

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
2	05/02/2026

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. CP08.1-1 Obtener datos de todas las montas.....	20
3.18. CP08.1-2 Obtener datos de todas las montas por conejo.....	21
3.19. CP08.1-3 Obtener datos de todas las montas.....	21
3.20. CP08.1-4 Probar todas las funcionalidades del sistema.....	21
3.21. CP08.1-1 (Registro de datos de la monta).....	22
3.22. CP08.1-2 (Obtener datos de la monta).....	22
3.23. CP08.1-3 (Obtener datos de la monta por conejo).....	23
3.24. CP08.1-4 (Registrar datos de la monta).....	23
3.25. CP08.1-1 Obtener datos de todas las montas.....	24
3.26. CP08.1-2 Obtener datos de todas las montas por conejo.....	24
3.27. CP08.1-3 Obtener datos de todas las montas.....	25
3.28. CP08.1-4 Probar todas las funcionalidades del sistema.....	25
3.29. CP08.1-1 Obtener datos de todas las montas.....	25
3.30. CP08.1-2 Obtener datos de todas las montas por conejo.....	26
3.31. CP08.1-3 Obtener datos de todas las montas.....	26
3.32. CP08.1-4 Probar todas las funcionalidades del sistema.....	27
3.33. CP08.1-1 Obtener datos de todas las montas.....	27

3.34. CP08.1-2 Obtener datos de todas las montas por conejo.....	28
3.35. CP08.1-3 Obtener datos de todas las montas.....	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.

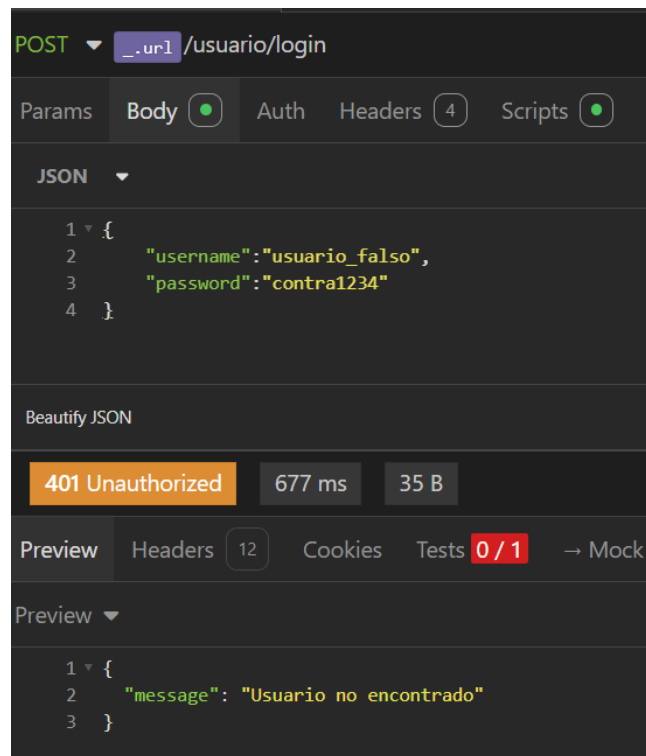
3. Casos de Prueba

3.1. CP01.1-1 Inicio de Sesión

IDENTIFICADOR	CP01.1-1
---------------	----------

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



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

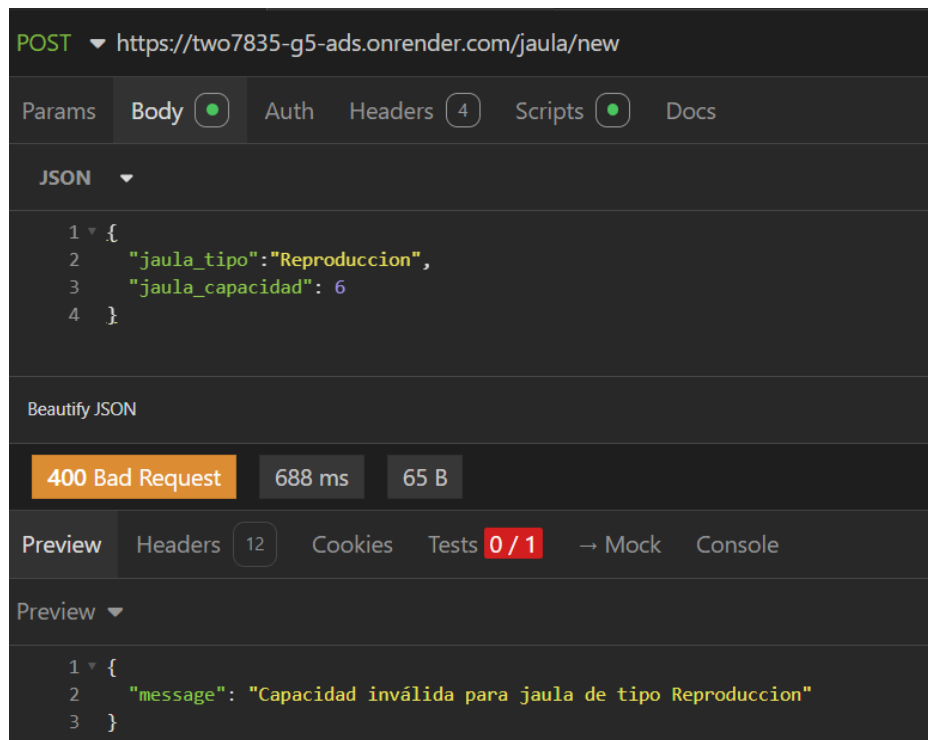
PRECONDICIONES	El usuario ha iniciado sesión y tiene un token de autorización
DATOS DE ENTRADA	<pre>{ "jaula_tipo": "Engorde", "jaula_capacidad": 6 }</pre>
RESULTADO ESPERADO	El sistema crea la jaula exitosamente y muestra un mensaje de confirmación
RESULTADO OBTENIDO	El sistema muestra un mensaje de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

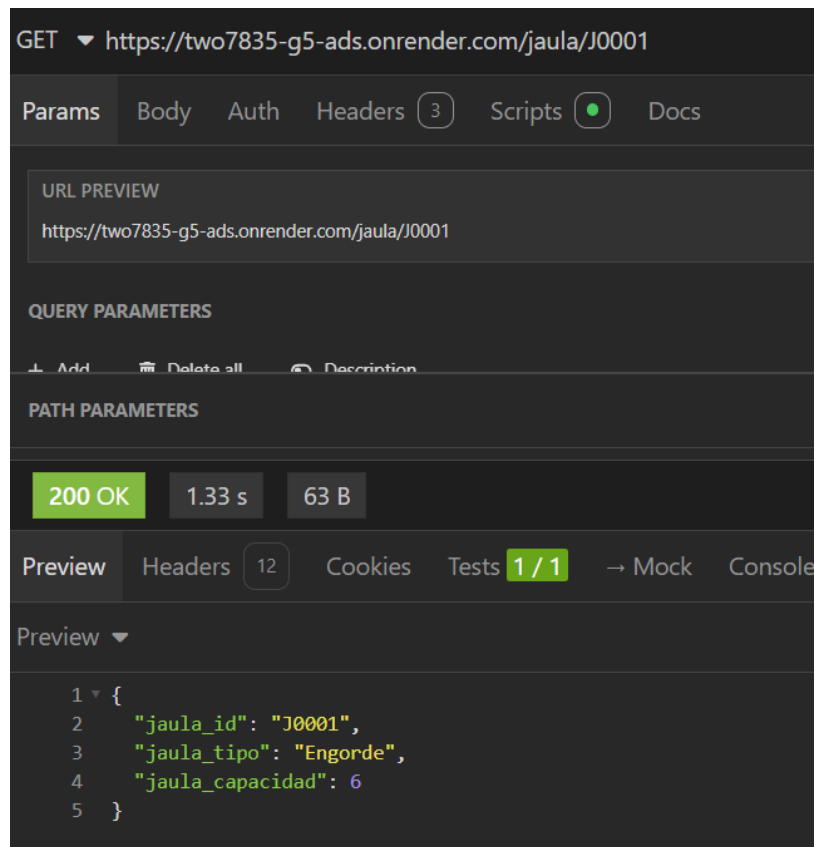
Evidencia



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

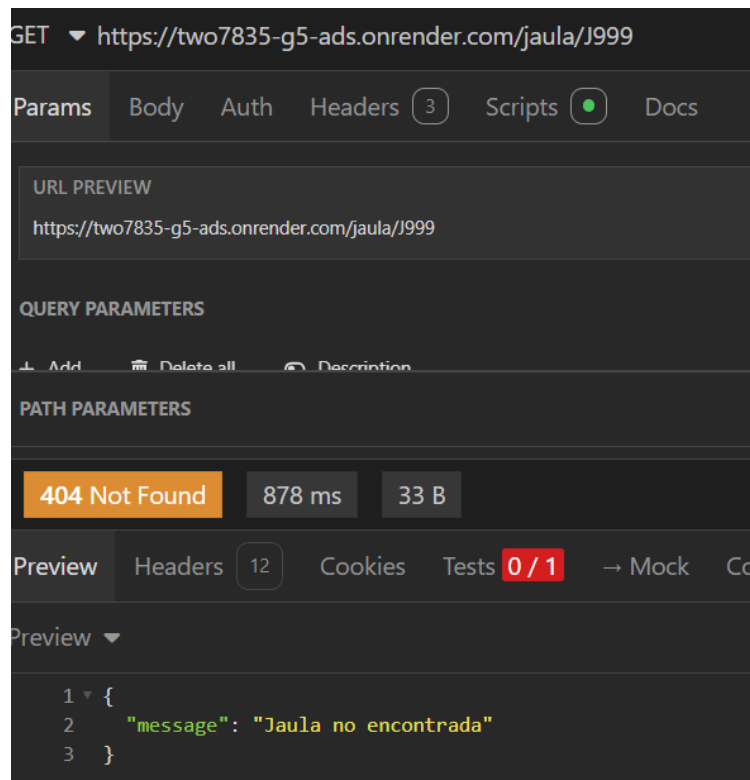
Evidencia



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

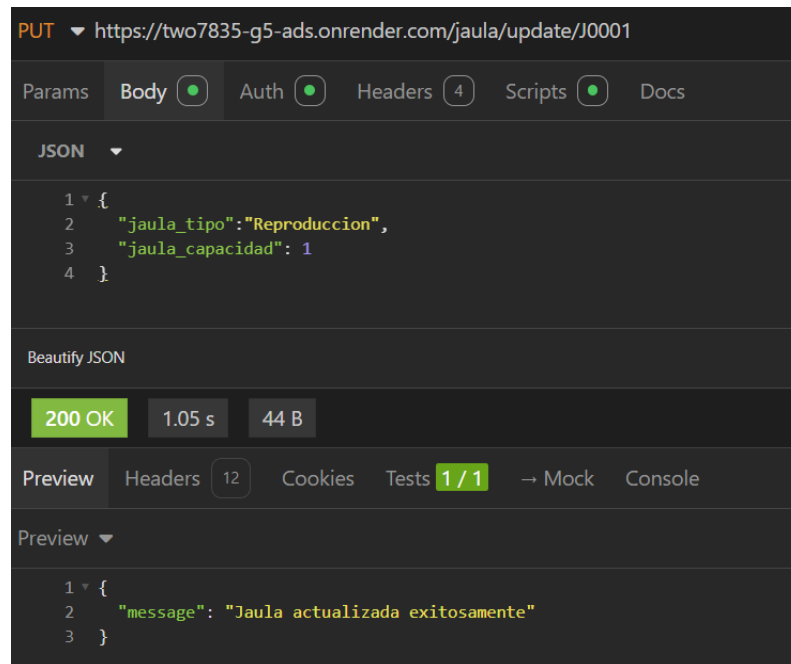


3.7. CP02.3-1 - Modificar de datos de un conejo

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	<p>Parámetros: ID de la jaula a modificar</p> <p><code>https://two7835-g5-ads.onrender.com/jaula/update/J0001</code></p> <p>Cuerpo: Nuevos datos</p> <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



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

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

Params Body Auth Headers 4 Scripts Docs

JSON ▼

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

Beautify JSON

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>

Params Body Auth Headers 3 Scripts Docs

No Body ▼

200 OK 582 ms 68 B

Preview Headers 12 Cookies Tests 1 / 1 → Mock

Preview ▼

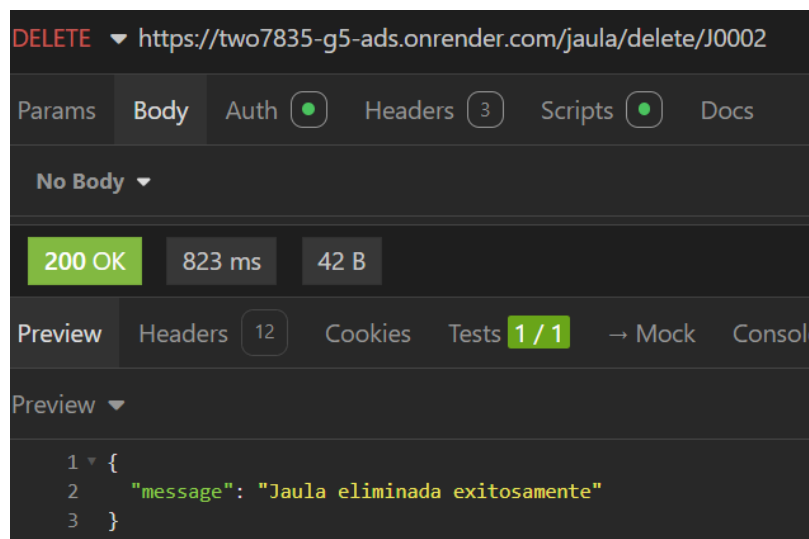
```
1 {
2   "jaula_id": "J0001",
3   "jaula_tipo": "Reproduccion",
4   "jaula_capacidad": 1
5 }
```

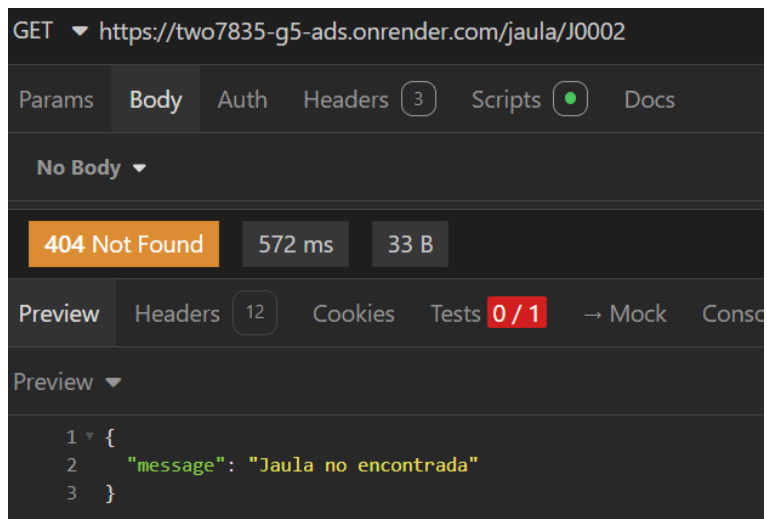
3.9. CP02.4-1 - Eliminar una jaula

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 DELETE ▼ https://two7835-g5-ads.onrender.com/jaula/delete/J0002
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





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

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

Params Body Auth Headers Scripts Docs

No Body

400 Bad Request 1.12 s 118 B

Preview Headers 12 Cookies Tests 0/1 → Mock Console

Preview

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

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

Params Body Auth Headers Scripts Docs

No Body

200 OK 593 ms 68 B

Preview Headers 12 Cookies Tests 1/1 → Mock Console

Preview

```
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 REST client interface with the following details:

- Method:** POST
- URL:** `_.url /raza/new`
- Body (JSON):**

```

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 }

```
- Status:** 201 Created
- Time:** 403 ms
- Size:** 476 B
- Preview (JSON):**

```

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 4 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. CP04.1 1 - 1 Registro de raza

IDENTIFICADOR	CP04.1 1 - 1
DESCRIPCIÓN	Registro de raza con todos los datos ingresados de forma correcta
PRECONDICIONES	El usuario ha iniciado sesión y tiene un token de autorización
DATOS DE ENTRADA	<pre>{ "conejo_raza_nombre": "Nube", "conejo_raza_descripcion": "Un conejito blanco y esponjoso, curioso y tranquilo, que adora mordisquear zanahorias y echar siestas al sol"</pre>

	}
RESULTADO ESPERADO	El sistema devuelve un mensaje de raza creada exitosamente
RESULTADO OBTENIDO	El sistema muestra los datos de la raza creada
ESTADO	Ejecutado

POST
.url /raza/new
Send

Params
Body
Auth
Headers 4
Scripts
Docs

JSON

```

1 {
2   "conejo_raza_nombre": "Nube",
3   "conejo_raza_descripcion": "Un conejito blanco y esponjoso, curioso y
4   tranquilo, que adora mordisquear zanahorias y echar siestas al sol."
5 }

```

Beautify JSON

201 Created
1.47 s
241 B
Just Now

Preview
Headers 12
Cookies
Tests 1 / 1
Mock
Console

Preview

```

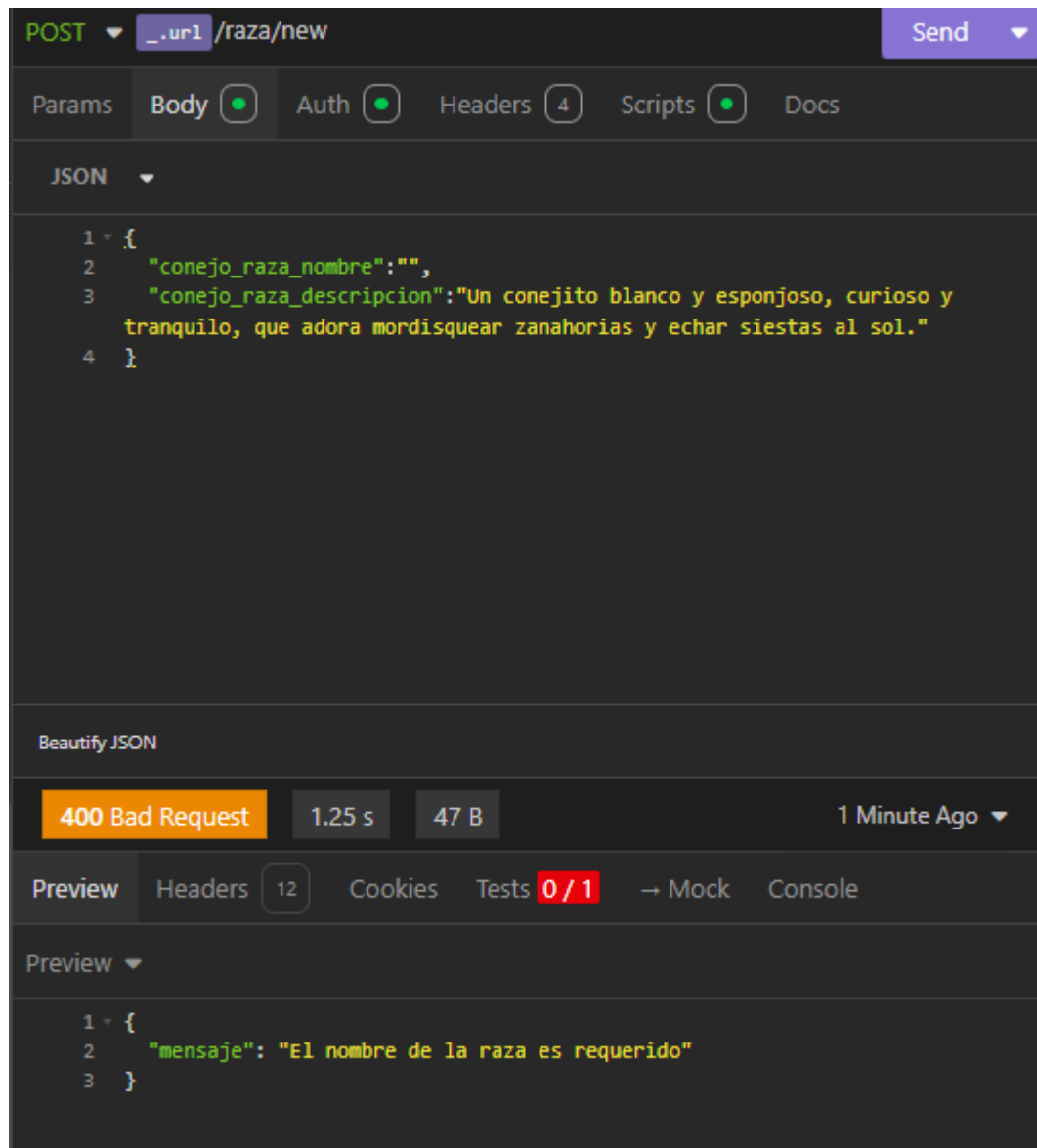
1 {
2   "mensaje": "Raza creada exitosamente",
3   "raza": {
4     "conejo_raza_id": "CRZ00004",
5     "conejo_raza_nombre": "Nube",
6     "conejo_raza_descripcion": "Un conejito blanco y esponjoso, curioso
7     y tranquilo, que adora mordisquear zanahorias y echar siestas al sol."
8   }
9 }

```

\$.store.books[*].author

3.18. CP04.1 1- 2 Registro de raza

IDENTIFICADOR	CP04.1 1 - 2
DESCRIPCIÓN	Validación del sistema ante un ingreso de datos incompletos
PRECONDICIONES	El usuario ha iniciado sesión y tiene un token de autorización
DATOS DE ENTRADA	<pre>1 { 2 "conejo_raza_nombre":""," 3 "conejo_raza_descripcion":"Un conejito blanco y esponjoso, curioso y tranquilo, que adora mordisquear zanahorias y echar siestas al sol." 4 }</pre>
RESULTADO ESPERADO	Mensaje y validacion de error por parte del sistema
RESULTADO OBTENIDO	Mensaje de validación por parte del sistema indicando que el nombre es requerido
ESTADO	Ejecutado



3.19. CP04.1 1 -3 Registro duplicado

IDENTIFICADOR	CP04.1 1 - 3
DESCRIPCIÓN	Validación del sistema ante un ingreso de raza duplicado
PRECONDICIONES	El usuario ha iniciado sesión y tiene un token de autorización
DATOS DE ENTRADA	<pre>{ "conejo_raza_nombre": "Nube", "conejo_raza_descripcion": "Un conejito blanco y esponjoso, curioso y tranquilo, que adora mordisquear zanahorias y echar siestas al sol." }</pre>
RESULTADO ESPERADO	El sistema expone un error y bloquea el registro

RESULTADO OBTENIDO	El sistema lanza el mensaje de que ya existe una raza con el nombre 'Nube'
ESTADO	Ejecutado

POST `_.url /raza/new` Send

Params Body Auth Headers 4 Scripts Docs

JSON

```

1 {
2   "conejo_raza_nombre": "Nube",
3   "conejo_raza_descripcion": "Un conejito blanco y esponjoso, curioso y
4   tranquilo, que adora mordisquear zanahorias y echar siestas al sol."
5 }

```

Beautify JSON

400 Bad Request 2.58 s 95 B 4 Minutes Ago

Preview Headers 12 Cookies Tests 0/1 → Mock Console

Preview

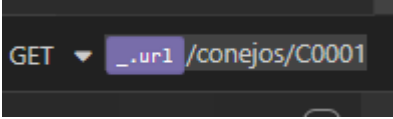
```

1 {
2   "mensaje": "Ya existe una raza con el nombre \"Nube\". Los nombres de
3   raza deben ser únicos."
4 }

```

3.20. CP04.2 1- 1 Búsqueda conejos

IDENTIFICADOR	CP04.2 1- 1
DESCRIPCIÓN	Ingreso de ID y búsqueda de un conejo existente
PRECONDICIONES	El usuario ha iniciado sesión y tiene un token de autorización

DATOS DE ENTRADA	 endpoint/IDconejo
RESULTADO ESPERADO	El sistema proporciona todos los datos del conejo seleccionado
RESULTADO OBTENIDO	El sistema proporciona todos los datos del conejo seleccionado
ESTADO	Ejecutado

GET

_url

 /conejos/C0001

Send

Params

Body

Auth

Headers3

Scripts

Docs

+ Add

Delete all

Description

Accept

/

Host

<calculated at runtime>

User-Agent

insomnia/11.3.0

Bulk Edit

200 OK

943 ms

285 B

Just Now

Preview

Headers12

Cookies

Tests1 / 1

→ Mock

Console

Preview

1 {

2 "conejo_id": "C0001",

3 "conejo_sexo": "Hembra",

4 "conejo_edad": 4,

5 "conejo_peso": "2.85",

6 "conejo_proposito": "Reproduccion",

7 "conejo_fecha_nacimiento": "2025-06-15T00:00:00.000Z",

8 "jaula_id": "J0001",

9 "conejo_raza_id": "CRZ00001",

10 "conejo_estado": "Preñada",

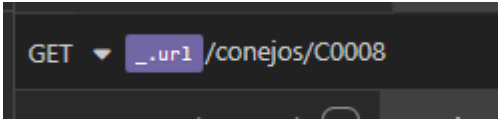
11 "raza": {

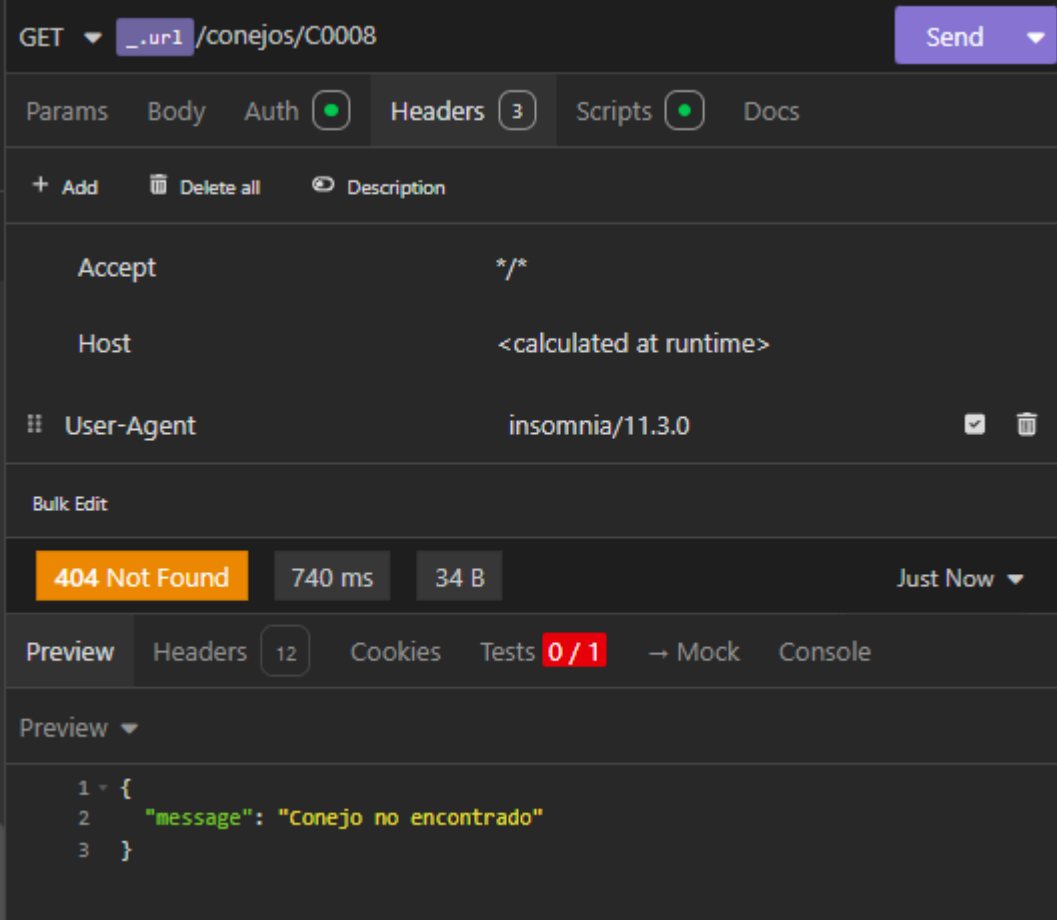
12 "conejo_raza_nombre": "CALIFORNIA"

13 }

14 }

3.21. CP04.2 1 - 1 Búsqueda conejos

IDENTIFICADOR	CP04.2 1 - 2
DESCRIPCIÓN	Mensaje de error por parte del sistema al momento de ingresar a la búsqueda un ID de un conejo inexistente
PRECONDICIONES	El usuario ha iniciado sesión y tiene un token de autorización
DATOS DE ENTRADA	endpoint/IDconejo 
RESULTADO ESPERADO	El sistema lanza un mensaje sobre que el conejo no existe
RESULTADO OBTENIDO	El sistema lanza un mensaje sobre que el conejo no fue encontrado
ESTADO	Ejecutado



GET `_url_/conejos/C0008` Send

Params Body Auth ● Headers 3 Scripts ● Docs

+ Add 🗑️ Delete all 📄 Description

Accept `*/*`

Host `<calculated at runtime>`

⚙️ User-Agent `insomnia/11.3.0` ☑️ 🗑️

Bulk Edit

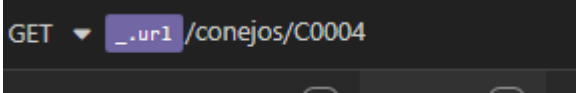
404 Not Found 740 ms 34 B Just Now ▼

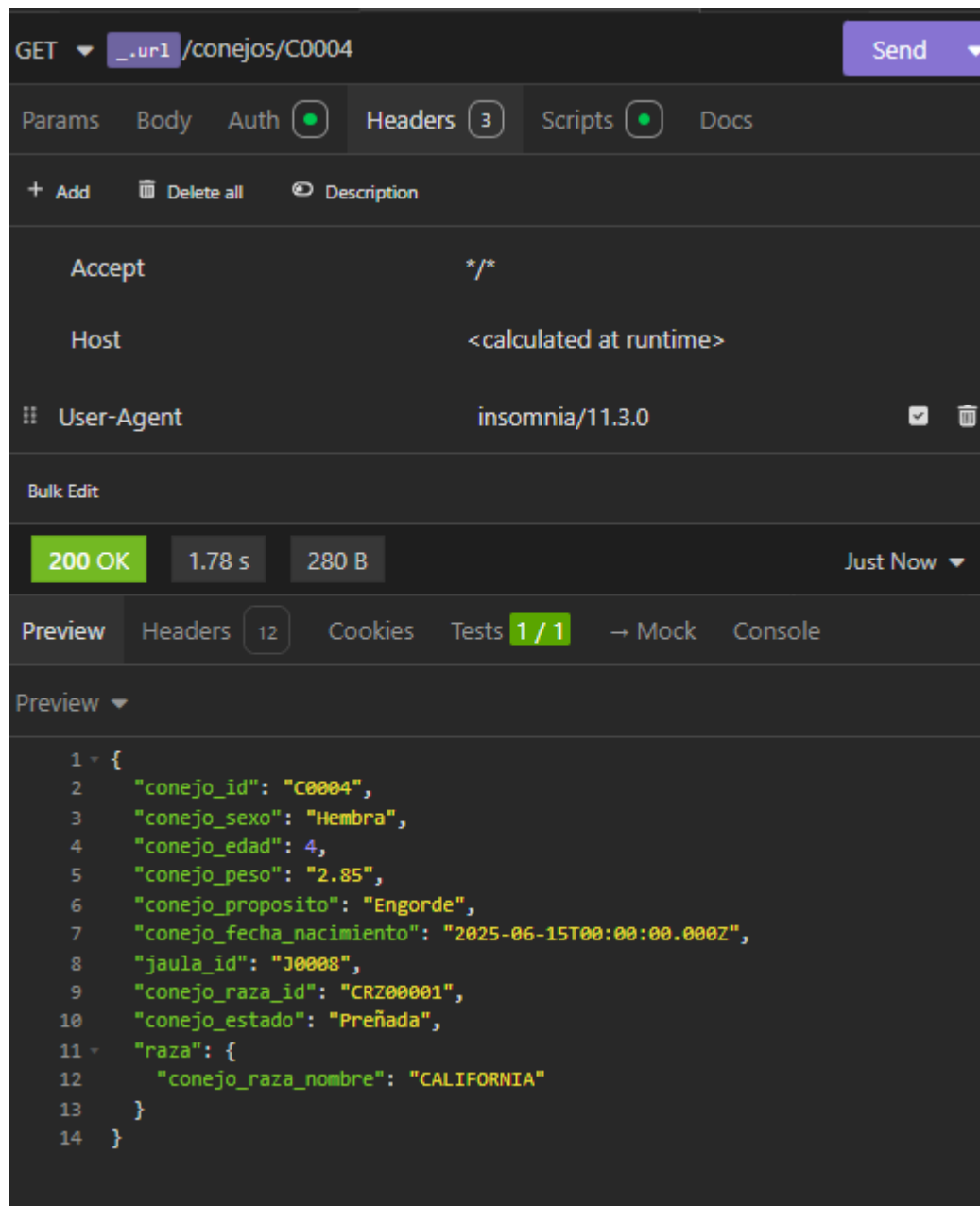
Preview Headers 12 Cookies Tests 0/1 → Mock Console

Preview ▼

```
1 {  
2   "message": "Conejo no encontrado"  
3 }
```

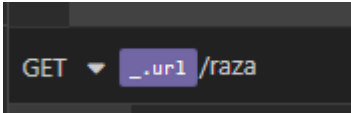
3.22. CP04.2 1- 3 Búsqueda conejos

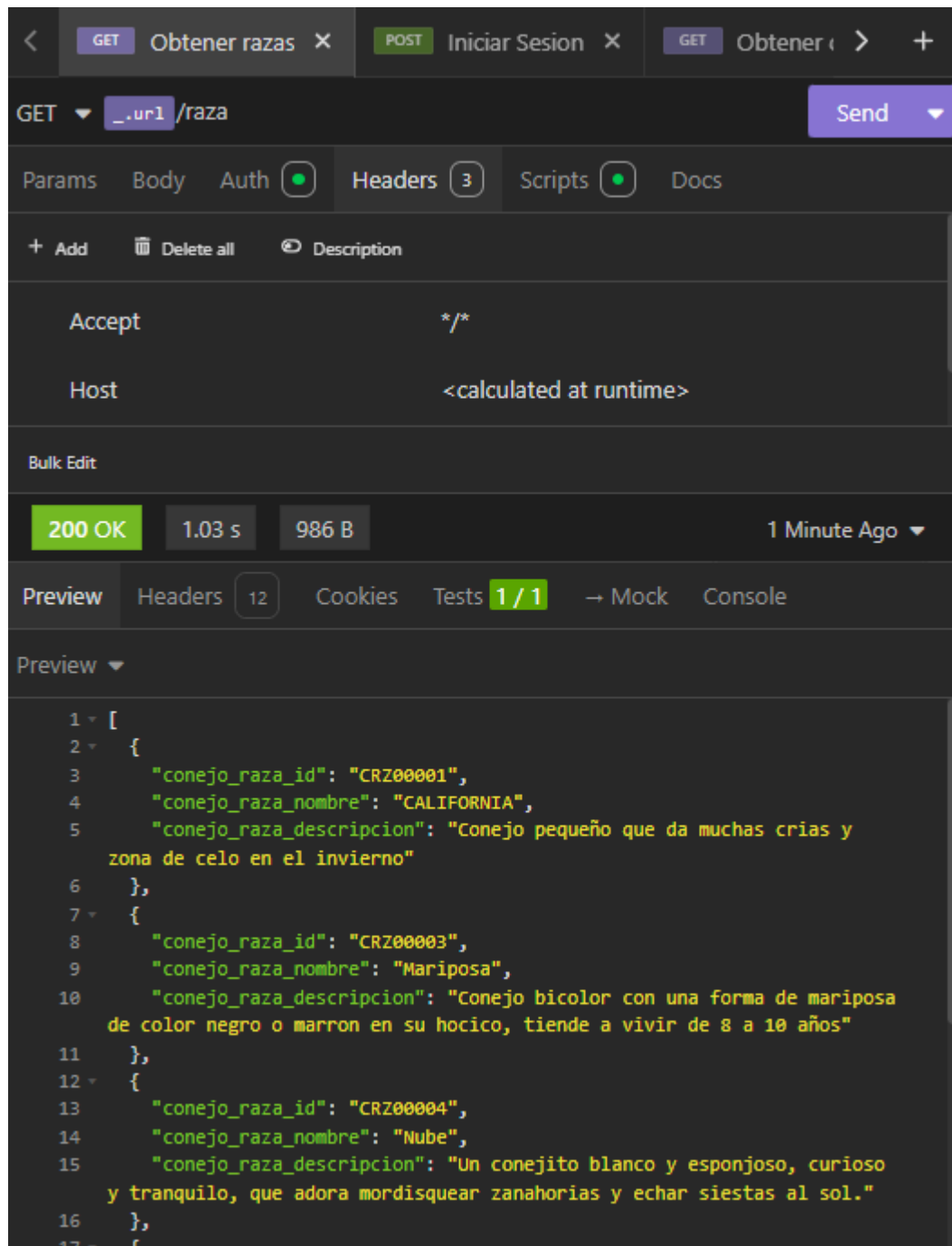
IDENTIFICADOR	CP04.1 - 3
DESCRIPCIÓN	Verificación de impresión de datos en el formato deseado
PRECONDICIONES	El usuario ha iniciado sesión y tiene un token de autorización
DATOS DE ENTRADA	 endpoint/id
RESULTADO ESPERADO	El sistema imprime los datos en el formato Letra + 3 dígitos
RESULTADO OBTENIDO	El sistema imprime los datos en el formato Letra + 3 dígitos
ESTADO	Ejecutado



3.23. CP04.3 1- 1 Búsqueda de razas

IDENTIFICADOR	CP04.3 1 - 1
DESCRIPCIÓN	Verificar el despliegue completo de todas las razas guardadas
PRECONDICIONES	El usuario ha iniciado sesión y tiene un token de autorización
DATOS DE ENTRADA	/endpoint

	
RESULTADO ESPERADO	El sistema muestra todas las razas guardadas en el formato deseado
RESULTADO OBTENIDO	<p>Los datos se imprimen en el formato deseado</p> <pre>{ "conejo_raza_id": "CRZ00001", "conejo_raza_nombre": "CALIFORNIA", "conejo_raza_descripcion": "Conejo pequeño que da muchas crías y zona de celo en el invierno" }</pre>
ESTADO	Ejecutado



3.24. CP04.3 1 - 2 Búsqueda de razas

IDENTIFICADOR	CP04.3 1-2
DESCRIPCIÓN	Impresión única de los datos de una raza seleccionada
PRECONDICIONES	El usuario ha iniciado sesión y tiene un token de autorización.

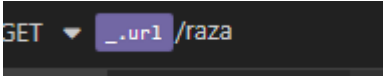
DATOS DE ENTRADA	/endpoint/IDRaza
RESULTADO ESPERADO	El sistema muestra los datos de la raza seleccionada
RESULTADO OBTENIDO	<p>El sistema muestra los datos de la raza seleccionada</p> <pre>{ "conejo_raza_id": "CRZ00003", "conejo_raza_nombre": "Mariposa", "conejo_raza_descripcion": "Conejo bicolor con una forma de mariposa de color negro o marron en su hocico, tiende a vivir de 8 a 10 años" }</pre>
ESTADO	Ejecutado

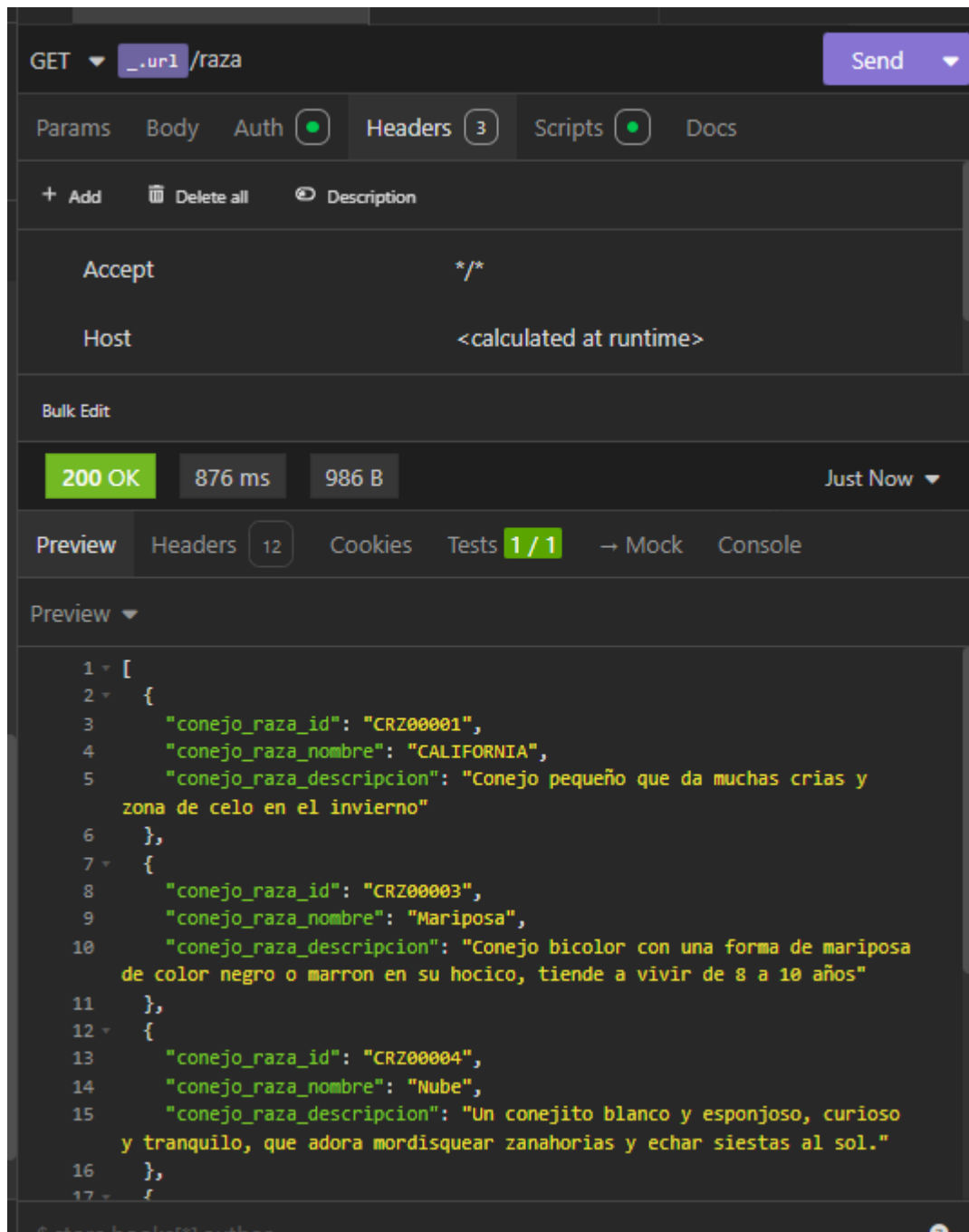
The screenshot shows a REST client interface with the following details:

- Method:** GET
- URL:** `_.url /raza/CRZ00003`
- Buttons:** Send, Params, Body, Auth, Headers (3), Scripts, Docs.
- Headers:**
 - Accept: `*/*`
 - Host: `<calculated at runtime>`
- Status Bar:** 200 OK, 899 ms, 199 B, Just Now.
- Preview:** Headers (12), Cookies, Tests (1 / 1), Mock, Console.
- Response Body (JSON):**

```
1 {
2   "conejo_raza_id": "CRZ00003",
3   "conejo_raza_nombre": "Mariposa",
4   "conejo_raza_descripcion": "Conejo bicolor con una forma de mariposa de
   color negro o marron en su hocico, tiende a vivir de 8 a 10 años"
5 }
```

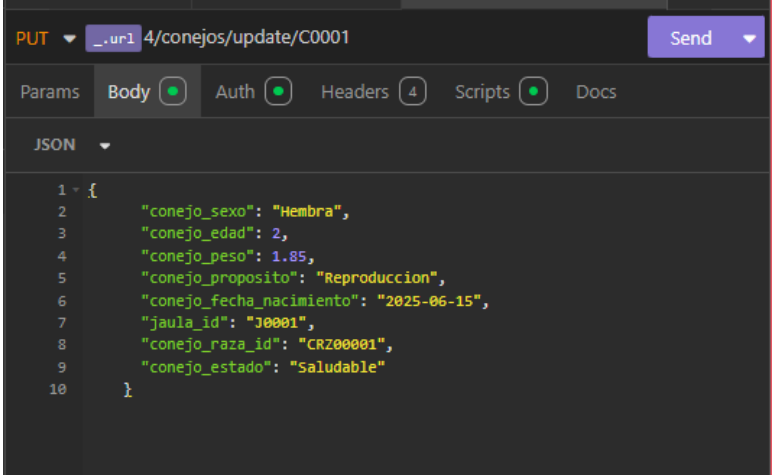
3.25. CP04.3 1 - 3 Búsqueda de razas

IDENTIFICADOR	CP04.3 1-3
DESCRIPCIÓN	Correcta desactivación del filtro de búsqueda de las razas de conejo
PRECONDICIONES	El usuario ha iniciado sesión y tiene un token de autorización.
DATOS DE ENTRADA	/endpoint 
RESULTADO ESPERADO	El sistema muestra nuevamente todas las razas de conejo guardadas con normalidad una vez se desactive el filtro
RESULTADO OBTENIDO	El sistema muestra nuevamente todas las razas de conejo guardadas con normalidad una vez se desactive el filtro
ESTADO	Ejecutado



3.26. CP04.4 1 - 1 Modificar conejo

IDENTIFICADOR	CP04.4 1-1
DESCRIPCIÓN	Modificación de los datos de un conejo seleccionado
PRECONDICIONES	El usuario ha iniciado sesión y tiene un token de autorización.

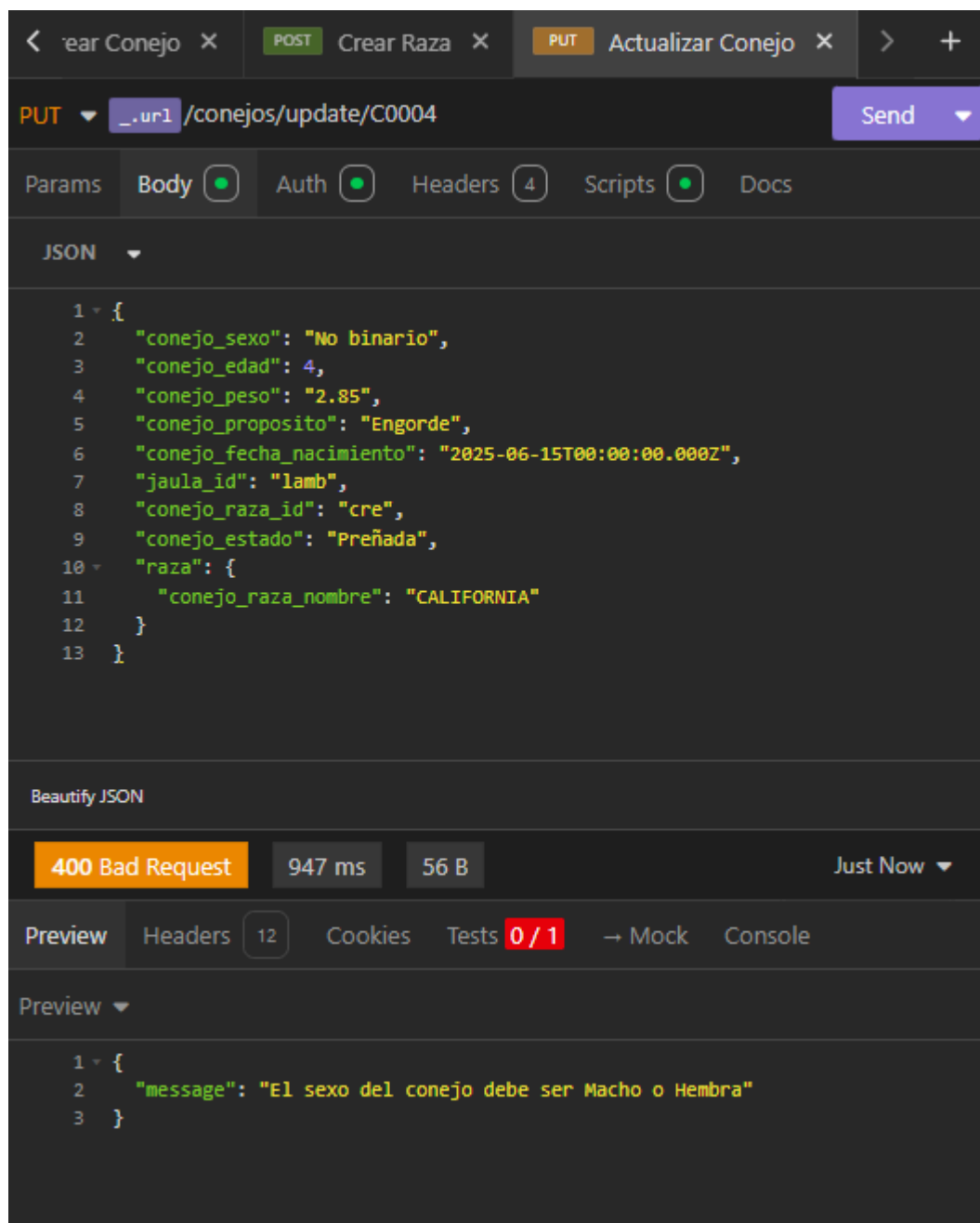
DATOS DE ENTRADA	
RESULTADO ESPERADO	El sistema debe mostrar los datos actualizados del conejo
RESULTADO OBTENIDO	Los datos se actualizan correctamente y se muestran
ESTADO	Ejecutado

```
{
  "conejo_sexo": "Hembra",
  "conejo_edad": 2,
  "conejo_peso": 1.85,
  "conejo_proposito": "Reproduccion",
  "conejo_fecha_nacimiento": "2025-06-15",
  "jaula_id": "J0001",
  "conejo_raza_id": "CRZ00001",
  "conejo_estado": "Saludable"
}
```

3.27. CP04.4 1- 2 Modificar conejo

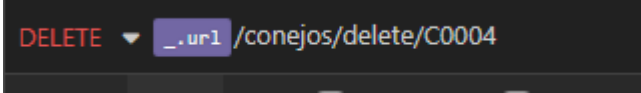
IDENTIFICADOR	CP04.4 1-2
DESCRIPCIÓN	Selección de un conejo y modificación de datos de manera incorrecta.
PRECONDICIONES	El usuario ha iniciado sesión y tiene un token de autorización.

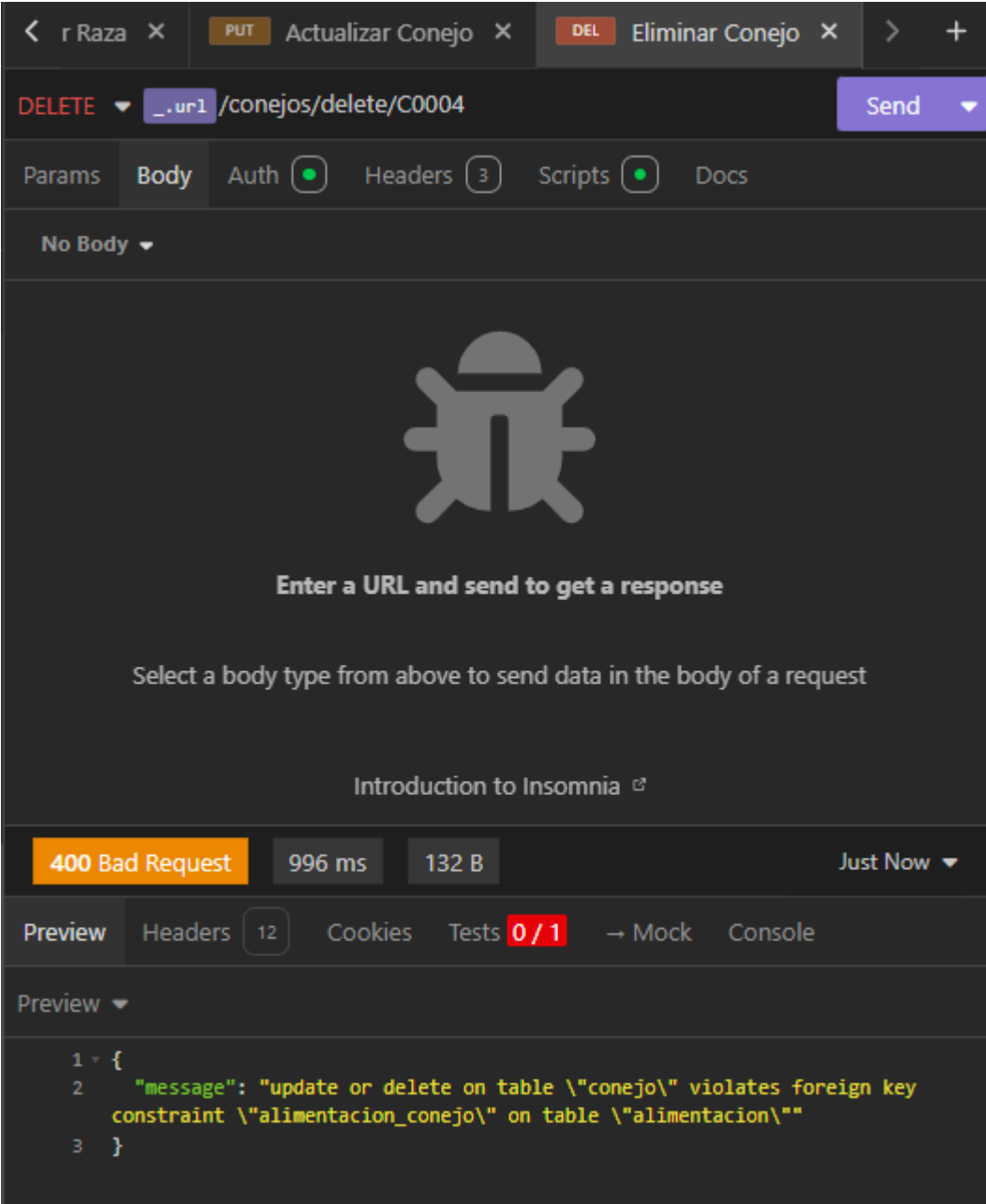
DATOS DE ENTRADA	
RESULTADO ESPERADO	Mensaje de validación indicando el dato ingresado de manera incorrecta
RESULTADO OBTENIDO	Mensaje con ingreso incorrecto respecto al sexo del conejo
ESTADO	Ejecutado



3.28. CP04.5 1 - 1 Eliminar Conejo

IDENTIFICADOR	CP04.5 1-1
DESCRIPCIÓN	Eliminación de un conejo que pertenece a una jaula
PRECONDICIONES	El usuario ha iniciado sesión y tiene un token de autorización.
DATOS DE ENTRADA	/endpoint/IdConejo

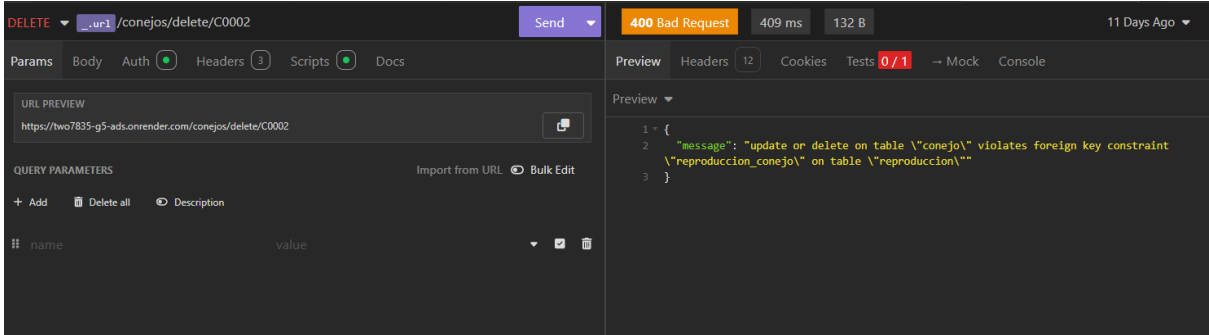
	
RESULTADO ESPERADO	El sistema muestra un mensaje de error e impide realizar la acción
RESULTADO OBTENIDO	Mensaje de error
ESTADO	Ejecutado



The screenshot shows the Insomnia REST client interface. At the top, there are tabs for different requests: 'r Raza', 'Actualizar Conejo', and 'Eliminar Conejo'. The 'Eliminar Conejo' tab is active, showing a DELETE request to the URL `_.url /conejos/delete/C0004`. Below the URL bar, there are tabs for 'Params', 'Body', 'Auth', 'Headers', 'Scripts', and 'Docs'. The 'Body' tab is selected, showing 'No Body'. The main area displays a large grey bug icon and the text 'Enter a URL and send to get a response'. Below this, it says 'Select a body type from above to send data in the body of a request' and provides a link to 'Introduction to Insomnia'. At the bottom, the response status is '400 Bad Request' with a response time of '996 ms' and a size of '132 B'. The response body is shown in the 'Preview' tab, displaying a JSON error message: `{ "message": "update or delete on table \"conejo\" violates foreign key constraint \"alimentacion_conejo\" on table \"alimentacion\""`.

3.29. CP04.5 1 - 2 Eliminar Conejo

IDENTIFICADOR	CP04.5 1-2
DESCRIPCIÓN	Eliminación de un conejo que no pertenece a ninguna jaula
PRECONDICIONES	El usuario ha iniciado sesión y tiene un token de autorización.
DATOS DE ENTRADA	/endpoint/IDConejo
RESULTADO ESPERADO	El sistema realiza de manera exitosa la eliminación del conejo dando un mensaje de advertencia
RESULTADO OBTENIDO	El sistema muestra un mensaje de advertencia con cambio irreversible y el conejo es eliminado
ESTADO	No Ejecutado

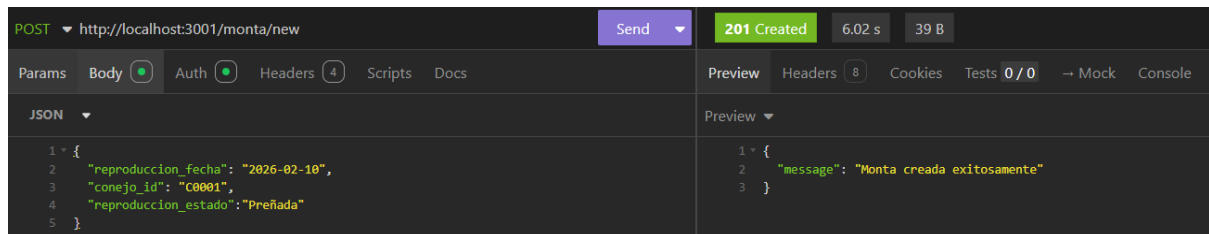


3.30. CP08.1-1 Registrar datos de la monta

IDENTIFICADOR	CP08.1-1
DESCRIPCIÓN	Se i <div><div><div>Request Order</div><div>Advanced</div><div>Tests 14 / 20</div><div>History</div><div>Console</div></div><div><div>Unselect All</div><div>All</div><div>Passed</div><div>Failed</div><div>Skipped</div></div><div><div>ITERATION 1</div><div>Iniciar Sesión - [url] [/usuario/login]</div><div>Crear Vacunación Conejo - [url] [/vacunacion/new/conejo]</div><div>Actualizar Vacunación - [url] [/vacunacion/V000001]</div><div>Crear Vacunación por jaula - [url] [/vacunacion/new/conejo]</div><div>Check If Status is 200 error: AssertionError: expected 400 to deeply equal 201 ACTUAL: After-response Test (0.1 ms)</div><div>Obtener Vacunación por conejo - [url] [/vacunacion/conejo/C0001]</div><div>Obtener Vacunación por id - [url] [/vacunacion/V000001]</div><div>Get Desparasitación por id - [url] [/desparasitacion/OES000001]</div><div>Get Desparasitación por id de conejo - [url] [/desparasitacion/conejo/C0001]</div><div>Crear Desparasitación - [url] [/desparasitacion/new]</div><div>Check If Status is 200 error: AssertionError: expected 200 to deeply equal 201 ACTUAL: After-response Test (0.1 ms)</div><div>Crear Desparasitación por jaula - [url] [/desparasitacion/new_cage]</div><div>Check If Status is 200 error: AssertionError: expected 200 to deeply equal 201 ACTUAL: After-response Test (0.2 ms)</div><div>Crear monta - [url] [/monta/new]</div><div>Check If Status is 201 error: AssertionError: expected 201 to deeply equal 200 ACTUAL: After-response Test (0.5 ms)</div></div></div>

	ngresan 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

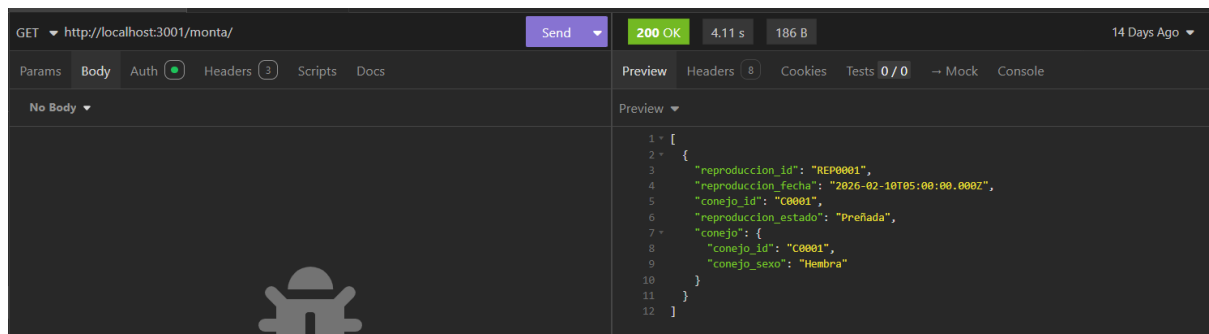


3.31. CP08.1-2 Obtener datos de la monta

IDENTIFICADOR	CP08.1-2
DESCRIPCIÓN	Se obtienen los dato s 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

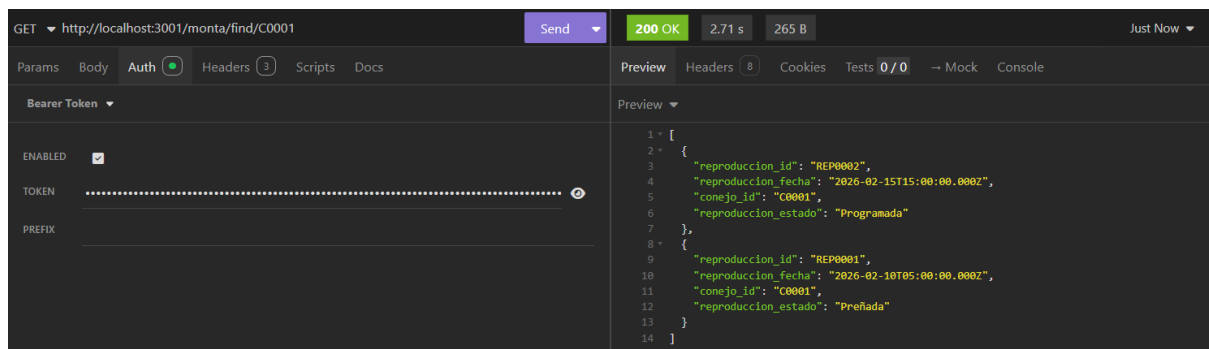


3.32. CP08.1-3 Obtener datos de la monta por conejo

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

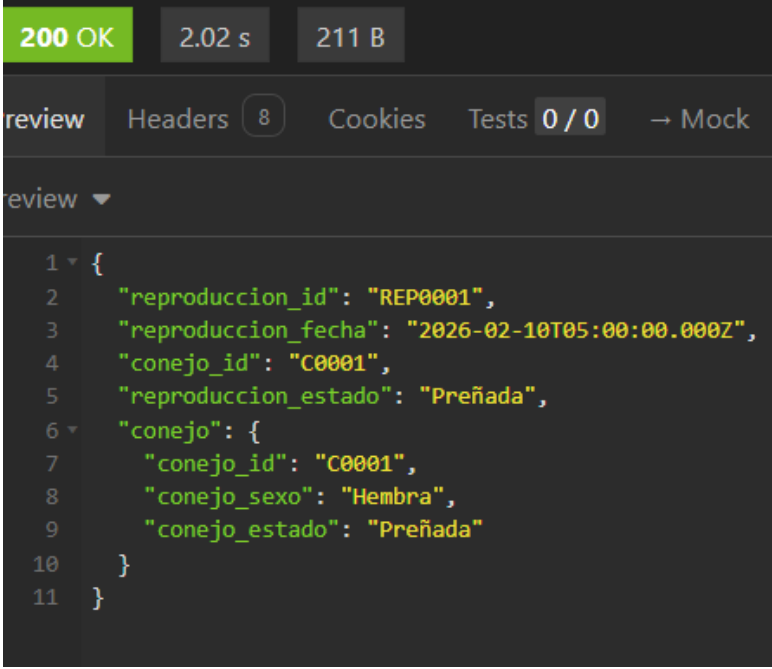
RESULTADO OBTENIDO	<pre>[{ "reproduccion_id": "REP0001", "reproduccion_fecha": "2026-02-10T05:00:00.000Z", "conejo_id": "C0001", "reproduccion_estado": "Preñada", "conejo": { "conejo_id": "C0001", "conejo_sexo": "Hembra" } }]</pre>
ESTADO	Aprobado

Evidencia

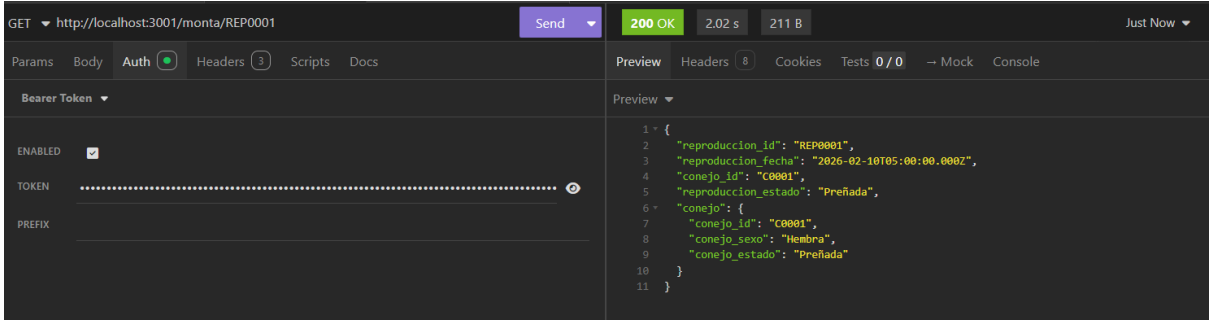


3.33. CP08.1-4 Registrar datos de la monta

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	 <pre> 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

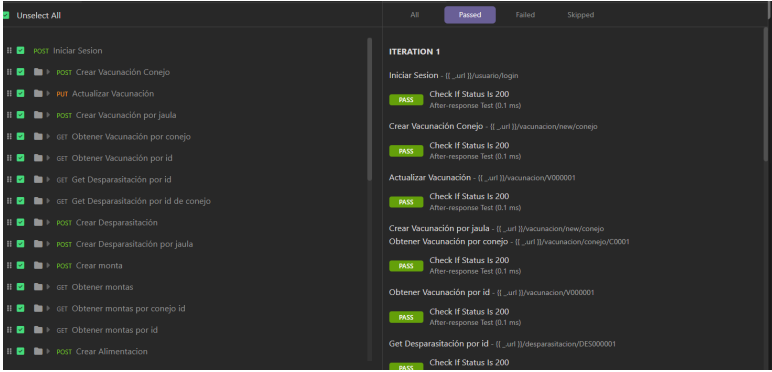
Evidencia



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

<div> <div>Unselect All</div> <div> <div>POST Iniciar Sesión</div> <div>POST Crear Vacunación Conejo</div> <div>PUT Actualizar Vacunación</div> <div>POST Crear Vacunación por jaula</div> <div>GET Obtener Vacunación por conejo</div> <div>GET Obtener Vacunación por id</div> <div>GET Get Desparasitación por id</div> <div>GET Get Desparasitación por id de conejo</div> <div>POST Crear Desparasitación</div> <div>POST Crear Desparasitación por jaula</div> <div>POST Crear monta</div> <div>GET Obtener montas</div> <div>GET Obtener montas por conejo id</div> <div>GET Obtener montas por id</div> <div>POST Crear Alimentación</div> </div> </div>	<div> <div>All</div> <div>Passed</div> <div>Failed</div> <div>Skipped</div> </div> <div> <div>ITERATION 1</div> <div>Iniciar Sesión - ({{_url}}/usuario/login)</div> <div>PASS Check If Status Is 200</div> <div>After-response Test (0.1 ms)</div> <div>Crear Vacunación Conejo - ({{_url}}/vacunacion/new/conejo)</div> <div>PASS Check If Status Is 200</div> <div>After-response Test (0.1 ms)</div> <div>Actualizar Vacunación - ({{_url}}/vacunacion/V000001)</div> <div>PASS Check If Status Is 200</div> <div>After-response Test (0.1 ms)</div> <div>Crear Vacunación por jaula - ({{_url}}/vacunacion/new/conejo)</div> <div>Obtener Vacunación por conejo - ({{_url}}/vacunacion/conejo/C0001)</div> <div>PASS Check If Status Is 200</div> <div>After-response Test (0.1 ms)</div> <div>Obtener Vacunación por id - ({{_url}}/vacunacion/V000001)</div> <div>PASS Check If Status Is 200</div> <div>After-response Test (0.1 ms)</div> <div>Get Desparasitación por id - ({{_url}}/desparasitacion/DES000001)</div> <div>PASS Check If Status Is 200</div> </div>
--	---

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

Evidencia