**TEST CASES PETS STORE**

|  |  |
| --- | --- |
| Identificador | **TC-001** |
| Escenario | Creación de lista de usuarios |
| Descripción | Creación de usuarios mediante la utilización del recurso  POST /user/createWithList |
| Precondiciones | * el API PetShop debe estar desplegado y accesible desde internet * tener instalado un cliente REST como Postman |
| Pasos | Consumir el recurso POST/ user/createWithList enviando información correcta en el body de la petición |
| Resultado Esperado | 200 ok |
| Probador | Daniel |
| Estado | Exitoso |

|  |  |
| --- | --- |
| Identificador | **TC-002** |
| Escenario | Creación de lista de usuarios |
| Descripción | Creación de usuarios mediante la utilización del recurso  POST /user/createWithList |
| Precondiciones | * el API PetShop debe estar desplegado y accesible desde internet * tener instalado un cliente REST como Postman |
| Pasos | Consumir el recurso POST/ user/createWithList enviando información errada en el id del body de la petición |
| Resultado Esperado | 500 Server Error |
| Probador | Daniel |
| Estado | Fallido |

|  |  |
| --- | --- |
| Identificador | **TC-003** |
| Escenario | Creación de lista de usuarios con matriz |
| Descripción | Creación de usuarios mediante la utilización del recurso  POST /user/createWithArray |
| Precondiciones | * el API PetShop debe estar desplegado y accesible desde internet * tener instalado un cliente REST como Postman |
| Pasos | Consumir el recurso POST /user/createWithArray enviando información valida en el body de la petición |
| Resultado Esperado | 200 ok |
| Probador | Daniel |
| Estado | Exitoso |

|  |  |
| --- | --- |
| Identificador | **TC-004** |
| Escenario | Creación de lista de usuarios con matriz |
| Descripción | Creación de usuarios mediante la utilización del recurso  POST /user/createWithArray |
| Precondiciones | * el API PetShop debe estar desplegado y accesible desde internet * tener instalado un cliente REST como Postman |
| Pasos | Consumir el recurso POST /user/createWithArray enviando información vacia en el campo ID del body de la petición |
| Resultado Esperado | 500 Server Error |
| Probador | Daniel |
| Estado | Fallido |

|  |  |
| --- | --- |
| Identificador | **TC-005** |
| Escenario | Creación de lista de usuarios con matriz |
| Descripción | Creación de usuarios mediante la utilización del recurso  POST /user |
| Precondiciones | * el API PetShop debe estar desplegado y accesible desde internet * tener instalado un cliente REST como Postman |
| Pasos | Consumir el recurso POST /user enviando información valida en el body de la petición |
| Resultado Esperado | 200 ok |
| Probador | Daniel |
| Estado | Exitoso |

|  |  |
| --- | --- |
| Identificador | **TC-006** |
| Escenario | Creación de lista de usuarios con matriz |
| Descripción | Creación de usuarios mediante la utilización del recurso  POST /user/createWithArray |
| Precondiciones | * El API PetShop debe estar desplegado y accesible desde internet * Tener instalado un cliente REST como Postman |
| Pasos | Consumir el recurso POST /user/createWithArray enviando información vacia en el campo ID del body de la petición |
| Resultado Esperado | 400 Bad Request |
| Probador | Daniel |
| Estado | Fallido |

|  |  |
| --- | --- |
| Identificador | **TC-007** |
| Escenario | Registra al usuario en el sistema |
| Descripción | Creación de usuarios mediante la utilización del recurso  GET /user/login?username=IDTEST&password=pass |
| Precondiciones | * El API PetShop debe estar desplegado y accesible desde internet * Tener instalado un cliente REST como Postman |
| Pasos | Consumir el recurso GET /user/login?username=IDTEST&password=pass enviando información correcta con el usuario y el password creados en los campos de Params |
| Resultado Esperado | 200 ok |
| Probador | Daniel |
| Estado | Exitoso |

|  |  |
| --- | --- |
| Identificador | **TC-008** |
| Escenario | Registra al usuario en el sistema |
| Descripción | Creación de usuarios mediante la utilización del recurso  GET /user/login?username=IDTEST&password=pass |
| Precondiciones | * El API PetShop debe estar desplegado y accesible desde internet * Tener instalado un cliente REST como Postman |
| Pasos | Consumir el recurso GET /user/createWithArray enviando información vacia en el campo ID del body de la petición |
| Resultado Esperado | 400 Bad Request |
| Probador | Daniel |
| Estado | Fallido |
| Identificador | **TC-009** |
| Escenario | Cierra la sesión actual del usuario conectado |
| Descripción | Cerrar Sesión mediante la utilización del recurso  GET /user/logout |
| Precondiciones | * El API PetShop debe estar desplegado y accesible desde internet * Tener instalado un cliente REST como Postman |
| Pasos | Consumir el recurso GET /user/logout sin parametros |
| Resultado Esperado | 200 ok |
| Probador | Daniel |
| Estado | Exitoso |

|  |  |
| --- | --- |
| Identificador | **TC-010** |
| Escenario | Obtener usuario por nombre de usuario |
| Descripción | Buscar usuarios por nombre de usuario mediante la utilización del recurso  GET /user/:username |
| Precondiciones | * El API PetShop debe estar desplegado y accesible desde internet * Tener instalado un cliente REST como Postman |
| Pasos | Consumir el recurso GET /user/:username con nombre de usuario creado y valido en Params |
| Resultado Esperado | 200 ok |
| Probador | Daniel |
| Estado | Exitoso |

|  |  |
| --- | --- |
| Identificador | **TC-011** |
| Escenario | Obtener usuario por nombre de usuario |
| Descripción | Buscar usuarios por nombre de usuario mediante la utilización del recurso  GET /user/:username |
| Precondiciones | * El API PetShop debe estar desplegado y accesible desde internet * Tener instalado un cliente REST como Postman |
| Pasos | Consumir el recurso GET /user/:username con nombre de usuario no creado e invalido en Params |
| Resultado Esperado | 404 Not Found |
| Probador | Daniel |
| Estado | Fallido |

|  |  |
| --- | --- |
| Identificador | **TC-012** |
| Escenario | Usuario actualizado |
| Descripción | Buscar usuarios por nombre de usuario mediante la utilización del recurso  PUT /user/:username |
| Precondiciones | * El API PetShop debe estar desplegado y accesible desde internet * Tener instalado un cliente REST como Postman |
| Pasos | Consumir el recurso PUT /user/:username con información valida a actualizar |
| Resultado Esperado | 200 ok |
| Probador | Daniel |
| Estado | Positiva |

|  |  |
| --- | --- |
| Identificador | **TC-013** |
| Escenario | Usuario actualizado |
| Descripción | Actualizar usuario por nombre de usuario mediante la utilización del recurso  PUT /user/:username |
| Precondiciones | * El API PetShop debe estar desplegado y accesible desde internet * Tener instalado un cliente REST como Postman |
| Pasos | Consumir el recurso PUT /user/:username con usuario invalido a actualizar |
| Resultado Esperado | 404 Not Found |
| Probador | Daniel |
| Estado | Fallida |

|  |  |
| --- | --- |
| Identificador | **TC-014** |
| Escenario | Borrar Usuario |
| Descripción | Borrar usuarios por nombre de usuario mediante la utilización del recurso  DELETE /user/:username |
| Precondiciones | * El API PetShop debe estar desplegado y accesible desde internet * Tener instalado un cliente REST como Postman |
| Pasos | Consumir el recurso DELETE /user/:username con usuario valido a borrar |
| Resultado Esperado | 200 ok |
| Probador | Daniel |
| Estado | Exitosa |

|  |  |
| --- | --- |
| Identificador | **TC-015** |
| Escenario | Borrar Usuario |
| Descripción | Borrar usuarios por nombre de usuario mediante la utilización del recurso  DELETE /user/:username |
| Precondiciones | * El API PetShop debe estar desplegado y accesible desde internet * Tener instalado un cliente REST como Postman |
| Pasos | Consumir el recurso DELETE /user/:username con usuario invalido a borrar |
| Resultado Esperado | 404 Not Found |
| Probador | Daniel |
| Estado | Fallida |

|  |  |
| --- | --- |
| Identificador | **TC-016** |
| Escenario | Agregar una nueva mascota a la tienda |
| Descripción | Agregar nueva mascota mediante la utilización del recurso  POST /pet |
| Precondiciones | * El API PetShop debe estar desplegado y accesible desde internet * Tener instalado un cliente REST como Postman |
| Pasos | Consumir el recurso POST /pet con información valida en el body |
| Resultado Esperado | 200 ok |
| Probador | Daniel |
| Estado | Exitosa |

|  |  |
| --- | --- |
| Identificador | **TC-017** |
| Escenario | Agregar una nueva mascota a la tienda |
| Descripción | Agregar nueva mascota mediante la utilización del recurso  POST /pet |
| Precondiciones | * El API PetShop debe estar desplegado y accesible desde internet * Tener instalado un cliente REST como Postman |
| Pasos | Consumir el recurso POST /pet con id errado |
| Resultado Esperado | 400 Bad content |
| Probador | Daniel |
| Estado | Fallida |

|  |  |
| --- | --- |
| Identificador | **TC-018** |
| Escenario | Actualizar una mascota existente |
| Descripción | Actualizar una mascota existente mediante la utilización del recurso  PUT /pet |
| Precondiciones | * El API PetShop debe estar desplegado y accesible desde internet * Tener instalado un cliente REST como Postman |
| Pasos | Consumir el recurso PUT /pet con datos validos en el body |
| Resultado Esperado | 200 ok |
| Probador | Daniel |
| Estado | Exitosa |

|  |  |
| --- | --- |
| Identificador | **TC- 019** |
| Escenario | Actualizar una mascota existente |
| Descripción | Actualizar una mascota existente mediante la utilización del recurso  PUT /pet |
| Precondiciones | * El API PetShop debe estar desplegado y accesible desde internet * Tener instalado un cliente REST como Postman |
| Pasos | Consumir el recurso PUT /pet con datos no validos en el body |
| Resultado Esperado | 400 Bad Request |
| Probador | Daniel |
| Estado | Fallido |

|  |  |
| --- | --- |
| Identificador | **TC- 020** |
| Escenario | Encontrar mascotas por estado |
| Descripción | Encontrar mascotas por estado existente mediante la utilización del recurso  GET /pet/findByStatus?status=available&status=available |
| Precondiciones | * El API PetShop debe estar desplegado y accesible desde internet * Tener instalado un cliente REST como Postman |
| Pasos | Consumir el recurso GET /pet/findByStatus?status=available&status=available con datos validos en el params |
| Resultado Esperado | 200 ok |
| Probador | Daniel |
| Estado | Exitosa |

|  |  |
| --- | --- |
| Identificador | **TC-021** |
| Escenario | Encontrar mascotas por estado |
| Descripción | Encontrar mascotas por estado existente mediante la utilización del recurso  GET /pet/findByStatus?status=available&status=available |
| Precondiciones | * El API PetShop debe estar desplegado y accesible desde internet * Tener instalado un cliente REST como Postman |
| Pasos | Consumir el recurso GET /pet/findByStatus?status=available&status=available con datos no validos en el params |
| Resultado Esperado | 400 Bad Request |
| Probador | Daniel |
| Estado | Fallido |

|  |  |
| --- | --- |
| Identificador | **TC-022** |
| Escenario | Encontrar mascotas por etiquetas |
| Descripción | Encontrar mascotas por etiquetas existente mediante la utilización del recurso  GET /pet/findByTags?tags=PetTest |
| Precondiciones | * El API PetShop debe estar desplegado y accesible desde internet * Tener instalado un cliente REST como Postman |
| Pasos | Consumir el recurso GET /pet/findByStatus?status=available&status=available con datos validos en el params |
| Resultado Esperado | 200 ok |
| Probador | Daniel |
| Estado | Exitosa |

|  |  |
| --- | --- |
| Identificador | **TC-023** |
| Escenario | Encontrar mascotas por etiquetas |
| Descripción | Encontrar mascotas por etiquetas existente mediante la utilización del recurso  GET /pet/:petId |
| Precondiciones | * El API PetShop debe estar desplegado y accesible desde internet * Tener instalado un cliente REST como Postman |
| Pasos | Consumir el recurso GET /pet/:petId con tags no validos en el params |
| Resultado Esperado | 404 Not Found |
| Probador | Daniel |
| Estado | Fallido |

|  |  |
| --- | --- |
| Identificador | **TC-024** |
| Escenario | Encontrar mascotas por ID |
| Descripción | Encontrar mascotas por ID existente mediante la utilización del recurso  GET /pet/:petId |
| Precondiciones | * El API PetShop debe estar desplegado y accesible desde internet * Tener instalado un cliente REST como Postman |
| Pasos | Consumir el recurso GET /pet/:petId con ID valido en el params |
| Resultado Esperado | 200 ok |
| Probador | Daniel |
| Estado | Exitoso |

|  |  |
| --- | --- |
| Identificador | **TC-025** |
| Escenario | Encontrar mascotas por ID |
| Descripción | Encontrar mascotas por ID existente mediante la utilización del recurso  GET /pet/:petId |
| Precondiciones | * El API PetShop debe estar desplegado y accesible desde internet * Tener instalado un cliente REST como Postman |
| Pasos | Consumir el recurso GET /pet/:petId con ID no valido en el params |
| Resultado Esperado | 404 Not Found |
| Probador | Daniel |
| Estado | Fallido |

|  |  |
| --- | --- |
| Identificador | **TC-026** |
| Escenario | Encontrar mascotas por ID |
| Descripción | Encontrar mascotas por ID existente mediante la utilización del recurso  GET /pet/:petId |
| Precondiciones | * El API PetShop debe estar desplegado y accesible desde internet * Tener instalado un cliente REST como Postman |
| Pasos | Consumir el recurso GET /pet/:petId con ID no valido en el params |
| Resultado Esperado | 404 Not Found |
| Probador | Daniel |
| Estado | Fallido |