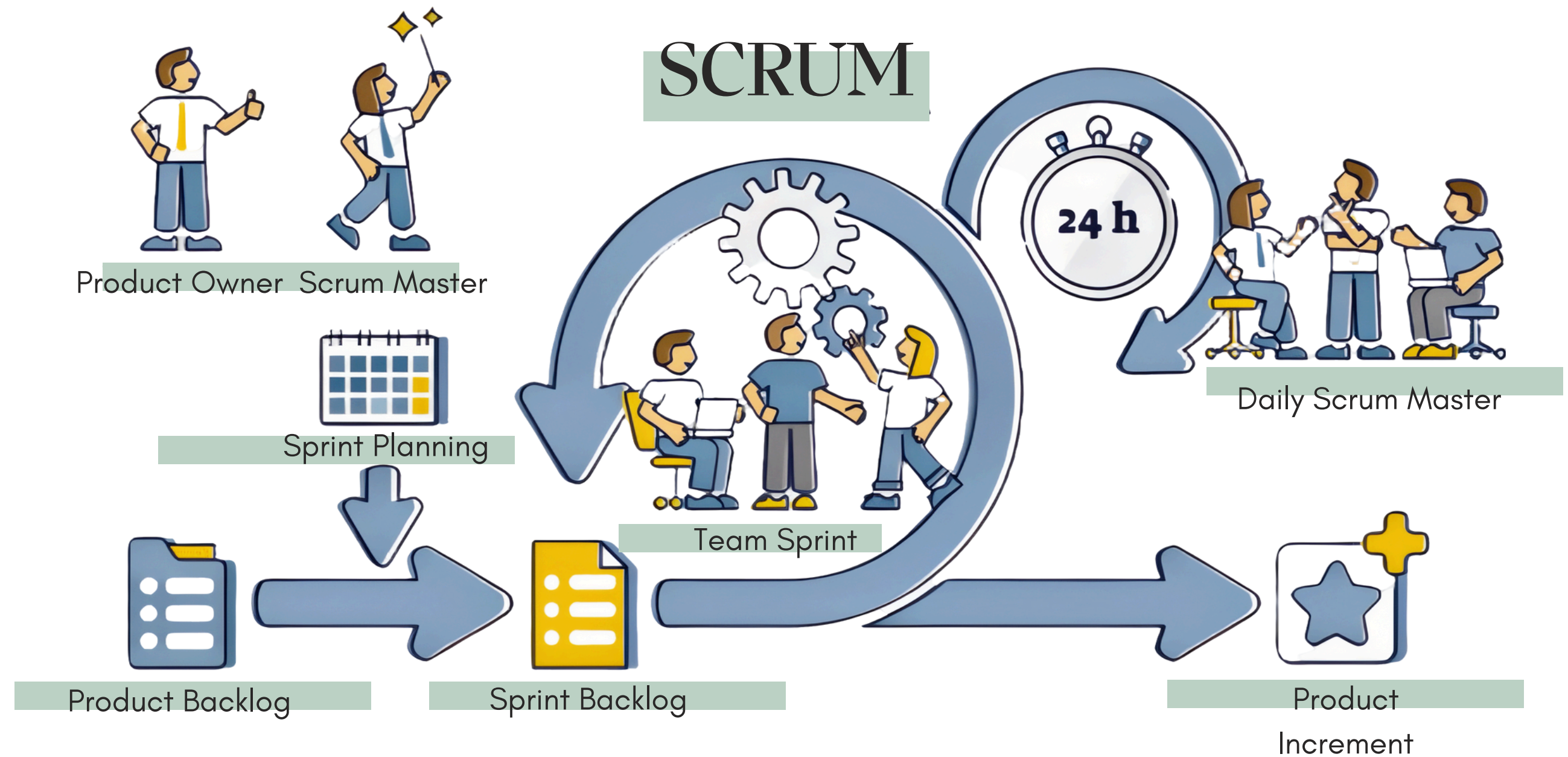
## DESARROLLO DE SOFTWARE

- Construir soluciones sostenibles.
- Asegurar el desarrollo de las diferentes funcionalidades y logro de objetivos.



# METODOLOGÍA DE TRABAJO

- Garantizar productos de calidad.
- Organización de equipo,
- Adaptable.

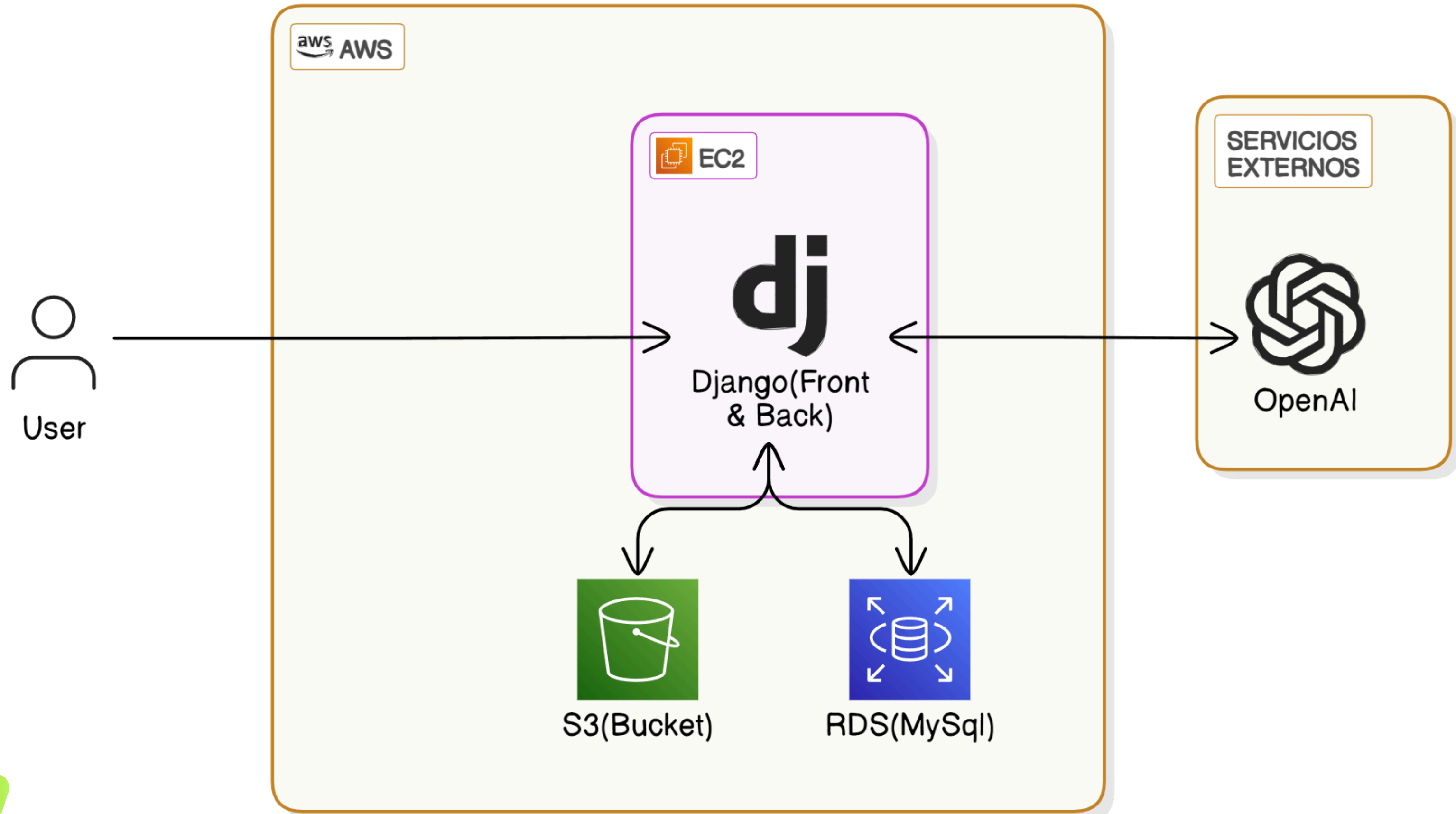


# CRONOGRAMA PARA EL DESARROLLO

Implementación de IA y automatización
Conectarse con OpenAI y ejecutar prompt
Ejecución de pruebas con elementos identificados
Generación automática de casos de prueba
Refinar prompts de OpenAI
Ejecución y personalización de pruebas
Desarrollar y validar pruebas de concepto
Ejecutar pruebas automatizadas
Guardar y reutilizar pruebas
Migrar modelo a base de datos a Django
Registro y gestion de usuarios
Registro de usuario
Autenticación de usuario
Validación de seguridad de contraseña
Restablecimiento de contraseña
Cierre de sesión
Opción para cerrar sesión
Redirección luego de cerrar sesión

[illegible]

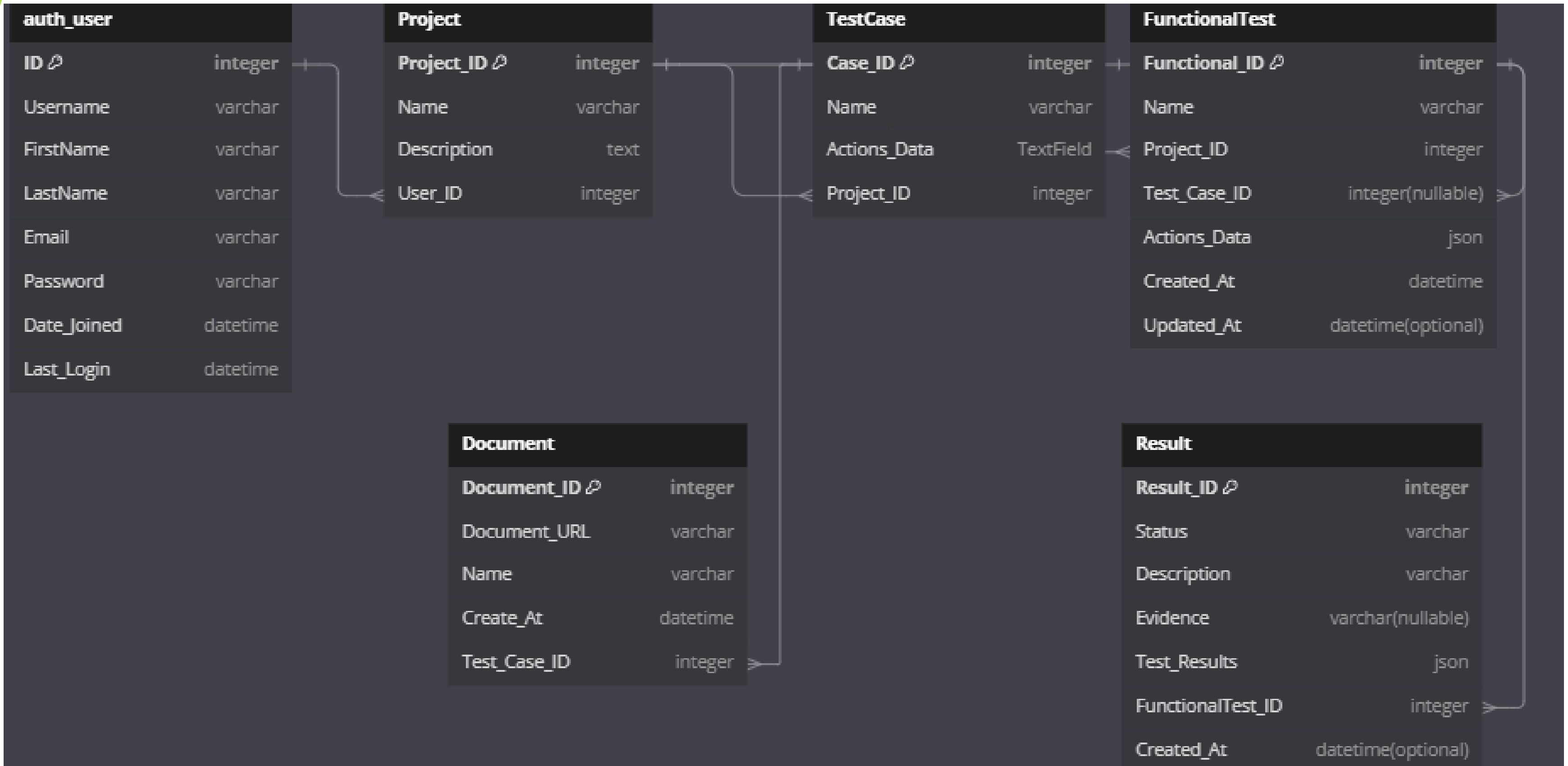
# ARQUITECTURA DE SOFTWARE



# TECNOLOGÍAS UTILIZADAS



# MODELO DE DATOS





# PROMPTS UTILIZADOS

# GENERACIÓN DE CASOS DE PRUEBA

URL



- Formato de salida
- Casos claros, separados y estructurados
- No incluir conclusiones, recomendaciones ni información adicional



- URL:
- ID:
- NOMBRE:
- PASO A PASO:
- RESULTADO ESPERADO:



# EJECUCIÓN DE PRUEBAS

- Casos de prueba generados con IA
- HTML de la pagina web seleccionada



- Formato de salida (muy específico)
- Ejemplos de salida
- Casos separados
- Valores no pueden ser nulos o vacíos
- No incluir explicaciones



- action:
- element\_type
- css\_selector:
- value:
- input\_value:



# **DEMOSTRACIÓN DEL PROYECTO**

# RESULTADOS OBTENIDOS

Inteligencia artificial para generación de casos de prueba.

Ejecución de pruebas automatizadas.

Usuarios.

Inicio de sesión y cierre de sesión.

# OBSTÁCULOS

- Inteligencia Artificial
- Tiempo
- Complejidad de la Automatización

**¡GRACIAS!**

**DuocUC** 