Q.A

Plan de pruebas

Proyecto: Pet Store

Versión: 1.1

Historial de versiones

|  |  |  |  |
| --- | --- | --- | --- |
| Versión | Autor | Descripción | Fecha |
| 1.0 | Daniela Gutierrez Montoya | Creación del documento | 19/07/2022 |
| 1.1 | Daniela Gutierrez Montoya | Agregación de ítem de infraestructura | 22/07/2022 |

Índice

[Introducción 2](#_Toc109411470)

[Recursos 2](#_Toc109411473)

[Alcance 3](#_Toc109411474)

[Fuera del alcance 4](#_Toc109411475)

[Supuestos 4](#_Toc109411476)

[Infraestructura 4](#_Toc109411477)

[Riesgos 4](#_Toc109411478)

# 

# Introducción

Este documento tiene como objetivo dar a conocer la estrategia de ejecución del plan de pruebas del sistema, así como sus métodos, técnicas y forma en que se diseñan, ejecutan, evalúan y reportan los resultados de las pruebas de sistema.

Luego de realizar las pruebas se verificará y validara si el sistema; subir imágenes de mascotas, agregar, actualizar información, buscar y eliminar una mascota, cumplen con los requisitos exigidos por el usuario, en los cuales se deberán obtener los resultados esperados, de no ser así, se determina que se han encontrado errores que habrá que reportar a los desarolladores

# Recursos

|  |  |  |
| --- | --- | --- |
| Tester | % Participación | Seniority |
| Daniela Gutierrez Montoya | 100 | Junior |

# 

# Alcance

Se validará el correcto funcionamiento de agregar mascota a la tienda, subir imagen de una mascota, actualizar información de una macota, filtrar una mascota por status y por id y eliminar una mascota, mediante el perfil del administrador y el correcto funcionamiento de actualizar una mascota en la tienda con datos de formulario por medio del perfil de usuario (dueño de mascota) Según los requerimientos establecidos.

Adicional a esto se validará, la estructura del response, el status code obtenido luego de realizar una petición, que los métodos estén bien realizados y respondan de la manera correcta de acuerdo a los criterios de aceptación de las historias de usuario trabajadas y a su vez las siguientes reglas de negocios.

* Agregar una mascota en la tienda desde el perfil del administrador, validar el endpoint (/pet) de tipo POST cuando se envían datos correctos y se agregue una nueva mascota, validar el status code (201) he incorrectos (405).
* Subir una imagen de mascota en la tienda desde el perfil del administrador, validar el endpoint (/pet/{petId}/uploadImage) de tipo POST, cuando se envían datos correctos se validar el status code (200), status (400) pet no found cuando el campo “petid” no sea valido y status (405) invalid input cuando algún valor es invalido.
* Actualizar información de una mascota desde el perfil del administrador, validar el endpoint (/pet) de tipo PUT, cuando se envían datos correctos se validar el status code (200), status (400) pet no found cuando el campo “petid” no sea válido y status (405) invalid input cuando algún valor es invalido.
* Actualizar una mascota en la tienda con datos de formulario por medio del perfil del usuario (dueño de la mascota), validar el endpoint (/pet/{petId) de tipo POST, cuando se envían datos correctos se validar el status code (200), status (400) pet no found cuando el campo “petid” no sea válido y status (405) invalid input cuando algún valor es invalido.
* filtrar una mascota con (Estado) por medio del perfil del administrador, validar el endpoint (/pet/findByStatus) de tipo GET, cuando se envían datos correctos se validar el status code (200), status (400) pet no found cuando el estado es incorrecto.
* filtrar una mascota con (Id) por medio del perfil del administrador, validar el endpoint (/pet/{petId}) de tipo GET, cuando se envían datos correctos se validar el status code (200), status (400) pet no found cuando el campo “petid” no sea válido.
* Eliminar una mascota por medio del perfil del administrador, validar el endpoint (/pet/{petId}) de tipo DELETE, cuando se envían datos correctos se validar el status code (200), status (400) pet no found cuando el campo “petid” no sea válido.

Se verificará la respuesta del status code generado.

# Fuera del alcance

* Servidores por los que va a pasar el servicio
* Estructura del response.
* Pruebas de performance.

# Supuestos

* Que las pruebas unitarias estén correctamente probadas
* Las pruebas unitarias cubren el 98% del código

# Infraestructura

* Acceso al ambiente de pruebas.
* Acceso a Postman
* Acceso al repositorio (para agregar test automatizados)

# Riesgos

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ID | Riesgo | Probabilidad (1-5) | Impacto (1-5) | Severidad (Probabilidad por impacto) | Plan de mitigación |
| R1 | Rotaciones en el equipo | 2 | 4 | 8 | Negociar las horas del equipo hasta su finalización |
| R2 | Que el sistema no pueda soportar muchos registros de mascotas | 2 | 5 | 10 | Antes de determinar esto debemos enviar trafico falso de registros para saber cómo responde. Hacer pruebas de performance |