

UNIVERSIDAD DE LAS FUERZAS ARMADAS ESPE



Departamento de Ciencias de la Computación

Análisis y Diseño de Software

Especificación de Casos de Prueba - Proyecto Hoptolt

Versión 1

Elaborado por:

Kevin Asmal

Diego Delgado

Gabriel López

Marcelo Pareja

Ecuador 2026-01-24

Versión	Fecha de Revisión
1	25/01/2025

Tabla de Contenidos

1. Información General del Proyecto.....	5
2. Funcionalidades.....	5
2.1. Inicio de sesión:.....	5
2.2. Gestión de Jaulas:.....	5
2.3. Gestión de Razas:.....	5
3. Casos de Prueba.....	5
3.1. CP01.1-1 Inicio de Sesión.....	5
3.2. CP01.1-2 Inicio de Sesión con credenciales incorrectas.....	6
3.3. CP02.1-1 Creación de jaula.....	7
3.4. CP02.1-2 Creación de jaula con datos correctos.....	8
3.5. CP02.2-1 Buscar información de una jaula.....	9
3.6. CP02.2-2 - Buscar información de jaula inexistente.....	10
3.7. CP02.3-1 - Modificar de datos de un conejo.....	11
3.8. CP02.3-2 - Modificar de datos de un conejo con información incorrecta.....	12
3.9. CP02.4-1 - Eliminar una jaula.....	13
3.10. CP02.4-2 - Eliminar una jaula con conejos.....	15
3.11. CP03.1-1 - Registro de nueva raza.....	16
3.13. CP03.2-1 - Buscar raza por nombre.....	18
3.14. CP03.2-2 - Buscar raza por nombre con datos incorrectos.....	19
3.15. CP03.3-1 - Modificar información de raza.....	19
3.16. CP03.3-2 - Modificar información de raza con datos inválidos.....	20
3.17. CP0.....	20
3.18. CP0.....	21
3.19. CP0.....	21
3.20. CP0.....	21
3.21. CP0.....	22
3.22. CP0.....	22
3.23. CP0.....	23
3.24. CP0.....	23
3.25. CP0.....	24
3.26. CP0.....	24
3.27. CP0.....	25
3.28. CP0.....	25
3.29. CP0.....	25
3.30. CP0.....	26
3.31. CP0.....	26
3.32. CP0.....	27
3.33. CP0.....	27
3.34. CP0.....	28
3.35. CP0.....	28

3.36. CP08.1-1 Registrar datos de la monta.....	29
3.37. CP08.1-2 Obtener datos de la monta.....	29
3.38. CP08.1-3 Obtener datos de la monta por conejo.....	30
3.39. CP08.1-4 Registrar datos de la monta.....	31
4. Pruebas de Flujo.....	32

1. Información General del Proyecto

Hoptolt es un proyecto orientado a la gestión del proceso de crianza de los conejos. El sistema está pensado para permitir a los cuidadores realizar actividades de control sobre sus conejos, agrupándolos en jaulas para monta o engorde, y llevando un seguimiento del crecimiento, dieta y tratamientos médicos de cada conejo. El sistema además permite generar reportes de todos estos aspectos, para facilitar el control y poder tener registros de cada cierto tiempo.

2. Funcionalidades

2.1. Inicio de sesión:

El sistema implementa un login para proteger el acceso a la información del sistema. El usuario debe ingresar usuario y contraseña, para poder ingresar y usar todas las funcionalidades del sistema.

2.2. Gestión de Jaulas:

El actor accede a la sección para gestionar jaulas. El sistema le permitirá registrar, editar, eliminar y consultar datos sobre jaulas. El actor selecciona la acción que desea realizar.

2.3. Gestión de Razas:

El actor accede a la sección para gestionar la raza. El sistema le permitirá: registrar, editar, eliminar datos sobre jaulas.

3. Casos de Prueba

3.1. CP01.1-1 Inicio de Sesión

IDENTIFICADOR	CP01.1-1
DESCRIPCIÓN	Se ingresan credenciales (usuario y contraseña) correctos para iniciar sesión.

PRECONDICIONES	El usuario debe existir
DATOS DE ENTRADA	<pre>{ "username": "usuario_pruebas", "password": "p1234" }</pre>
RESULTADO ESPERADO	Se inicia la sesión correctamente. El sistema devuelve un token para el uso de las demás funcionalidades.
RESULTADO OBTENIDO	El usuario inicia sesión
ESTADO	Aprobado

Evidencia

POST `_url /usuario/login` Send

Params Body Auth Headers 4 Scripts 0 Docs

JSON

```

1 * {
2     "username": "usuario_pruebas",
3     "password": "p1234"
4 }
```

Beautify JSON

200 OK 23.3 s 259 B 5 Minutes Ago

Preview Headers 12 Cookies Tests 1 / 1 → Mock Console

Preview

```

1 * {
2     "token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpZCI6IjEwMDEwMTAwMTAxIiwidXNlYXJpbI6InVzdWFyaW9fcHJ1ZWJhcylsImlhdcI6MTc2OTMxNTM5OCwiZXhwIjoxNzY5MzE4OTk4fQ.-GqnOFOPDVQN6ktzOWrpx047_PsLcZ3HlorZo94v2w",
3     "user": {
4         "id": "10010100101",
5         "usuario": "usuario_pruebas"
6     }
7 }
```

3.2. CP01.1-2 Inicio de Sesión con credenciales incorrectas

IDENTIFICADOR	CP01.1-2
DESCRIPCIÓN	Se ingresan credenciales (usuario y contraseña) incorrectos para iniciar sesión
PRECONDICIONES	El usuario debe existir

DATOS DE ENTRADA	<pre>{ "username": "usuario_falso", "password": "contra1234" }</pre>
RESULTADO ESPERADO	El sistema devuelve un mensaje indicando credenciales inválidas y no inicia sesión.
RESULTADO OBTENIDO	El sistema indica que el mensaje no se ha encontrado y no devuelve token de inicio de sesión
ESTADO	Aprobado

Evidencia

The screenshot shows a POST request to the endpoint `/usuario/login`. The request body is set to JSON and contains the following data:

```

1 ▶ {
2     "username": "usuario_falso",
3     "password": "contra1234"
4 }
```

The response status is **401 Unauthorized**, with a response time of 677 ms and a response size of 35 B. The response body is:

```

1 ▶ {
2     "message": "Usuario no encontrado"
3 }
```

3.3. CP02.1-1 Creación de jaula

IDENTIFICADOR	CP02.1-1
DESCRIPCIÓN	Se ingresan los datos correctos de una jaula para monta o engorde

PRECONDICIONES	El usuario ha iniciado sesión y tiene un token de autorización
DATOS DE ENTRADA	<pre>{ "jaula_tipo": "Engorde", "jaula_capacidad": 6 }</pre>
RESULTADO ESPERADO	El sistema crea la jaula exitosamente y muestra un mensaje de confirmación
RESULTADO OBTENIDO	El sistema muestra un mensaje de éxito de la jaula creada
ESTADO	Aprobado

Evidencia

3.4. CP02.1-2 Creación de jaula con datos correctos

IDENTIFICADOR	CP02.1-2
DESCRIPCIÓN	Se ingresan datos de capacidad incorrectos para el tipo de jaula
PRECONDICIONES	El usuario ha iniciado sesión y tiene un token de autorización
DATOS DE ENTRADA	<pre>{ "jaula_tipo": "Reproduccion", "jaula_capacidad": 6 }</pre>
RESULTADO ESPERADO	El sistema devuelve un mensaje indicando que la capacidad no es válida para ese tipo de jaula
RESULTADO OBTENIDO	El sistema indica que la capacidad no es válida para el tipo de jaula ingresada.
ESTADO	Aprobado

Evidencia

POST ▼ https://two7835-g5-ads.onrender.com/jaula/new

Params Body (green circle) Auth Headers (4) Scripts (green circle) Docs

JSON ▼

```

1 ▾ {
2   "jaula_tipo": "Reproduccion",
3   "jaula_capacidad": 6
4 }
```

Beautify JSON

400 Bad Request 688 ms 65 B

Preview Headers (12) Cookies Tests 0 / 1 → Mock Console

Preview ▼

```

1 ▾ {
2   "message": "Capacidad inválida para jaula de tipo Reproduccion"
3 }
```

3.5. CP02.2-1 Buscar información de una jaula

IDENTIFICADOR	CP02.2-1
DESCRIPCIÓN	Se ingresa el ID de una jaula existente y se realiza la búsqueda
PRECONDICIONES	El usuario ha iniciado sesión y tiene un token de autorización
DATOS DE ENTRADA	Parámetros: id de una Jaula existente. “J0001” https://two7835-g5-ads.onrender.com/jaula/J0001
RESULTADO ESPERADO	El sistema devuelve los datos correctamente
RESULTADO OBTENIDO	El sistema devuelve los datos correctamente
ESTADO	Aprobado

Evidencia

GET ▼ <https://two7835-g5-ads.onrender.com/jaula/J0001>

Params Body Auth Headers (3) Scripts (green circle) Docs

URL PREVIEW
<https://two7835-g5-ads.onrender.com/jaula/J0001>

QUERY PARAMETERS

Add Delete all Description

PATH PARAMETERS

200 OK 1.33 s 63 B

Preview Headers (12) Cookies Tests 1 / 1 → Mock Console

Preview ▾

```

1 ▾ {
2   "jaula_id": "J0001",
3   "jaula_tipo": "Engorde",
4   "jaula_capacidad": 6
5 }
```

3.6. CP02.2-2 - Buscar información de jaula inexistente

IDENTIFICADOR	CP02.2-2
DESCRIPCIÓN	Se ingresa un ID inválido para la búsqueda
PRECONDICIONES	El usuario ha iniciado sesión y tiene un token de autorización
DATOS DE ENTRADA	ID de jaula que no existe en la base de datos: Ej: “J999” https://two7835-g5-ads.onrender.com/jaula/J999
RESULTADO ESPERADO	El sistema indica que no se encontró la jaula
RESULTADO OBTENIDO	El sistema responde con un mensaje de jaula no encontrada
ESTADO	Aprobado

Evidencia

The screenshot shows a REST API testing interface. At the top, it displays a GET request to <https://two7835-g5-ads.onrender.com/jaula/J999>. Below the URL, there are tabs for Params, Body, Auth, Headers, Scripts, and Docs. Under Headers, there are three entries. The main area shows a URL PREVIEW of <https://two7835-g5-ads.onrender.com/jaula/J999>. Below it, there are sections for QUERY PARAMETERS and PATH PARAMETERS, both of which are currently empty. A large orange box indicates a 404 Not Found response with a duration of 878 ms and a body size of 33 B. Below this, there are tabs for Preview, Headers (with 12 items), Cookies, Tests (0/1), Mock, and Co. The Preview section shows a JSON response with one message: "Jaula no encontrada".

```

1 ▾ {
2   "message": "Jaula no encontrada"
3 }

```

3.7. CP02.3-1 - Modificar de datos de un conejo

IDENTIFICADOR	CP02.3-1
DESCRIPCIÓN	Se modifica una jaula ya existente cambiando su tipo y capacidad de manera correcta
PRECONDICIONES	El usuario ha iniciado sesión y tiene un token de autorización
DATOS DE ENTRADA	Parámetros: ID de la jaula a modificar https://two7835-g5-ads.onrender.com/jaula/update/J0001 Cuerpo: Nuevos datos <pre>{ "jaula_tipo": "Reproduccion", "jaula_capacidad": 1 }</pre>
RESULTADO ESPERADO	El sistema indica que se han cambiado los datos con éxito.
RESULTADO OBTENIDO	Mensaje de confirmación del sistema

ESTADO	Aprobado
--------	----------

Evidencia

PUT ▼ <https://two7835-g5-ads.onrender.com/jaula/update/J0001>

Params Body (1) Auth (1) Headers (4) Scripts (1) Docs

JSON ▼

```

1 v {
2   "jaula_tipo": "Reproduccion",
3   "jaula_capacidad": 1
4 }
```

Beautify JSON

200 OK 1.05 s 44 B

Preview Headers (12) Cookies Tests 1/1 → Mock Console

Preview ▼

```

1 v {
2   "message": "Jaula actualizada exitosamente"
3 }
```

3.8. CP02.3-2 - Modificar de datos de un conejo con información incorrecta

IDENTIFICADOR	CP02.3-2
DESCRIPCIÓN	Se modifica una jaula ya existente con información de tipo y capacidad incorrecta. Luego se busca al conejo por ID.
PRECONDICIONES	El usuario ha iniciado sesión y tiene un token de autorización
DATOS DE ENTRADA	<p>Parámetros: ID de la jaula a modificar</p> <p>https://two7835-g5-ads.onrender.com/jaula/update/J0001</p> <p>Cuerpo: Nuevos datos</p> <pre>{ "jaula_tipo": "Engorde", "jaula_capacidad": 9 }</pre>
RESULTADO ESPERADO	El sistema indica que la capacidad no es adecuada para el tipo de jaula y no modifica la jaula.

RESULTADO OBTENIDO	El sistema devuelve un mensaje de error, y no se realizaron cambios en la jaula
ESTADO	Aprobado

Evidencia

The screenshot shows two API requests in a testing interface:

PUT https://two7835-g5-ads.onrender.com/jaula/update/J0001

Request Body (JSON):

```

1 ▶ {
2   "jaula_tipo": "Engorde",
3   "jaula_capacidad": 9
4 }

```

Response: 400 Bad Request | 1.21 s | 58 B

Preview | Headers 12 | Cookies | Tests 0 / 1 | → Mock | Console

Preview

```

1 ▶ {
2   "message": "Capacidad no permitida para jaula de Engorde"
3 }

```

GET https://two7835-g5-ads.onrender.com/jaula/J0001

Request Body:

No Body

Response: 200 OK | 582 ms | 68 B

Preview | Headers 12 | Cookies | Tests 1 / 1 | → Mock

Preview

```

1 ▶ {
2   "jaula_id": "J0001",
3   "jaula_tipo": "Reproduccion",
4   "jaula_capacidad": 1
5 }

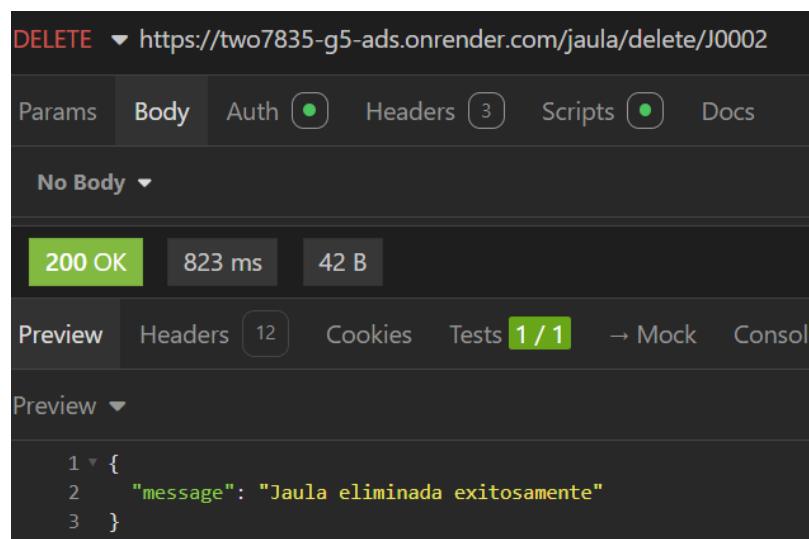
```

3.9. CP02.4-1 - Eliminar una jaula

IDENTIFICADOR	CP02.4-1
----------------------	----------

DESCRIPCIÓN	Se selecciona una jaula sin conejos y se elimina. Luego se busca el ID de esa jaula
PRECONDICIONES	El usuario ha iniciado sesión y tiene un token de autorización
DATOS DE ENTRADA	Parámetros: ID de la jaula a eliminar <code>DELETE ▾ https://two7835-g5-ads.onrender.com/jaula/delete/J0002</code>
RESULTADO ESPERADO	El sistema confirma el borrado de la jaula y no se encuentra en la búsqueda
RESULTADO OBTENIDO	El sistema envía un mensaje indicado que la jaula se eliminó exitosamente. La búsqueda no devuelve resultados.
ESTADO	Aprobado

Evidencia



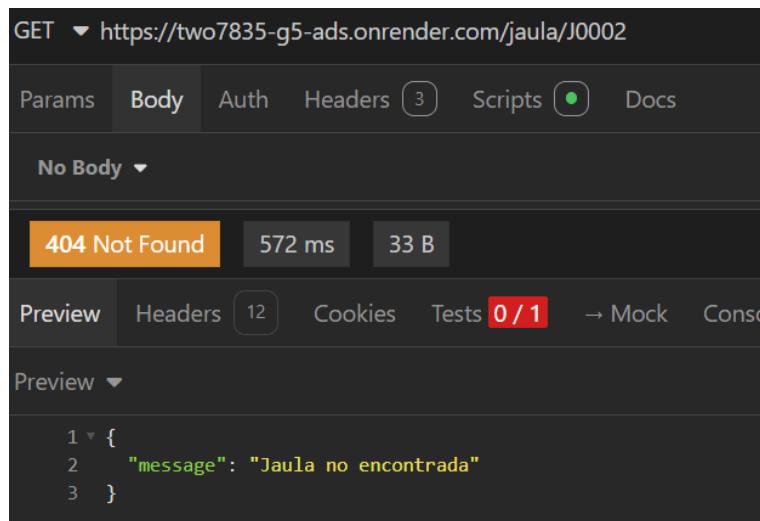
The screenshot shows a POSTMAN interface with the following details:

- Method:** DELETE
- URL:** https://two7835-g5-ads.onrender.com/jaula/delete/J0002
- Body:** No Body
- Response Status:** 200 OK
- Time:** 823 ms
- Size:** 42 B
- Preview:** Shows a JSON response with one message: "Jaula eliminada exitosamente".
- Headers:** 12
- Cookies:** None
- Tests:** 1 / 1
- Mock:** None
- Console:** None

```

1 ▾ {
2   "message": "Jaula eliminada exitosamente"
3 }

```



3.10. CP02.4-2 - Eliminar una jaula con conejos

IDENTIFICADOR	CP02.4-2
DESCRIPCIÓN	Se escoge una jaula que tenga conejos y se elimina. Luego se busca la jaula por ID
PRECONDICIONES	El usuario ha iniciado sesión y tiene un token de autorización
DATOS DE ENTRADA	Parámetros: ID de la jaula a eliminar https://two7835-g5-ads.onrender.com/jaula/update/J0001
RESULTADO ESPERADO	El sistema emite un mensaje de error indicando que la jaula tiene conejos y que deben quitarse antes de eliminarla. La jaula aparece al buscarla
RESULTADO OBTENIDO	El sistema emite un mensaje de error indicando que la jaula no se puede eliminar por una restricción de la base de datos. La búsqueda devuelve la jaula
ESTADO	Cumple parcialmente

Evidencia

The screenshot shows a REST API testing interface with two requests for the endpoint `https://two7835-g5-ads.onrender.com/jaula`.

Request 1: DELETE /jaula/J0001

- Method: `DELETE`
- URL: `https://two7835-g5-ads.onrender.com/jaula/J0001`
- Status: `400 Bad Request`
- Time: `1.12 s`
- Size: `118 B`
- Headers: `No Body`
- Tests: `0/1`
- Preview: `No Body`
- Response:

```
1 ▶ {
2   "message": "update or delete on table \"jaula\" violates foreign key constraint \"conejo_jaula\" on table \"conejo\""
3 }
```

Request 2: GET /jaula/J0001

- Method: `GET`
- URL: `https://two7835-g5-ads.onrender.com/jaula/J0001`
- Status: `200 OK`
- Time: `593 ms`
- Size: `68 B`
- Headers: `No Body`
- Tests: `1/1`
- Preview: `No Body`
- Response:

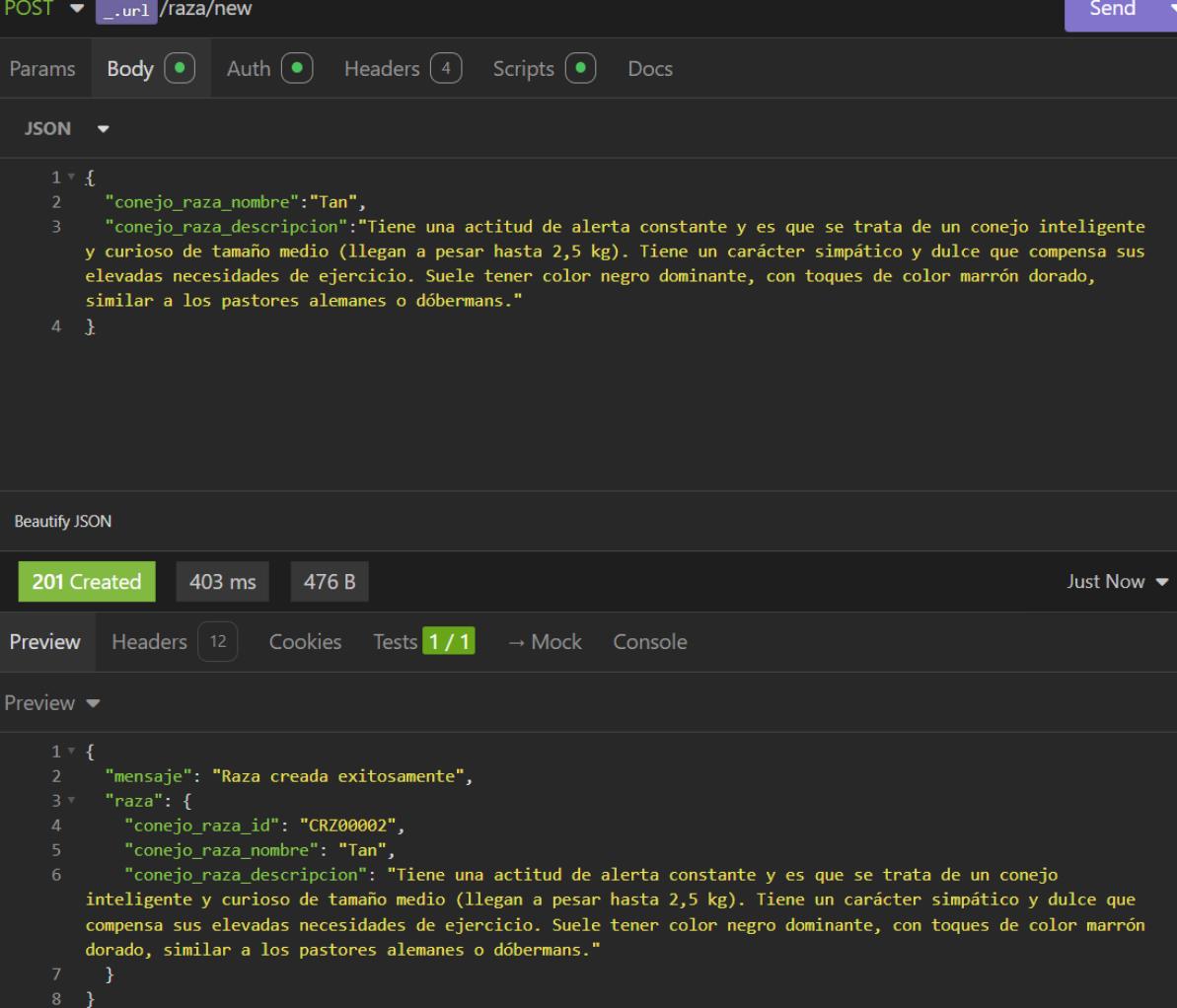
```
1 ▶ {
2   "jaula_id": "J0001",
3   "jaula_tipo": "Reproduccion",
4   "jaula_capacidad": 1
5 }
```

3.11. CP03.1-1 - Registro de nueva raza

IDENTIFICADOR	CP03.1-1
DESCRIPCIÓN	Se registra una raza con un nuevo nombre y descripción.
PRECONDICIONES	El usuario ha iniciado sesión y tiene un token de autorización
DATOS DE ENTRADA	<pre>{ "conejo_raza_nombre": "Tan", "conejo_raza_descripcion": "Tiene una actitud de alerta constante y es que se trata de un conejo inteligente y curioso de tamaño medio (llegan a pesar hasta 2,5 kg). Tiene un carácter simpático y dulce que compensa sus elevadas necesidades de ejercicio. Suele tener color negro dominante, con toques de color marrón dorado, similar a los pastores alemanes o dóbermans." }</pre>
RESULTADO ESPERADO	El sistema devuelve un mensaje indicando que la raza se creó con éxito

RESULTADO OBTENIDO	El sistema indica que la raza se creó exitosamente y muestra información de la nueva raza creada
ESTADO	Aprobado

Evidencia



The screenshot shows a Postman API request for creating a new rabbit breed. The URL is `POST /raza/new`. The request body is a JSON object:

```

1 ▶ {
2   "conejo_raza_nombre": "Tan",
3   "conejo_raza_descripcion": "Tiene una actitud de alerta constante y es que se trata de un conejo inteligente y curioso de tamaño medio (llegan a pesar hasta 2,5 kg). Tiene un carácter simpático y dulce que compensa sus elevadas necesidades de ejercicio. Suele tener color negro dominante, con toques de color marrón dorado, similar a los pastores alemanes o dóbermans."
4 }

```

The response status is **201 Created**, with a time of **403 ms** and a size of **476 B**. The response body is:

```

1 ▶ {
2   "mensaje": "Raza creada exitosamente",
3   "raza": {
4     "conejo_raza_id": "CRZ00002",
5     "conejo_raza_nombre": "Tan",
6     "conejo_raza_descripcion": "Tiene una actitud de alerta constante y es que se trata de un conejo inteligente y curioso de tamaño medio (llegan a pesar hasta 2,5 kg). Tiene un carácter simpático y dulce que compensa sus elevadas necesidades de ejercicio. Suele tener color negro dominante, con toques de color marrón dorado, similar a los pastores alemanes o dóbermans."
7   }
8 }

```

3.12. CP03.1-2 - Registrar nueva raza con información incompleta

IDENTIFICADOR	CP03.1-2
DESCRIPCIÓN	Se registra una raza solamente con nombre y sin descripción
PRECONDICIONES	El usuario ha iniciado sesión y tiene un token de autorización

DATOS DE ENTRADA	<pre>{ "conejo_raza_nombre": "Conejo Mariposa" }</pre>
RESULTADO ESPERADO	El sistema devuelve un mensaje de error pidiendo que se completen todos los campos
RESULTADO OBTENIDO	El sistema devuelve un mensaje de error que indica que los datos ingresados violan las reglas de integridad de la base de datos.
ESTADO	Cumple parcialmente

Evidencia

POST `_url` /raza/new

Params Body Auth Headers Scripts Docs

JSON

```

1 ▶ {
2   "conejo_raza_nombre": "Conejo Mariposa"
3 }
```

Beautify JSON

400 Bad Request 786 ms 119 B

Preview Headers 12 Cookies Tests 0/1 → Mock Console

Preview ▾

```

1 ▶ {
2   "mensaje": "null value in column \\\"conejo_raza_descripcion\\\" of relation \\\"conejo_raza\\\" violates not-null constraint"
3 }
```

3.13. CP03.2-1 - Buscar raza por nombre

IDENTIFICADOR	CP03.2-1
DESCRIPCIÓN	Se busca la información de la raza por el nombre
PRECONDICIONES	El usuario ha iniciado sesión y tiene un token de autorización
DATOS DE ENTRADA	
RESULTADO ESPERADO	El sistema devuelve la información de la raza buscada

RESULTADO OBTENIDO	
ESTADO	No ejecutado

Evidencia

3.14. CP03.2-2 - Buscar raza por nombre con datos incorrectos

IDENTIFICADOR	CP03.2-2
DESCRIPCIÓN	Se busca información de la raza con un nombre inválido
PRECONDICIONES	El usuario ha iniciado sesión y tiene un token de autorización
DATOS DE ENTRADA	
RESULTADO ESPERADO	El sistema envía un mensaje de error indicando que no se encontró la raza
RESULTADO OBTENIDO	
ESTADO	No ejecutado

Evidencia

3.15. CP03.3-1 - Modificar información de raza

IDENTIFICADOR	CP03.3-1
DESCRIPCIÓN	Se escoge una raza existente y se modifican su nombre y descripción. Luego se busca por ID
PRECONDICIONES	El usuario ha iniciado sesión y tiene un token de autorización
DATOS DE ENTRADA	
RESULTADO ESPERADO	El sistema indica que se han cambiado los datos con éxito. La búsqueda refleja el cambio
RESULTADO OBTENIDO	

ESTADO	No ejecutado
---------------	--------------

Evidencia

3.16. CP03.3-2 - Modificar información de raza con datos inválidos

IDENTIFICADOR	CP03.3-2
DESCRIPCIÓN	Se escoge una raza para modificar, pero se ingresan datos erróneos. Luego se busca la raza por ID.
PRECONDICIONES	El usuario ha iniciado sesión y tiene un token de autorización
DATOS DE ENTRADA	
RESULTADO ESPERADO	El sistema indica que los datos son incorrectos, y la búsqueda no muestra cambios en los datos
RESULTADO OBTENIDO	
ESTADO	No ejecutado

Evidencia

3.17. CP0

IDENTIFICADOR	
DESCRIPCIÓN	
PRECONDICIONES	El usuario ha iniciado sesión y tiene un token de autorización
DATOS DE ENTRADA	
RESULTADO ESPERADO	
RESULTADO OBTENIDO	
ESTADO	

Evidencia

3.18. CP0

IDENTIFICADOR	
DESCRIPCIÓN	
PRECONDICIONES	El usuario ha iniciado sesión y tiene un token de autorización
DATOS DE ENTRADA	
RESULTADO ESPERADO	
RESULTADO OBTENIDO	
ESTADO	

Evidencia

3.19. CP0

IDENTIFICADOR	
DESCRIPCIÓN	
PRECONDICIONES	El usuario ha iniciado sesión y tiene un token de autorización
DATOS DE ENTRADA	
RESULTADO ESPERADO	
RESULTADO OBTENIDO	
ESTADO	

Evidencia

3.20. CP0

IDENTIFICADOR	
DESCRIPCIÓN	

PRECONDICIONES	El usuario ha iniciado sesión y tiene un token de autorización
DATOS DE ENTRADA	
RESULTADO ESPERADO	
RESULTADO OBTENIDO	
ESTADO	

Evidencia

3.21. CP0

IDENTIFICADOR	
DESCRIPCIÓN	
PRECONDICIONES	El usuario ha iniciado sesión y tiene un token de autorización
DATOS DE ENTRADA	
RESULTADO ESPERADO	
RESULTADO OBTENIDO	
ESTADO	

Evidencia

3.22. CP0

IDENTIFICADOR	
DESCRIPCIÓN	
PRECONDICIONES	El usuario ha iniciado sesión y tiene un token de autorización
DATOS DE ENTRADA	

RESULTADO ESPERADO	
RESULTADO OBTENIDO	
ESTADO	

Evidencia

3.23. CP0

IDENTIFICADOR	
DESCRIPCIÓN	
PRECONDICIONES	El usuario ha iniciado sesión y tiene un token de autorización
DATOS DE ENTRADA	
RESULTADO ESPERADO	
RESULTADO OBTENIDO	
ESTADO	

Evidencia

3.24. CP0

IDENTIFICADOR	
DESCRIPCIÓN	
PRECONDICIONES	El usuario ha iniciado sesión y tiene un token de autorización
DATOS DE ENTRADA	
RESULTADO ESPERADO	
RESULTADO OBTENIDO	
ESTADO	

Evidencia

3.25. CP0

IDENTIFICADOR	
DESCRIPCIÓN	
PRECONDICIONES	El usuario ha iniciado sesión y tiene un token de autorización
DATOS DE ENTRADA	
RESULTADO ESPERADO	
RESULTADO OBTENIDO	
ESTADO	

Evidencia

3.26. CP0

IDENTIFICADOR	
DESCRIPCIÓN	
PRECONDICIONES	El usuario ha iniciado sesión y tiene un token de autorización
DATOS DE ENTRADA	
RESULTADO ESPERADO	
RESULTADO OBTENIDO	
ESTADO	

Evidencia

3.27. CP0

IDENTIFICADOR	
DESCRIPCIÓN	
PRECONDICIONES	El usuario ha iniciado sesión y tiene un token de autorización
DATOS DE ENTRADA	
RESULTADO ESPERADO	
RESULTADO OBTENIDO	
ESTADO	

Evidencia

3.28. CP0

IDENTIFICADOR	
DESCRIPCIÓN	
PRECONDICIONES	El usuario ha iniciado sesión y tiene un token de autorización
DATOS DE ENTRADA	
RESULTADO ESPERADO	
RESULTADO OBTENIDO	
ESTADO	

Evidencia

3.29. CP0

IDENTIFICADOR	
DESCRIPCIÓN	

PRECONDICIONES	El usuario ha iniciado sesión y tiene un token de autorización
DATOS DE ENTRADA	
RESULTADO ESPERADO	
RESULTADO OBTENIDO	
ESTADO	

Evidencia

3.30. CP0

IDENTIFICADOR	
DESCRIPCIÓN	
PRECONDICIONES	El usuario ha iniciado sesión y tiene un token de autorización
DATOS DE ENTRADA	
RESULTADO ESPERADO	
RESULTADO OBTENIDO	
ESTADO	

Evidencia

3.31. CP0

IDENTIFICADOR	
DESCRIPCIÓN	
PRECONDICIONES	El usuario ha iniciado sesión y tiene un token de autorización
DATOS DE ENTRADA	

RESULTADO ESPERADO	
RESULTADO OBTENIDO	
ESTADO	

Evidencia

3.32. CP0

IDENTIFICADOR	
DESCRIPCIÓN	
PRECONDICIONES	El usuario ha iniciado sesión y tiene un token de autorización
DATOS DE ENTRADA	
RESULTADO ESPERADO	
RESULTADO OBTENIDO	
ESTADO	

Evidencia

3.33. CP0

IDENTIFICADOR	
DESCRIPCIÓN	
PRECONDICIONES	El usuario ha iniciado sesión y tiene un token de autorización
DATOS DE ENTRADA	
RESULTADO ESPERADO	
RESULTADO OBTENIDO	
ESTADO	

Evidencia

3.34. CP0

IDENTIFICADOR	
DESCRIPCIÓN	
PRECONDICIONES	El usuario ha iniciado sesión y tiene un token de autorización
DATOS DE ENTRADA	
RESULTADO ESPERADO	
RESULTADO OBTENIDO	
ESTADO	

Evidencia

3.35. CP0

IDENTIFICADOR	
DESCRIPCIÓN	
PRECONDICIONES	El usuario ha iniciado sesión y tiene un token de autorización
DATOS DE ENTRADA	
RESULTADO ESPERADO	
RESULTADO OBTENIDO	
ESTADO	

Evidencia

3.36. CP08.1-1 Registrar datos de la monta

IDENTIFICADOR	CP08.1-1
DESCRIPCIÓN	Se ingresan los datos de la monta para su guardado en la base de datos
PRECONDICIONES	El usuario ha iniciado sesión y tiene un token de autorización
DATOS DE ENTRADA	<pre>{ "reproduccion_fecha": "2026-02-10", "conejo_id": "C0001", "reproduccion_estado": "Preñada" }</pre>
RESULTADO ESPERADO	El sistema crea una monta exitosamente y muestra un mensaje de confirmación
RESULTADO OBTENIDO	<pre>{ "message": "Monta creada exitosamente" }</pre>
ESTADO	Aprobado

Evidencia

POST ▾ http://localhost:3001/monta/new

Send ▾

201 Created 6.02 s 39 B

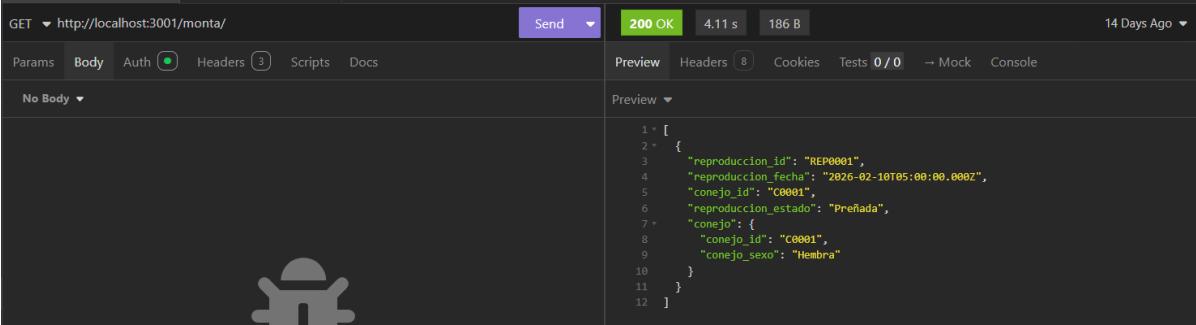
Params	Body (1)	Auth (1)	Headers (4)	Scripts	Docs
Preview ▾					
Headers (8) Cookies Tests 0 / 0 → Mock Console					
Preview ▾					
1 ▾ { 2 "reproduccion_fecha": "2026-02-10", 3 "conejo_id": "C0001", 4 "reproduccion_estado": "Preñada" 5 }					
1 ▾ { 2 "message": "Monta creada exitosamente" 3 }					

3.37. CP08.1-2 Obtener datos de la monta

IDENTIFICADOR	CP08.1-2
DESCRIPCIÓN	Se obtienen los datos de todas las montas
PRECONDICIONES	El usuario ha iniciado sesión y tiene un token de autorización
DATOS DE ENTRADA	

RESULTADO ESPERADO	El sistema obtiene la información de montas generadas
RESULTADO OBTENIDO	<pre>[{ "reproduccion_id": "REP0001", "reproduccion_fecha": "2026-02-10T05:00:00.000Z", "conejo_id": "C0001", "reproduccion_estado": "Preñada", "conejo": { "conejo_id": "C0001", "conejo_sexo": "Hembra" } }]</pre>
ESTADO	Aprobado

Evidencia



GET ▼ http://localhost:3001/monta/

Send ▼ **200 OK** 4.11 s 186 B 14 Days Ago ▾

Params Body Auth Headers ③ Scripts Docs

No Body ▾

Preview ▾

```

1 + [
2 +   {
3 +     "reproduccion_id": "REP0001",
4 +     "reproduccion_fecha": "2026-02-10T05:00:00.000Z",
5 +     "conejo_id": "C0001",
6 +     "reproduccion_estado": "Preñada",
7 +     "conejo": {
8 +       "conejo_id": "C0001",
9 +       "conejo_sexo": "Hembra"
10    }
11  }
12 ]

```

3.38. CP08.1-3 Obtener datos de la monta por conejo

IDENTIFICADOR	CP08.1-3
DESCRIPCIÓN	Se obtienen los datos de todas las montas por conejo
PRECONDICIONES	El usuario ha iniciado sesión y tiene un token de autorización
DATOS DE ENTRADA	
RESULTADO ESPERADO	El sistema obtiene la información de montas generadas

RESULTADO OBTENIDO	<pre>[{ "reproduccion_id": "REP0001", "reproduccion_fecha": "2026-02-10T05:00:00.000Z", "conejo_id": "C0001", "reproduccion_estado": "Preñada", "conejo": { "conejo_id": "C0001", "conejo_sexo": "Hembra" } }]</pre>
ESTADO	Aprobado

Evidencia

The screenshot shows a REST API tool interface. At the top, it says "GET ▾ http://localhost:3001/monta/find/C0001". Below that, there are tabs for "Params", "Body", "Auth" (which is selected and has a green checkmark), "Headers" (with a count of 3), "Scripts", and "Docs". On the right, there are buttons for "Send", "200 OK", "2.71 s", "265 B", and "Just Now". Under "Auth", there is a dropdown for "Bearer Token" and fields for "ENABLED" (checkbox checked), "TOKEN" (redacted), and "PREFIX" (redacted). On the right, there are buttons for "Preview", "Headers" (with a count of 8), "Cookies", "Tests 0 / 0", "Mock", and "Console". The "Preview" section shows the JSON response:

```

1 ↴ [
2 ↴   {
3 ↴     "reproduccion_id": "REP0002",
4 ↴     "reproduccion_fecha": "2026-02-15T15:00:00.000Z",
5 ↴     "conejo_id": "C0001",
6 ↴     "reproduccion_estado": "Programada"
7 ↴   },
8 ↴   {
9 ↴     "reproduccion_id": "REP0001",
10    "reproduccion_fecha": "2026-02-10T05:00:00.000Z",
11    "conejo_id": "C0001",
12    "reproduccion_estado": "Preñada"
13  }
14 ]

```

3.39. CP08.1-4 Registrar datos de la monta

IDENTIFICADOR	CP08.1-4
DESCRIPCIÓN	Se obtienen los datos de todas las montas
PRECONDICIONES	El usuario ha iniciado sesión y tiene un token de autorización
DATOS DE ENTRADA	
RESULTADO ESPERADO	El sistema obtiene la información de montas generadas por id

RESULTADO OBTENIDO	<p>200 OK 2.02 s 211 B</p> <p>review Headers (8) Cookies Tests 0 / 0 → Mock</p> <pre>review ▾ 1 ▾ { 2 "reproduccion_id": "REP0001", 3 "reproduccion_fecha": "2026-02-10T05:00:00.000Z", 4 "conejo_id": "C0001", 5 "reproduccion_estado": "Preñada", 6 "conejo": { 7 "conejo_id": "C0001", 8 "conejo_sexo": "Hembra", 9 "conejo_estado": "Preñada" 10 } 11 }</pre>
ESTADO	Aprobado

Evidencia

GET ▾ http://localhost:3001/monta/REP0001

Send ▾ **200 OK** 2.02 s 211 B Just Now ▾

Params	Body	Auth <input checked="" type="checkbox"/>	Headers (3)	Scripts	Docs
--------	------	--	-------------	---------	------

Preview ▾

Bearer Token ▾

ENABLED

TOKEN

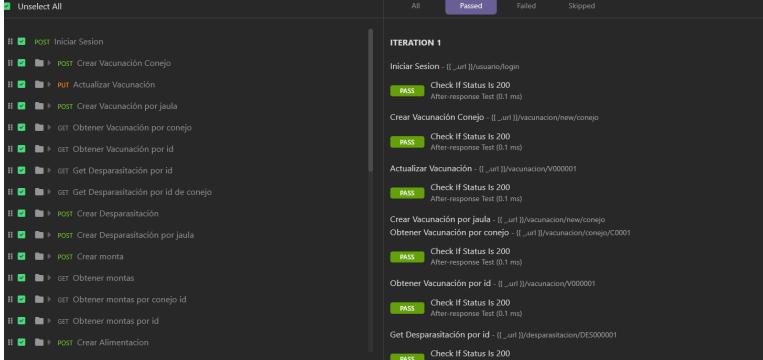
PREFIX

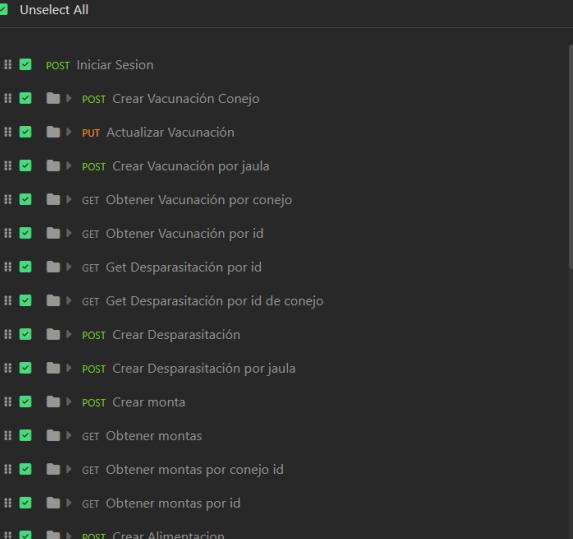
1 ▾ {
 2 "reproduccion_id": "REP0001",
 3 "reproduccion_fecha": "2026-02-10T05:00:00.000Z",
 4 "conejo_id": "C0001",
 5 "reproduccion_estado": "Preñada",
 6 "conejo": {
 7 "conejo_id": "C0001",
 8 "conejo_sexo": "Hembra",
 9 "conejo_estado": "Preñada"
 10 }
 11 }

4. Pruebas de Flujo

4.1. CP09.1-1 Flujo Sistema Backend Completo

IDENTIFICADOR	CP08.1-4
DESCRIPCIÓN	Se probó todas las funcionalidades del sistema
PRECONDICIONES	
DATOS DE ENTRADA	

RESULTADO ESPERADO	El sistema obtiene la información de montas generadas por id
RESULTADO OBTENIDO	
ESTADO	Aprobado

	All Passed Failed Skipped
ITERATION 1	

INICIO DE SESIÓN - {{ _url }} /usuario/login

PASS Check If Status Is 200
After-response Test (0.1 ms)

CREAR VACUNACIÓN CONEJO - {{ _url }} /vacunacion/new/conejo

PASS Check If Status Is 200
After-response Test (0.1 ms)

ACTUALIZAR VACUNACIÓN - {{ _url }} /vacunacion/V000001

PASS Check If Status Is 200
After-response Test (0.1 ms)

CREAR VACUNACIÓN EN JAUZA - {{ _url }} /vacunacion/new/jaula

PASS Check If Status Is 200
After-response Test (0.1 ms)

OBTENER VACUNACIÓN POR ID - {{ _url }} /vacunacion/conejo/C0001

PASS Check If Status Is 200
After-response Test (0.1 ms)

OBTENER VACUNACIÓN POR CONEJO ID - {{ _url }} /vacunacion/V000001

PASS Check If Status Is 200
After-response Test (0.1 ms)

OBTENER MONTAS POR ID - {{ _url }} /desparasitacion/DES000001

PASS Check If Status Is 200
After-response Test (0.1 ms)

4.2. CP9.1-2 Flujo Sistema Backend Completo y excepciones

IDENTIFICADOR	CP08.1-4
DESCRIPCIÓN	Se probó todas las funcionalidades del sistema y comprobó que se cumplan las detenciones correspondientes
PRECONDICIONES	
DATOS DE ENTRADA	
RESULTADO ESPERADO	El sistema obtiene la información de montas generadas por id

RESULTADO OBTENIDO

	<p>ITERATION 1</p> <p>Iniciar Sesion - {{ _url }}/usuario/login Crear Vacunación Conejo - {{ _url }}/vacunacion/new/conejo Actualizar Vacunación - {{ _url }}/vacunacion/V000001 Crear Vacunación por jaula - {{ _url }}/vacunacion/new/conejo</p> <p>FAIL Check If Status Is 200 error: AssertionError: expected 400 to deeply equal 201 ACTUAL: After-response Test (0.3 ms)</p> <p>Obtener Vacunación por conejo - {{ _url }}/vacunacion/conejo/C0001 Obtener Vacunación por id - {{ _url }}/vacunacion/V000001 Get Desparasitación por id - {{ _url }}/desparasitacion/DES000001 Get Desparasitación por id de conejo - {{ _url }}/desparasitacion/conejo/C0001 Crear Desparasitación - {{ _url }}/desparasitacion/new</p> <p>FAIL Check If Status Is 200 error: AssertionError: expected 200 to deeply equal 201 ACTUAL: After-response Test (0.1 ms)</p> <p>Crear Desparasitación por jaula - {{ _url }}/desparasitacion/new_cage</p> <p>FAIL Check If Status Is 200 error: AssertionError: expected 200 to deeply equal 201 ACTUAL: After-response Test (0.2 ms)</p> <p>Crear monta - {{ _url }}/monta/new</p> <p>FAIL Check If Status Is 201 error: AssertionError: expected 201 to deeply equal 200 ACTUAL: After-response Test (0.5 ms)</p> <p>-</p>
ESTADO	Aprobado

Evidencia

Request Order	Advanced	Tests 14 / 20	History	Console	
		All	Passed	Failed	Skipped
<input checked="" type="checkbox"/> Unselect All					
# <input checked="" type="checkbox"/> POST Iniciar Sesion					
# <input checked="" type="checkbox"/> POST Crear Vacunación Conejo					
# <input checked="" type="checkbox"/> PUT Actualizar Vacunación					
# <input checked="" type="checkbox"/> POST Crear Vacunación por jaula					
# <input checked="" type="checkbox"/> GET Obtener Vacunación por conejo					
# <input checked="" type="checkbox"/> GET Obtener Vacunación por id					
# <input checked="" type="checkbox"/> GET Get Desparasitación por id					
# <input checked="" type="checkbox"/> GET Get Desparasitación por id de conejo					
# <input checked="" type="checkbox"/> POST Crear Desparasitación					
# <input checked="" type="checkbox"/> POST Crear Desparasitación por jaula					
# <input checked="" type="checkbox"/> POST Crear monta					
# <input checked="" type="checkbox"/> GET Obtener montas					
# <input checked="" type="checkbox"/> GET Obtener montas por conejo id					
# <input checked="" type="checkbox"/> GET Obtener montas por id					
# <input checked="" type="checkbox"/> POST Crear Alimentacion					

ITERATION 1

Iniciar Sesion - {{ _url }}/usuario/login
 Crear Vacunación Conejo - {{ _url }}/vacunacion/new/conejo
 Actualizar Vacunación - {{ _url }}/vacunacion/V000001
 Crear Vacunación por jaula - {{ _url }}/vacunacion/new/conejo

FAIL Check If Status Is 200 | error: AssertionError: expected 400 to deeply equal 201 | ACTUAL:
 After-response Test (0.3 ms)

Obtener Vacunación por conejo - {{ _url }}/vacunacion/conejo/C0001
 Obtener Vacunación por id - {{ _url }}/vacunacion/V000001
 Get Desparasitación por id - {{ _url }}/desparasitacion/DES000001
 Get Desparasitación por id de conejo - {{ _url }}/desparasitacion/conejo/C0001
 Crear Desparasitación - {{ _url }}/desparasitacion/new

FAIL Check If Status Is 200 | error: AssertionError: expected 200 to deeply equal 201 | ACTUAL:
 After-response Test (0.1 ms)

Crear Desparasitación por jaula - {{ _url }}/desparasitacion/new_cage

FAIL Check If Status Is 200 | error: AssertionError: expected 200 to deeply equal 201 | ACTUAL:
 After-response Test (0.2 ms)

Crear monta - {{ _url }}/monta/new

FAIL Check If Status Is 201 | error: AssertionError: expected 201 to deeply equal 200 | ACTUAL:
 After-response Test (0.5 ms)

-