



UNIVERSIDAD AUTONOMA DE SINALOA

ADMINISTRACIÓN DE PROYECTOS II

GRUPO 4-01 INGENIERIA DE SOFTWARE

AVANCE DE PROYECTO 2

PROFESOR:
JUAN CARLOS GUZMAN PRECIADO

ALUMNOS:
BECERRA VEGA LUIS ENRIQUE
ESCALANTE SIANUQUI IRVING JARED

FECHA:
07/11/2025

Indice

GESTION DE ALCANCE	4
Enunciado:.....	4
Entregables:.....	4
Premisas:.....	4
Restricciones:.....	4
EDT/WBS del producto	5
EDT/WBS de actividades	5
Diccionario de EDT:	5
Verificación de alcance:.....	6
Control de alcance:.....	6
Gestión de Tiempo.....	7
Línea Base del Cronograma	10
Gestión de Costos	12
Línea Base de Costos.....	12
Gestión de Riegos.....	12
Lista de Riegos Identificados.....	12
Lista de respuestas potenciales	13
Gestión de Recursos Humanos.....	15
Organigrama del proyecto	15
Matriz de roles y responsabilidades	15
Gestión de Comunicaciones.....	16
Matriz de las comunicaciones	16
Herramientas de comunicación válidas para el proyecto	17
Gestión de Adquisiciones	17
Hardware	17
Software	18
Seguimiento del proyecto en Trello.....	20
Anexos	20
Pruebas no funcionales	20

GESTION DE ALCANCE

Enunciado:

Desarrollo de una página web para una clínica veterinaria que ofrezca funcionalidades de parte del cliente como para la administración. Los clientes podrán agendar citas, obtener recomendaciones de cuidados y poder descargar las cartillas de las mascotas en formato PDF, mientras que de parte de la administración se permitirá consultar el historial médico de las mascotas, obtener un resumen por IA de su historial médico, además de obtener una estadística de las mascotas atendidas

Entregables:

- Mascotas Felices: Pagina web mediante el cual los clientes podran agendar citas con la veterinaria, consultar cartillas y obtener recomendaciones, mientras que el personal podrá consultar el historial de las mascotas y las respectivas estadísticas de la veterinaria
- La estructura de Base de Datos con MySQL
- Documentación del proyecto

Premisas:

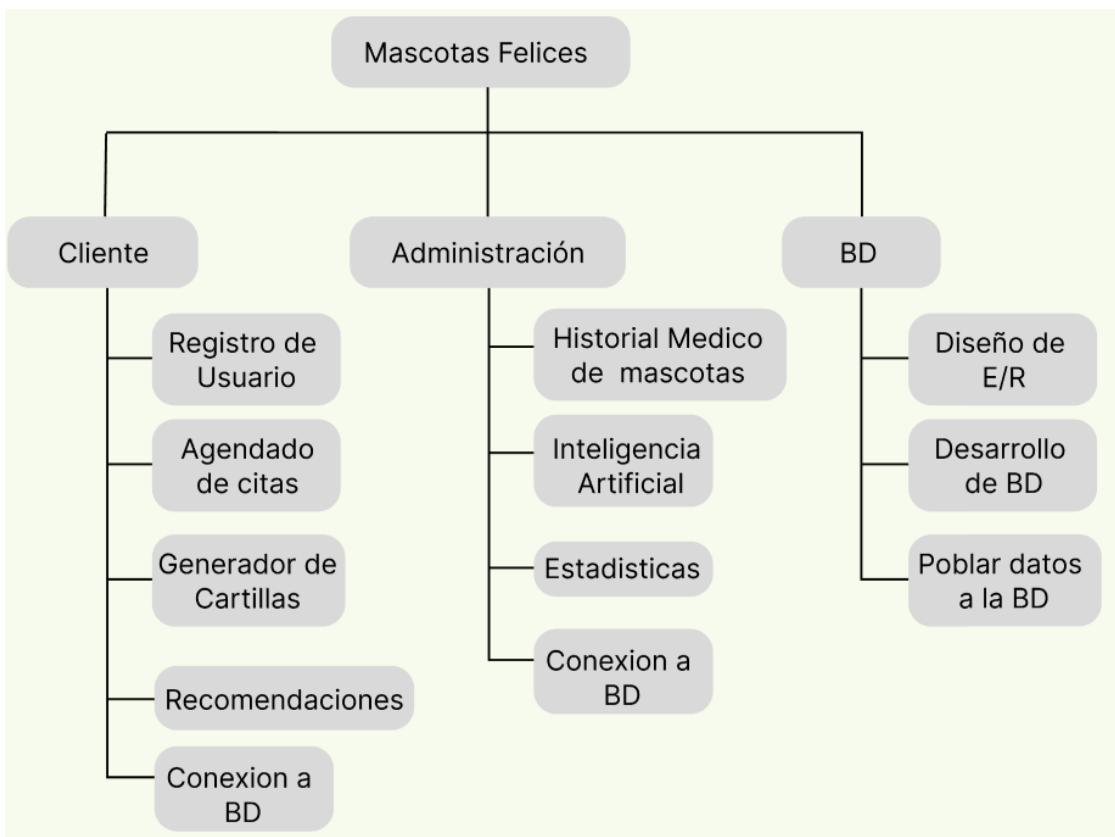
Las ideas que se consideran durante el desarrollo de la web de Mascotas Felices son:

- Se requerirá el uso obligatorio de internet tanto para el acceso a la aplicación como para el funcionamiento de los servicios.
- Contará con una base de datos donde se almacenarán repoblada con los datos de la veterinaria.

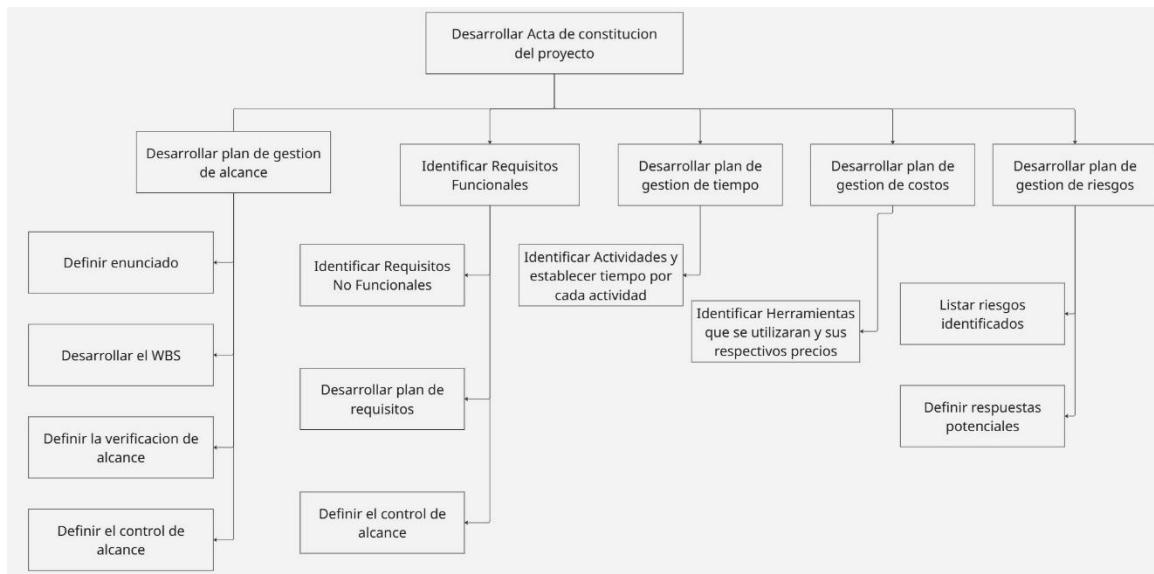
Restricciones:

- Cada tipo de usuario tendrá acceso a sus respectivos módulos.
- La aplicación web dependerá enteramente del uso de internet.
- Los datos de los clientes y sus mascotas se mantendrán con su estricto cuidado para que solo personal autorizado acceda a él.
- La entrega de los avances con las fechas establecidas por el profesor.

EDT/WBS del producto



EDT/WBS de actividades



Diccionario de EDT:

- **Mascotas Felices:** Nombre con el cual nos referiremos a nuestra pagina web

- **Cliente:** Subsistema del cliente
- **Administración:** Subsistema de administración
- **Registro de usuario:** Modulo del lado del cliente que permite el registro de usuarios al sistema
- **Agendado de citas:** Modulo del lado del cliente que permite agendar citas
- **Generador de cartillas:** Modulo de parte del cliente que permite generar de cartillas mediante una consulta
- **Recomendaciones:** Modulo de parte del cliente de recomendaciones de cuidados para las mascotas
- **Historial medico de mascotas:** Modulo de parte de administracion en el que se muestra el historial médico de las mascotas
- **Inteligencia Artificial:** Analiza el historial medico para generar resúmenes
- **Estadísticas:** Modulo de parte de la administración para mostrar las estadísticas de las mascotas atendidas
- **Diseño E/R:** Diagrama entidad relación de la base de datos
- **Desarrollo de la BD:** Desarrollo de la base de datos por medio de un sistema gestor
- **Poblar datos de la BD:** Es la información Contenida en la Base de Datos de Pruebas.

Verificación de alcance:

Se han entregado documentación que contiene la definición del proyecto junto con los módulos y funcionalidades a desarrollar. Aquella documentación fue elaborada a partir de discusiones entre los miembros del equipo y revisiones del profesor, con el objetivo de que el proyecto pueda satisfacer los estándares establecidos.

Control de alcance:

Con tal de mantener controlado este proyecto ante posibles cambios en los requisitos del sistema, se realizarán los siguientes procesos:

- Todos los miembros del equipo serán informados sobre cualquier cambio solicitado.
- Entre el equipo se evaluará el impacto que provocará realizar este cambio en cuestiones de tiempo.
- Una vez se discute sobre el cambio, se tomará la decisión de implementarse, alternativas mejores a los cambios propuestos o solamente rechazar el cambio.
- Los cambios también serán mencionados en la documentación.

Gestión de Tiempo

ID	TAREA	Septiembre				Octubre				Noviembre				Diciembre			
		1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
	Administración (2 Semanas)																
TOO1	Planificar el Proyecto (Alcance, Costo y Tiempo)	●															
TOO2	Recopilar los requerimientos funcionales y no funcionales		●														
	Base de Datos (2 Semanas)																
TO03	Diseñar el Modelo E/R				●												
TO04	Crear las Tablas y sus Relaciones				●												
TO05	Desarrollar los Triggers Necesarios					●											
TO06	Desarrollar los Procedimientos Almacenados Necesarios						●										
T007	Desarrollar las Vistas necesarias							●									
T008	Poblar la Base de Datos							●									
	Sistema Mascotas felices Global (1 Semanas)																
TO09	Diseñar la Arquitectura del Sistema Mascotas felices							●									
	Subsistema Mascotas felices-SRV (3 Semanas)																
TO10	Desarrollar el módulo de Búsquedas del Subsistema Mascotas felices							●									
TO11	Desarrollar el módulo de conexión a base de datos del Subsistema Mascotas felices-SRV							●		●							

Línea Base del Cronograma

ID	TAREA	LINEA BASE	
Administración (2 Semanas)		FECHA INI.	FECHA FIN.
TO01	Planificar el Proyecto (Alcance, Costo y Tiempo).	01/Sep	07/Sep
TO02	Recopilar los requerimientos funcionales y no funcionales.	08/Sep	14/Sep
Base de Datos (2 Semanas)			
TO03	Diseñar el Modelo E/R.	15/Sep	20/Sep
TO04	Crear las Tablas y sus Relaciones.	15/Sep	20/Sep
TO05	Desarrollar los Triggers Necesarios.	21/Sep	31/Sep
TO06	Desarrollar los Procedimientos Almacenados Necesarios.	21/Sep	31/Sep
TO07	Desarrollar las Vistas necesarias.	21/Sep	31/Sep
TO08	Poblar la Base de Datos.	21/Sep	31/Sep
Sistema Mascotas felices Global (1 Semanas)			
TO09	Diseñar la Arquitectura del Sistema Mascotas felices.	1/Oct	7/Oct
Subsistema Mascotas felices -SRV (3 Semanas)			
TO10	Desarrollar el módulo de Búsquedas del Subsistema Mascotas felices-SRV.	8/Oct	14/Oct
TO11	Desarrollar el módulo de conexión a base de datos del Subsistema Mascotas felices -SRV.	15/Oct	21/Oct
TO12	Integrar los módulos del Subsistema Mascotas felices -SRV.	22/Oct	30/Oct
TO13	Probar el Subsistema Mascotas felices - SRV.	22/Oct	30/Oct
Subsistema Mascotas felices -CTE (1 mes)			

TO14	Desarrollar el módulo de inicio de sesión y registro de usuarios del subsistema Mascotas felices-CTE.	01/Nov	7/Nov
TO15	Desarrollar el módulo de agenda de citas del Subsistema Mascotas felices-SRV.	1/Nov	7/Nov
TO16	Desarrollar el módulo de administración del Subsistema Mascotas felices-SRV.	8/Nov	14/Nov
TO17	Desarrollar el módulo de Perfil de usuario y gestión de mascotas del Subsistema Mascotas felices -SRV.	15/Nov	21/Nov
TO18	Implementar configuración de la cuenta (Cambio de contraseña, nombre, apellidos).	15/Nov	21/Nov
TO19	Desarrollar el módulo de conexión a base de datos del Subsistema Mascotas felices -CTE.	22/Nov	31/Nov
TO20	Integrar los módulos del Subsistema Mascotas felices -CTE.	22/Nov	31/Nov
TO21	Probar el Subsistema Mascotas felices - CTE.	22/Nov	31/Nov
Integración (2 Semanas)			
T022	Integrar los Subsistemas en Mascotas felices.	1/Dic	7/Dic
T023	Realizar pruebas de integración.	1/Dic	7/Dic

Gestión de Costos

Línea Base de Costos

Adquisiciones	Presupuesto
Laptop o PC de escritorio (2).	\$ 50,000 (único)
Herramientas de software.	\$ 0.0
Api Gemini Flash (para generación de recomendaciones según el análisis clínico de la mascota).	\$ 110 (mes)
Render - Hosting de api.	\$ 350 (mes)
Render - Hosting del frontend.	
Recursos humanos	
Director de proyecto	\$ 30,000 (mes)
Ingenieros de software (2)	\$ 25,000 c/u (mes)
Tester / QA	\$ 20,000 (mes)
Gastos operativos	
Oficina	\$ 5,000 (mes)
Luz	\$ 2,000 (mes)
Agua	\$ 1,000 (mes)
Limpieza	\$ 2,000 (mes)
Internet	\$ 600 (mes)

Gestión de Riesgos

Lista de Riesgos Identificados

CÓDIGO	DESCRIPCIÓN
R001	Dificultades en la implementación de backend con frontend
R002	Falta de experiencia en el uso de frameworks (Angular)
R003	Daño y/o perdida del equipo de trabajo (materiales)
R004	Falta de satisfacción del cliente sobre los avances del proyecto
R005	Cambios en las fechas de entrega de los avances
R006	Descarga incompleta de archivos de un repositorio
R007	Incompatibilidad con servicios de terceros por restricciones de licencias
R008	Fallo en los componentes de software seleccionados
R009	Pérdida de respaldos del proyecto
R010	Mala experiencia de usuario por un diseño poco intuitivo

R011	Mala gestión de permisos de usuarios (roles y accesos)
------	--

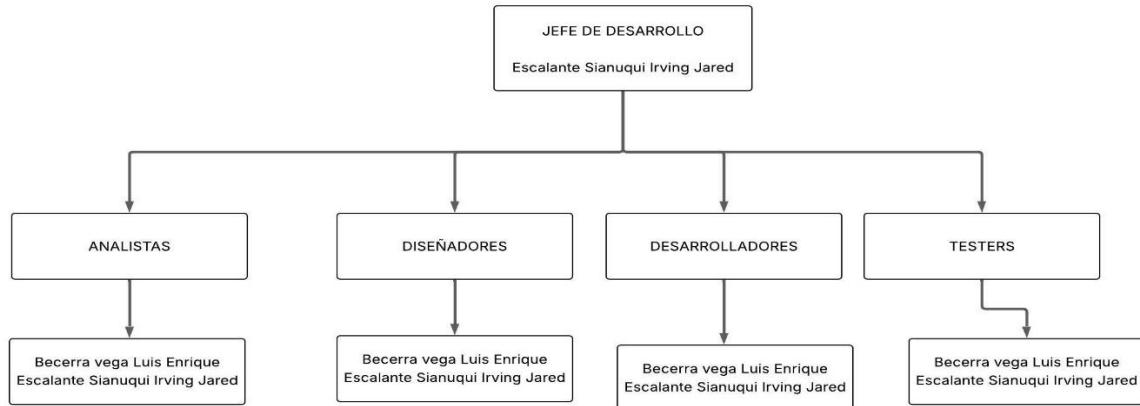
Lista de respuestas potenciales

RIESGO	ACCION PREVENTIVA (Antes del que el riesgo se presente)	ACCION CORRECTIVA (Una vez que el riesgo se presenta)
R001	Revisar la documentación oficial, además de realizar pruebas en entornos de prueba.	Se consultará documentación, foros y videos donde hagan uso de este, realizando pruebas en su implementación.
R002	Hacer uso de cursos sobre el framework que se utilizara	Contratar ayuda externa que ayude en la capacitación del personal, además de hagan uso de cursos en línea a lo largo del desarrollo
R003	Asegurar el manejo de los diferentes materiales/equipos con base a las políticas aplicables.	Reponer de forma inmediata el equipo dañado.
R004	Cumplir con los requerimientos que se establecieron con el cliente	Recolectar retroalimentación que nos ayude a mejorar el software que se está realizando
R005	Proponer ciertos objetivos a realizar en el transcurso de cada semana, para poder tener un movimiento activo en el desarrollo del software	Reforzar la comunicación y hacer uso de herramientas de gestión de proyectos
R006	Usar herramientas de control de versiones y revisar documentación.	Comprobar logs de descarga, clonar correctamente el repositorio o solicitar los archivos faltantes.
R007	Verificar los requisitos que necesitan de servicios de terceros y comprobar la compatibilidad de estos con nuestros respectivos dispositivos	Implementar servicios equivalentes o conseguir otros dispositivos para el entorno de pruebas
R008	Desarrollar prototipos y hacer pruebas de concepto del software/componentes a utilizar.	Buscar software alterno que cumpla con las funciones requeridas.
R009	Contar con respaldos en la nube o hacer uso de sistemas gestores de versiones	Restaurar los respaldos más recientes.

R010	Realizar pruebas de usabilidad antes de presentarlo con el cliente.	Ajustar el diseño en base a la retroalimentación de los usuarios.
R011	Definir roles y permisos desde un principio y realizar pruebas de acceso	Corregir permisos incorrectos y reforzar la seguridad de acceso

Gestión de Recursos Humanos

Organigrama del proyecto



Matriz de roles y responsabilidades

Integrantes	Roles	Responsabilidades
Escalante Sianuqui Irving Jared	Jefe de Desarrollo	<ul style="list-style-type: none">• Aportar el material requerido para llevar a cabo el desarrollo del proyecto• Toma de decisiones en general• Administración del proyecto• Supervisar el equipo de desarrollo
-Becerra vega Luis enrique -Escalante Sianuqui Irving Jared	Analistas	<ul style="list-style-type: none">• Obtener los requisitos• Asesoría y capacitación a diseñadores y desarrolladores• Toma de decisiones• Definir los tiempos y presupuesto
-Becerra vega Luis enrique -Escalante Sianuqui Irving Jared	Diseñadores	<ul style="list-style-type: none">• Respaldar información del proyecto• Diseñar prototipos que priorizando la usabilidad y experiencia de usuario
-Becerra vega Luis enrique -Escalante Sianuqui Irving Jared	Desarrolladores	<ul style="list-style-type: none">• Desarrollar la aplicación y el sistema• Pruebas y corrección de errores• Optimización de la aplicación y el sistema• Implementar las funcionalidades en base a los requisitos
-Becerra vega Luis enrique -Escalante Sianuqui Irving Jared	Testers	<ul style="list-style-type: none">• Encargado de la gestión de la calidad del proyecto• Notificar acerca de errores• Realizan pruebas• Validan que se cumplan con los requisitos

Gestión de Comunicaciones

Matriz de las comunicaciones

Información requerida	Responsables de distribuir la información	A quién se le entregará la información	Método de comunicación a ser utilizado	Frecuencia de la comunicación
Avance del proyecto	Desarrolladores	Jefe de desarrollo y al profesor	Documentos que contienen los avances del proyecto, se enviarán mediante correo electrónico y se subirán al repositorio de Github	Mensual
Preparar presupuesto de costos	Analistas	Jefe de desarrollo y Profesor	Informe con los gastos presupuestados, Los cuales se enviarán por correo electrónico	Inicio del proyecto
Entrega de la documentación	Analistas y desarrolladores	Jefe de desarrollo y Profesor	Los avances en la documentación serán subidos a classroom y al repositorio de github	Semanal
Plantear estándares de calidad	analistas y diseñadores	Jefe de desarrollo, Desarrolladores y profesor	Documento donde se establecen los estándares que se subirá a classroom y al repositorio de github	Inicio del proyecto
Preparar la evaluación	Desarrolladores	Jefe de desarrollo y Profesor	Informe de evaluación documentado y	Quincenal

			subido a classroom	
--	--	--	-----------------------	--

Herramientas de comunicación válidas para el proyecto

Nombre de la herramienta	Tipo de herramienta
whatsapp	Grupos y mensajería instantánea
Gmail	Correo electrónico
Discord	Grupos, mensajería instantánea y llamadas de voz
Github	Gestión de cambios con comentarios colaborativos

Gestión de Adquisiciones

Hardware

Nombre	Descripción	Precio	Proveedor	Adquisicion
LAPTOP.	LAPTOP Huawei Matebook D15 Windows 11, RAM 6GB, Procesador Rizen5.	\$18,000 MXN	HUAWEI	Compra.
LAPTOP	LAPTOP Lenovo	\$15,000	LENOVO	Compra

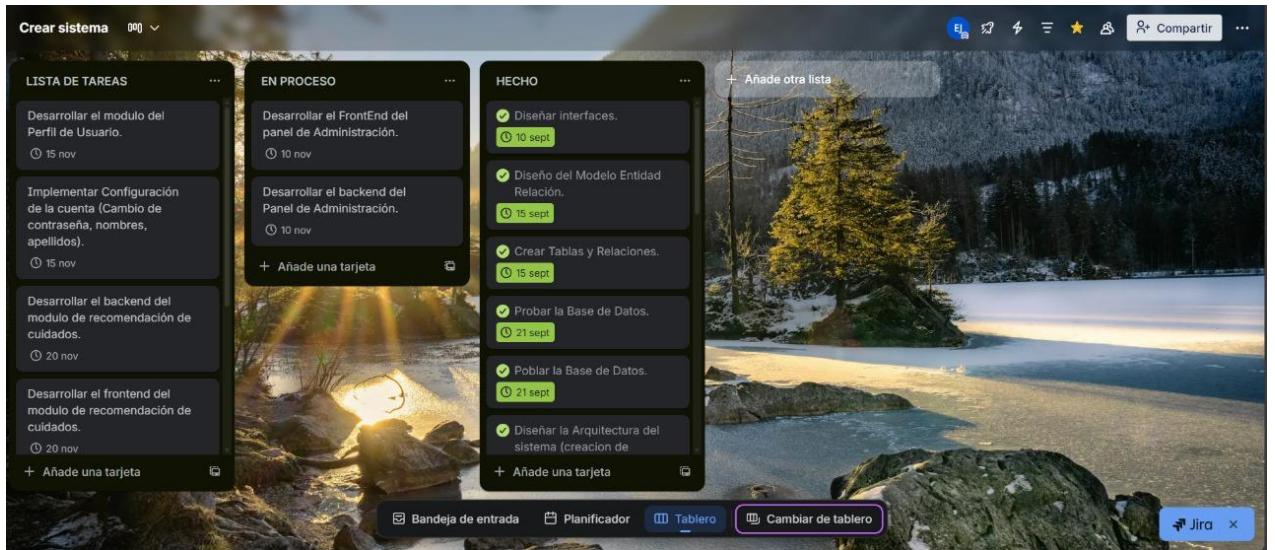
Software

Nombre	Descripción	Precio	Proveedor	Adquisición
Visual Studio	Visual Studio es un entorno de desarrollo para distintos lenguajes de programación. Este entorno fue desarrollado por Microsoft, aquí se va a desarrollar el frontend y la api.	\$0	Visual Studio: IDE y editor de código para el desarrollo de software	Gratis
PYCHARM.	PyCharm es un entorno de desarrollo integrado (IDE) para el lenguaje de programación Python, desarrollado por JetBrains . Aquí se desarrolló la api en el lenguaje de programación Python, que nos va a permitir comunicarnos con la base de datos	\$0	https://www.jetbrains.com/pycharm/	Gratis.
	para consultar, actualizar, insertar y			

	eliminar datos en el sistema.			
POSTMAN.	Postman es una plataforma integral para trabajar con APIs, que facilita la creación, prueba, documentación y colaboración en el desarrollo de APIs, desde el diseño hasta la entrega y monitoreo. Se usó para facilitar la creación de la api, realizar pruebas y monitoreo.	\$0	https://www.postman.com	Gratis.
MYSQL WORKBENCH	MySQL Workbench es una herramienta visual gratuita de gestión de bases de datos MySQL. Se uso para la gestión de la base de datos del sistema.	\$0	https://www.mysql.com/products/workbench	Gratis.
Render	Plataforma para que mi app este disponible desde cualquier	\$19	Cloud Application Platform Render	Pago Mensual

	navegador web.			
--	----------------	--	--	--

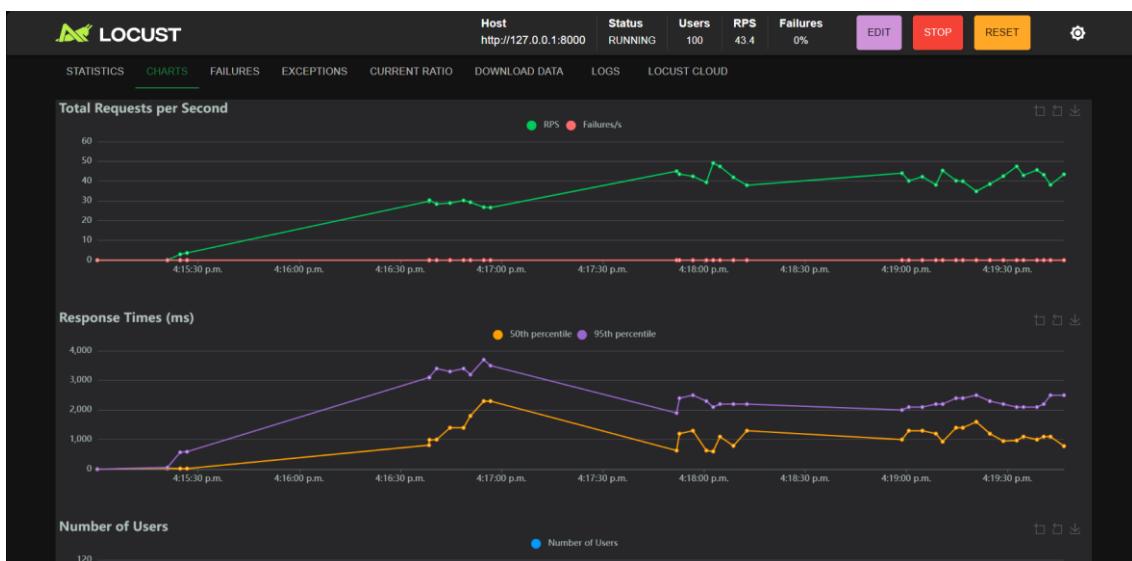
Seguimiento del proyecto en Trello



Anexos

Pruebas no funcionales

Pruebas de rendimiento en el que 100 usuarios realizaban 5 acciones por minuto (iniciar sesión, consultar citas, marcar asistencia e inasistencia, ver las mascotas relacionadas a un usuario). Se utilizo la librería de Locust de Python.



```
pruebas.py M
pruebas > pruebas.py

2
3     class VetUser(HttpUser):
4         wait_time = between(1, 3)
5
6         @task
7         def citas(self):
8             headers = {"Authorization": "Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZF91c3Vhcm1vIjoxMCwidXNlYX3pbIyI6Ikx1aXNbIisImNvcnJlbyI6ImV2ZkdhyMvjZk3yYTk3Mk8nbWFpb
9             self.client.get("/citas/pending/cita", headers=headers)
10            self.client.patch("/citas/mark/accept/4", headers=headers)
11            self.client.patch("/citas/mark/refused/4", headers=headers)
12
13         @task
14         def pets(self):
15             headers = {"Authorization": "Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZF91c3Vhcm1vIjoxLC1c3Vhcm1vIjoiHVYmVlbisImNvcnJlbyI6ImxizWNcnJhdWnVU8nbkfpbCSjb
16             self.client.get("/petpetsByUser/7", headers=headers)
17
18
19         @task
20         def login(self):
21             self.client.post("/auth/login", json={"usuario_email": "luv","contra": "luvbeen"})
22
23
24
25
26
27
28
29
29
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Aggregated 10584 0 (0.00%) | 994 9 4932 778 | 37.36 0.00

powerShell - pruebas + × ☰

Response time percentiles (approximated)	50%	66%	75%	80%	90%	95%	98%	99%	99.9%	99.99%	100%	# reqs
Type Name												
POST /auth/login	1900	2100	2300	2400	2900	3100	3600	3800	4400	4900	4900	2086
PATCH /citas/mark/accept/4	570	850	1100	1200	1600	1900	2300	2600	3000	3900	3900	2112
PATCH /citas/mark/refused/4	518	700	910	1100	1400	1800	2300	2500	3200	4000	4000	2096
GET /citas/pending/cita	656	1000	1200	1300	1700	2100	2500	2700	3700	4000	4000	2128
GET /petpetsByUser/7	680	1000	1200	1300	1700	2000	2500	2700	3600	4100	4100	2162
Aggregated	778	1200	1500	1700	2100	2500	2900	3200	4100	4600	4900	10584

Pruebas Funcionales

Se realiza inicio de sesión, creación de cuenta, creación de citas, consulta de citas pendientes, consulta de mascota por usuario, marcar asistencia e inasistencia

The screenshot shows a terminal window with the following output:

```
(.venv) PS C:\Users\lbece\OneDrive\Documents\GitHub\Backend-Veterinaria> pytest pruebas/prueba_funcional.py
```

pruebas@prueba_funcional.py

warnings summary

```
app\core\core.py:4
  C:\Users\lbece\OneDrive\Documents\GitHub\Backend-Veterinaria\app\core\core.py:4: PydanticDeprecatedSince20: Support for class-based `config` is deprecated, use ConfigDict instead. Deprecated in Pydantic instead. Deprecated in Pydantic v2.0 to be removed in v3.0. See Pydantic V2 Migration Guide at https://errors.pydantic.dev/2.12/migration/
    class Settings(BaseSettings):
```

```
app\utils\database.py:17
  C:\Users\lbece\OneDrive\Documents\GitHub\Backend-Veterinaria\app\utils\database.py:17: MovedIn20Warning: The ``declarative_base()`` function is now available as sqlalchemy.orm.declarative_base(). (deprecate since 2.0) (Background on SQLAlchemy 2.0 at: https://sqlalche.me/e/08d9)
    Base = declarative_base()
```

```
-- Docs: https://docs.pytest.org/en/stable/how-to/capture-warnings.html
```

```
= (.)venv) PS C:\Users\lbece\OneDrive\Documents\GitHub\Backend-Veterinaria> [ ]
```

A screenshot of a terminal window titled "pruebas > prueba_funcional.py". The window shows code for creating a consultation and running a pytest command. The output includes a warning summary and a success message indicating 6 passed tests and 2 warnings.

```
pruebas > prueba_funcional.py
 92 def create_cita():
 93     headers = {}
 94     payload = {
 95         "servicio": "consulta",
 96         "id_mascota":3,
 97         "fecha": "2025-15-11",
 98         "hora": "15:00",
 99         "id_veterinario":1,
100         "estado":"p"
101     }
102
103     response = client.post("/auth/login", json=payload, headers=headers)
104
105
106     assert response.status_code == 200
107
108     data = response.json()
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
(.venv) PS C:\Users\lbece\OneDrive\Documents\Github\Backend-Veterinaria> pytest pruebas/prueba_funcional.py
pruebas/prueba_funcional.py ..... [100%]
=====
===== warnings summary =====
app\core\core.py:4
  C:\Users\lbece\OneDrive\Documents\Github\Backend-Veterinaria\app\core\core.py:4: PydanticDeprecatedSince20: Support for class-based `config` is deprecated, use ConfigDict instead. Deprecated in Pydantic 2.0 instead. Deprecated in Pydantic v2.0 to be removed in v3.0. See Pydantic V2 Migration Guide at https://errors.pydantic.dev/2.12/migration/
    class Settings(BaseSettings):
        ...
app\utils\database.py:17
  C:\Users\lbece\OneDrive\Documents\Github\Backend-Veterinaria\app\utils\database.py:17: MovedIn20Warning: The ``declarative_base()`` function is now available as sqlalchemy.orm.declarative_base(). (deprecated since: 2.0) (Background on SQLAlchemy 2.0 at: https://sqlalchemy.me/rb8g)
    Base = declarative_base()
        ...
-- Docs: https://docs.pytest.org/en/stable/how-to/capture-warnings.html
=====
===== 6 passed, 2 warnings in 5.42s =====
< (.venv) PS C:\Users\lbece\OneDrive\Documents\Github\Backend-Veterinaria> []
```