Universidad de Sevilla

Escuela Técnica Superior de Ingeniería Informática

**GRUPO DP2-1920-G2-13**



Grado en Ingeniería Informática – Ingeniería del Software  
Diseño y Pruebas - 2

Curso 2019 – 2020

:

|  |  |
| --- | --- |
| **Grupo de prácticas** | **G2-13** |
| **Autores** | **Rol** |
| Fernández Rodríguez, Andrés | Desarrollador |
| Santos Tirado, Carlos | Desarrollador |
| Jiménez Hartman, Mario David | Desarrollador |
| Guerrero Luna, Gonzalo | Desarrollador |

Indice

[**1.** **Descripción general** 3](#_Toc36479324)

[**2.** **Descripción de las entidades y relaciones de la aplicación** 3](#_Toc36479325)

[**3.** **Historias de usuario de la aplicación y casos positivos y negativos** 3](#_Toc36479326)

[**4.** **Planificación de los diferentes sprint** 10](#_Toc36479327)

1. **Descripción general**

La propuesta del grupo para la realización del proyecto es de desarrollar una extensión para la aplicación base Pet Clinic. Dicha extensión consistirá en añadir dos módulos a gestionar dentro de la misma:

* Por un lado, la denominada “BeautyPet”, relacionado con la estética animal y enfocada a la gestión de citas y servicios por parte de trabajadores y de clientes.
* En segundo lugar, un apartado dedicado a la publicación de animales abandonados y/o encontrados en la calle para que estos puedan ser evaluados por los veterinarios y posteriormente publicados para que puedan ser adoptados.

1. **Descripción de las entidades y relaciones de la aplicación**

Se plantean un total de cinco entidades nuevas a añadir a la aplicación:

* Esteticista animal: como rol nuevo dentro de la aplicación aparte del propio veterinario y los propietarios. Tiene como atributos una colección de tipos de mascotas denominada especializaciones, un nombre, un apellido y tiene relación con usuario para poder acceder al sistema y con una colección de servicios estéticos asociado al esteticista.
* Servicio estético: asociados a grupos de animales predeterminados y que pueden ser de varios tipos. Tiene como atributos un nombre, una descripción del servicio y se relaciona con tipo de mascota a la que está destinado y con el esteticista que lo creó.
* Productos estéticos: ítems especiales y específicos para algunos animales que pueden ser solicitados por los clientes para sus mascotas. Posee como atributos un nombre, una descripción del producto, una propiedad que indica si está disponible o no y se relaciona con el esteticista que lo creó.
* Cita para servicio estético: solicitud de un cliente a un esteticista para la realización de un servicio. Sus atributos son una descripción y una fecha de inicio y se relaciona con la mascota para la que es solicitada la cita y con el servicio estético que se solicita.
* Petición de recogida: solicitud de un usuario para que un animal encontrado en la calle sea evaluado por un veterinario y puesto en anuncio para ser adoptado.

1. **Historias de usuario de la aplicación y casos positivos y negativos**

A partir de dichas entidades surgen las siguientes historias de usuario:

1. **Como** esteticista animal contratado por la clínica **quiero** poder registrarme en el sistema como dicho rol y editar mis atributos, añadiendo campos adicionales tales como la especialización de animales **para** poder usar el sistema y trabajar con él.

Casos positivos:

* Accedo al formulario para hacer el registro de beautician.
* Prueba unitaria del método initCreationFormBeautician de UserController en UserBeauticianControllerTest mediante el método testInitCreationForm.
* Prueba E2E del método initCreationFormBeautician de UserController en UserBeauticianControllerE2ETest mediante el método testInitCreationForm.
* Relleno el formulario con todos los campos completos y correctos.
* Prueba unitaria del método processCreationFom de UserController en UserBeauticianControllerTest en el método testProcessCreationFormSuccess.
* Prueba de servicios e integración de bd del método saveBeautician de BeauticianService en el método testSavingBeauticianCorrect de BeauticianServiceTests.
* Prueba E2E del método processCreationForm de UserController en UserBeauticianControllerE2ETests n el método testProcessCreationFormSuccess.
* Me logueo como Beautician y muestro mis credenciales en pantalla.
* Prueba unitaria del método showBeautician de BeauticianController en BeauticianControllerTest en el método testShowBeautician.
* Prueba E2E del método showBeautician de BeauticianController en BeauticianControllerE2ETest en el método testShowBeautician.
* Obtengo mis datos iniciales antes de actualizarlos.
* Prueba unitaria del método initUpdateBeauticianForm de BeauticianController en BeauticianControllerTests en el método testInitUpdateBeauticianForm.
* Prueba E2E del método initUpdateBeauticianForm de BeauticianController en BeauticianControllerE2ETests en el método testInitUpdateBeauticianForm.
* Actualizo los datos de usuario de mi cuenta.
* Prueba unitaria del método processUpdateBeauticianForm de BeauticianController en BeauticianControllerTests en el método testUpdateBeauticianFormSucess.
* Prueba de servicios e integración de bd del método saveBeautician de BeauticianService en el método testSavingBeauticianCorrect de BeautcianServiceTests.
* Prueba E2E del método processUpdateBeauticianForm de BeauticianController en el método testProcessUpdateBeauticianFormSuccess de BeauticianControllerE2ETests.

Casos negativos:

* Relleno el formulario dejando el nombre en blanco y salta un error.
* Prueba unitaria del método processCreationForm de UserController en UserBeauticianControllerTest en el método testProcessCreationFormHasErrorsFirstName.
* Prueba de servicios e integración de bd del método saveBeautician de BeauticianService en el método testSavingBeauticianWithIncorrectAttribute de BeauticianServiceTests.
* Prueba E2E del método processCreationForm de UserController en UserBeauticianControllerE2ETests en el método testProcessCreationFormHasErrorsFirstName.
* Relleno el formulario seleccionando una especialización incorrecta y salta un error.
* Prueba unitaria del método processCreationForm de UserController en UserBeauticianControllerTest en el método testProcessCreationFormHasErrorsSpecializations.
* Prueba de servicios e integración de bd del método saveBeautician de BeauticianService en el método testSavingBeauticianWithIncorrectSpecialization de BeauticianServiceTests.
* Prueba E2E del método processCreationForm de UserController en UserBeauticianControllerE2ETests en el método testProcessCreationFormHasErrorsSpecializations.
* Intento mostrar las credenciales de un usuario que no es mío.
* Prueba unitaria del método showBeautician en BeauticianController en el método showBeauticianIncorrectLogIn en BeauticianControllerTests.
* Prueba E2E del método showBeautician en BeauticianController en el método showBeauticianIncorrectLogIn en BeauticianControllerE2ETests.
* Actualizo el formulario dejando el nombre en blanco y salta un error.
* Prueba unitaria del método processUpdateBeauticianForm de BeauticianController en BeauticianControllerTest en el método testProcessUpdateFormHasErrorsFirstName.
* Prueba de servicios e integración de bd del método saveBeautician de BeauticianService en el método testSavingBeauticianWithIncorrectAttribute de BeauticianServiceTests.
* Prueba E2E del método processUpdateBeauticianForm de BeauticianController en BeauticianControllerE2ETests en el método testProcessUpdateFormHasErrorsFirstName.
* Actualizo el formulario seleccionando una especialización incorrecta y salta un error.
* Prueba unitaria del método processUpdateBeauticianForm de BeauticianController en BeauticianControllerTest en el método testProcessUpdateFormHasErrorsSpecializations.
* Prueba de servicios e integración de bd del método saveBeautician de BeauticianService en el método testSavingBeauticianWithIncorrectSpecialization de BeauticianServiceTests.
* Prueba E2E del método processUpdateBeauticianForm de BeauticianController en BeauticianControllerE2ETests en el método testProcessUpdateFormHasErrorsSpecializations.

1. **Como** esteticista animal **quiero** poder registrar los servicios que oferto, dichos servicios deben incluir un nombre, una descripción y tipo de animal que pueden recibirlo. Todos los campos son obligatorios a excepción de la descripción que es opcional.

Casos positivos:

* Intento acceder al formulario para crear un servicio estético con un id asociado a mi usuario y me deja entrar.
  + - Prueba unitaria del método de BeautyCenterController initCreationFormBeautyCenter en BeautyCenterControllerTests mediante el método testInitCreationForm.
    - Prueba E2E del método de BeautyCenterController initCreationFormBeautyCenter en BeautyCenterControllerE2ETests mediante el método testInitCreationForm.
* Relleno el formulario con todos los campos correctos.
  + - Prueba unitaria del método de BeautyCenterController processCreationFormBeautyCenter en BeautyCenterControllerTests mediante el método testProcessCreationFormSuccess.
    - Prueba E2E del método de BeautyCenterController processCreationFormBeautyCenter en BeautyCenterControllerE2ETests mediante el método testProcessCreationFormSuccess.
    - Prueba de servicio e integración con la bd del método save de la clase BeautyCenterService en la clase BeautyCenterServiceTests en el método testSaveBeautyCenterCorrect.
    - Prueba UI de la HU en la clase RegisterBeautyCenterSuccessUITest.
* Relleno el formulario con todos los campos correctos pero dejando la descripción vacía.
  + - Prueba unitaria del método de BeautyCenterController processCreationFormBeautyCenter en BeautyCenterControllerTests mediante el método testProcessCreationNoDescriptionFormSuccess.
    - Prueba E2E del método de BeautyCenterController processCreationFormBeautyCenter en BeautyCenterControllerE2ETests mediante el método testProcessCreationNoDescriptionFormSuccess.
    - Prueba de servicio e integración con la bd del método save de la clase BeautyCenterService en la clase BeautyCenterServiceTests en el método testSaveBeautyCenterNoDescriptionCorrect.

Casos negativos:

* Intento acceder al formulario usando un id de esteticista no asociado a mi usuario. La página devuelve error.
  + - Prueba unitaria del método de BeautyCenterController testInitCreationFormError en BeautyCenterControllerTests mediante el método testInitCreationForm.
    - Prueba E2E del método de BeautyCenterController initCreationFormBeautyCenter en BeautyCenterControllerE2ETests mediante el método testInitCreationFormError.
* Relleno el formulario con todos los campos correctos pero no proporciono un tipo de mascota devuelve a la página de creación mostrándome el error.
  + - Prueba unitaria del método de BeautyCenterController processCreationFormBeautyCenter en BeautyCenterControllerTests mediante el método testProcessCreationNoPetTypeHasErrors.
    - Prueba E2E del método de BeautyCenterController processCreationFormBeautyCenter en BeautyCenterControllerE2ETests mediante el método testProcessCreationNoPetTypeHasErrors.
    - Prueba de servicio e integración con la bd del método save de la clase BeautyCenterService en la clase BeautyCenterServiceTests en el método testSaveBeautyCenterNullPetType.
    - Prueba UI de la HU en la clase RegisterBeautyCenterErrorUITest.
* Relleno el formulario con todos los campos correctos pero el nombre es inferior a 3 caracteres. El sistema me devuelve a la página de creación mostrándome el error.
  + - Prueba unitaria del método de BeautyCenterController processCreationFormBeautyCenter en BeautyCenterControllerTests mediante el método testProcessCreationShortNameHasErrors.
    - Prueba E2E del método de BeautyCenterController processCreationFormBeautyCenter en BeautyCenterControllerE2ETests mediante el método testProcessCreationShortNameHasErrors.
    - Prueba de servicio e integración con la bd del método save de la clase BeautyCenterService en la clase BeautyCenterServiceTests en el método testSaveBeautyCenterShortName.
    - Prueba de servicio e integración con la bd del método save de la clase BeautyCenterService en la clase BeautyCenterTests en el método testSaveBeautyCenterNullName.

1. **Como** esteticista animal **quiero** poder modificar todos los campos de un servicio **para** poder permitir cambios sustanciales.

Casos positivos:

* Muestro los datos iniciales del servicio por pantalla.
* Prueba unitaria del método initUpdateFormBeautyCenter de la clase BeautyCenterController en el método testInitUpdateForm de la clase BeautyCenterControllerTests.
* Prueba E2E del método initUpdateFormBeautyCenter de la clase BeautyCenterController en el método testInitUpdateBeautyCenterForm.
* Modifico correctamente los atributos del servicio.
* Prueba unitaria del método processUpdateBeautyCenterForm de la clase BeautyCenterController en el método testProcessUpdateBeautyCenterSuccess de la clase BeautyCenterControllerTests.
* Prueba de servicios e integración de la bd del método update de la clase BeautyCenterService en el método testUpdateBeautyCenterSuccess en la clase BeautyCenterServiceTests.
* Prueba E2E del método processUpdateBeautyCenterForm de la clase BeautyCenterController en el método testProcessUpdateBeautyCenterSuccess de la clase BeautyCenterControllerE2ETests.
* Modifico correctamente los atributos del servicio dejando el parámetro descripción en blanco.
* Prueba unitaria del método processUpdateBeautyCenterForm de la clase BeautyCenterController en el método testProcessUpdateBeautyCenterSuccessNoDescription de la clase BeautyCenterControllerTests.
* Prueba de servicios e integración de la bd del método update de la clase BeautyCenterService en el método testUpdateBeautyCenterSuccessNoDescription en la clase BeautyCenterServiceTests.
* Prueba E2E del método processUpdateBeautyCenterForm de la clase BeautyCenterController en el método testProcessUpdateBeautyCenterSuccessNoDescription de la clase BeautyCenterControllerE2ETests.
* Modifico los datos seleccionando un tipo de mascota erróneo y salta error.
* Prueba unitaria del método processUpdateBeautyCenterForm de la clase BeautyCenterController en el método testProcessUpdateBeautyCenterFormHasErrorsPetType de la clase BeautyCenterControllerTests.
* Prueba de servicios e integración de la bd del método update de la clase BeautyCenterService en el método testUpdateBeautyCenterErrorPetType en la clase BeautyCenterServiceTests.
* Prueba E2E del método processUpdateBeautyCenterForm de la clase BeautyCenterController en el método testProcessUpdateBeautyCenterFormHasErrorsPetType de la clase BeautyCenterControllerE2ETests.
* Modifico los datos dejando en blanco un campo obligatorio, por ejemplo name.
* Prueba unitaria del método processUpdateBeautyCenterForm de la clase BeautyCenterController en el método testProcessUpdateBeautyCenterFormHasErrorsEmptyAttribute de la clase BeautyCenterControllerTests.
* Prueba de servicios e integración de la bd del método update de la clase BeautyCenterService en el método testUpdateBeautyCenterErrorEmptyAttribute en la clase BeautyCenterServiceTests.
* Prueba E2E del método processUpdateBeautyCenterForm de la clase BeautyCenterController en el método testProcessUpdateBeautyCenterFormHasErrorsEmptyAttribute de la clase BeautyCenterControllerE2ETests.

1. **Como** esteticista animal **quiero** poder eliminar los servicios que oferto **para** deshacerme de aquellos que ya no imparto.

Caso positivo:

* Un esteticista animal pulsa eliminar sobre un servicio suyo y este se elimina satisfactoriamente.
  + - Prueba unitaria del método de BeautyCenterController deleteBeautyCenter() en BeautyCenterControllerTests en el método testProcessDeleteBeautyCenterFormSuccess.
    - Prueba E2E del método de BeautyCenterController deleteBeautyCenter() en BeautyCenterControllerE2ETests en el método testProcessDeleteBeautyCenterFormSuccess.

Caso negativo:

* Un esteticista animal accede a eliminar un servicio que no es suyo mediante la url y la aplicación le lanza un mensaje de error.
  + - Prueba unitaria del método de BeautyCenterController deleteBeautyCenter() en BeautyCenterControllerTests en el método testProcessDeleteBeautyCenterFormWithErrors.
    - Prueba E2E del método de BeautyCenterController deleteBeautyCenter() en BeautyCenterControllerE2ETests en el método testProcessDeleteBeautyCenterFormWithErrors.

1. **Como** esteticista animal **quiero** mostrar y registrar productos estéticos, almacenando información como nombre, animal o animales asociados, descripción y disponibilidad **para** que los clientes soliciten dichos ítems en un servicio.

Casos positivos:

* Con una sesión iniciada con rol Beautician, accedo al formulario de crear producto e introduzco los datos de manera correcta.
  + - * + Prueba unitaria del método initCreationForm(…) de ProductController en el método testProductInitCreationForm() de ProductControllerTests.
        + Prueba unitaria del método processCreationForm(…) de ProductController en el método testProcessCreationFormSuccess() de ProductControllerTests.
        + Prueba E2E del método initCreationForm(…) de ProductController en el método testProductInitCreationForm() de ProductControllerE2ETests.
        + Prueba E2E del método processCreationForm(…) de ProductController en el método testProcessCreationFormSuccess() de ProductController E2ETests.
* -Prueba de interfaz de usuario de los métodos initCreationForm(…) y processCreationForm(…) de ProductController en CreateAndShowProductSuccessUITest.
* Con una sesión iniciada con rol Beautician, muestro la lista de productos que he creado.
  + - * + Prueba unitaria del método showProductList(…) de ProductController en el método testShowProduct() de ProductControllerTests.
        + Prueba E2E del método showProductList(…) de ProductController en el método testShowProduct() de ProductControllerE2ETests.
        + Prueba de interfaz de usuario del método showProductList (…) de ProductController en CreateAndShowProductSuccessUITest.

Casos negativos:

* Con una sesión iniciada con rol Beautician, cambio la url de acceso al formulario de crear producto introduciendo un id de Beautician diferente al mío, y se produce una redirección a una página de excepción.
  + - * + Prueba unitaria del método initCreationForm(…) de ProductController en el método testProductInitCreationFormError() de ProductControllerTests.
        + Prueba E2E del método initCreationForm(…) de ProductController en el método testProductInitCreationFormError() de ProductControllerE2ETests.
* Con una sesión iniciada con rol Beautician, introduzco todos los datos excepto la descripción (obligatoria), y se devuelve de nuevo al formulario.
  + - * + Prueba unitaria del método processCreationForm(…) de ProductController en el método testProcessCreationEmptyInputsHasErrors() de ProductControllerTests.
        + Prueba E2E del método processCreationForm(…) de ProductController en el método testProcessCreationEmptyInputsHasErrors() de ProductControllerE2ETests.
        + Prueba de interfaz de usuario del método processCreationForm (…) de ProductController en CreateAndShowProductsNoDescriptionErrorUITest.
* Con una sesión iniciada con rol Beautician, introduzco en el formulario un tipo de mascota inexistente y, y se devuelve de nuevo el formulario.
  + - * + Prueba unitaria del método processCreationForm(…) de ProductController en el método testProcessCreationWrongInputHasErrors() de ProductControllerTests.
        + Prueba E2E del método processCreationForm(…) de ProductController en el método testProcessCreationWrongInputHasErrors() de ProductControllerE2ETests.

1. **Como** esteticista animal **quiero** modificar los campos de un producto estético **para** que quede constancia de futuros cambios por parte de los fabricantes.

Caso positivo:

* Accedo a editar un producto que registré yo como usuario mediante mi id. La página me devuelve correctamente el formulario para editar el producto.
  + - Prueba unitaria del método de ProductController initUpdateForm en ProductControllerTests mediante el método testInitUpdateForm.
    - Prueba E2E del método de ProductController initUpdateForm en ProductControllerE2ETests mediante el método testInitUpdateForm.
* Edito un producto con todos los campos correctos. La página me actualiza correctamente el producto.
  + - Prueba unitaria del método de ProductController processUpdateForm en ProductControllerTests mediante el método testProcessUpdateFormSuccess.
    - Prueba E2E del método de ProductController processUpdateForm en ProductControllerE2ETests mediante el método testProcessUpdateFormSuccess.
    - Prueba UI de la HU en la clase UpdateProductSuccessUITest.

Caso negativo:

* Accedo a editar un producto que no registré yo como usuario mediante otra id. La página me devuelve un error.
  + - Prueba unitaria del método de ProductController initUpdateForm en ProductControllerTests mediante el método testInitUpdateFormError.
    - Prueba E2E del método de ProductController initUpdateForm en ProductControllerE2ETests mediante el método testInitUpdateFormError.
* Edito un producto con todos los campos correctos pero no introduzco descripción. En concreto introduzco nombre = other name, tipo de mascota = gato, description = new description, y disponibilidad = true. La página me actualiza correctamente el producto.
  + - Prueba unitaria del método de ProductController processUpdateForm en ProductControllerTests mediante el método testProcessUpdateFormNoDescriptionError.
    - Prueba E2E del método de ProductController processUpdateForm en ProductControllerE2ETests mediante el método testProcessUpdateFormNoDescriptionError.
    - Prueba UI de la HU en la clase UpdateProductErrorUITest.

1. **Como** esteticista animal **quiero** eliminar productos estéticos **para** que no haya posibilidad de elegir productos descatalogados.

Caso positivos:

* Con una sesión iniciada con rol Beautician, elimino el producto seleccionado de mi lista de productos y se me redirige a la página principal.
  + - * + Prueba unitaria del método delete(…) de ProductController en el método testProcessDeleteProductFormSuccess() de ProductControllerTests.
        + Prueba E2E del método delete(…) de ProductController en el método testProcessDeleteProductFormSuccess() de ProductControllerE2ETests.
        + Prueba de interfaz de usuario del método delete(…) de ProductController en DeleteProductSuccessUITest.

Casos negativos:

* Con una sesión iniciada con rol Beautician, elimino un producto de otro Beautician cambiando el id del producto en la url y se me redirige a una página de error.
  + - * + Prueba unitaria del método delete(…) de ProductController en el método testProcessDeleteProductFormErrors() de ProductControllerTests.
        + Prueba E2E del método delete(…) de ProductController en el método testProcessDeleteProductFormErrors() de ProductControllerE2ETests.
        + Prueba de interfaz de usuario del método delete(…) de ProductController en DeleteProductNoPermissionErrorUITest.

1. **Como** esteticista **quiero** que cualquier cliente pueda solicitar una cita para un servicio en el horario de lunes a viernes de 16:00 a 20:00 y que no haya otra cita que pise dicho horario **para** poder atender a un único cliente en una hora determinada y que no soliciten citas fuera de mi horario de trabajo.

Caso positivo:

* Solicito una cita para una mascota en un horario disponible para el esteticista. La cita se registra correctamente.
  + - Prueba unitaria del método de BeautyDateController processCreationFormBeautyDate en BeautyDateControllerTests mediante el método testProcessCreateBeautyDateFormSuccessCorrectTime.
    - Prueba E2E del método de BeautyDateController processCreationFormBeautyDate en BeautyDateControllerE2ETests mediante el método testProcessCreateBeautyDateFormSuccessCorrectTime.
    - Prueba de servicio e integración con la bd del método save de la clase BeautyDateService en la clase BeautyDateServiceTests en el método testSaveBeautyCenterCorrect.
    - Prueba UI de la HU en la clase InTimeBeautyDateSuccessUITest.

Caso negativo:

* Solicito una cita para una mascota sin citas previas. Me devuelve un error debido a que elegí una cita para un domingo.
  + - Prueba unitaria del método de BeautyDateController processCreationFormBeautyDate en BeautyDateControllerTests mediante el método testProcessCreateBeautyDateFormErrorWeekendDate.
    - Prueba E2E del método de BeautyDateController processCreationFormBeautyDate en BeautyDateControllerE2ETests mediante el método testProcessCreateBeautyDateFormErrorWeekendDate.
    - Prueba de servicio e integración con la bd del método save de la clase BeautyDateService en la clase BeautyDateServiceTests en el método testSavingBeautyIncorrectWeekend.
  + Solicito una cita para una mascota habiendo concurrencia.
    - Prueba unitaria del método de BeautyDateController processCreationFormBeautyDate en BeautyDateControllerTests mediante el método testProcessCreateBeautyDateFormErrorPetAlreadyHasDate.
    - Prueba E2E del método de BeautyDateController processCreationFormBeautyDate en BeautyDateControllerE2ETests mediante el método testProcessCreateBeautyDateFormErrorPetAlreadyHasDate.
    - Prueba de servicio e integración con la bd del método save de la clase BeautyDateService en la clase BeautyDateServiceTests en el método testSavingBeautyDateConcurrent.
  + Solicito una cita para una mascota fuera de horario.
    - Prueba unitaria del método de BeautyDateController processCreationFormBeautyDate en BeautyDateControllerTests mediante el método testProcessCreateBeautyDateFormErrorTimeDate.
    - Prueba E2E del método de BeautyDateController processCreationFormBeautyDate en BeautyDateControllerE2ETests mediante el método testProcessCreateBeautyDateFormErrorTimeDate.
    - Prueba de servicio e integración con la bd del método save de la clase BeautyDateService en la clase BeautyDateServiceTests en el método testSavingBeautyIncorrectTime.
    - Prueba UI de la HU en la clase InTimeBeautyDateErrorUITest.

1. **Como** esteticista **quiero** ver todas las citas futuras según un filtro seleccionado solicitadas a algunos de mis servicios **para** saber cuando tengo que trabajar.

Caso positivo:

* + Al seleccionar el listado de citas, se despliega una tabla de citas con los correspondientes datos de los usuarios que la han solicitado y de sus mascotas, y cuyo servicio se corresponde con el rol de esteticista.
    - * + Prueba unitaria del método searchBeautyDates(…) de BeauticianController en el método testSearchBeautyDates() de BeauticianControllerTests.
        + Prueba E2E del método searchBeautyDates(…) de BeauticianController en el método testSearchBeautyDates() de BeauticianControllerE2ETests.
        + Prueba unitaria del método searchBeautyDatesId(…) de BeauticianController en el método testSearchBeautyDatesId () de BeauticianControllerTests.
        + Prueba E2E del método searchBeautyDatesId(…) de BeauticianController en el método testSearchBeautyDatesId() de BeauticianControllerE2ETests.
        + Prueba unitaria del método showBeautyDates(…) de BeauticianController en el método testShowBeautyDates() de BeauticianControllerTests.
        + Prueba E2E del método showBeautyDates(…) de BeauticianController en el método testShowBeautyDates() de BeauticianControllerE2ETests.

Casos negativos:

* + El usuario no tiene permiso para acceder a la página y se le redirecciona a la página de error.
    - * + Prueba unitaria del método searchBeautyDatesId(…) de BeauticianController en el método testSearchBeautyDatesIdNoPermission() de BeauticianControllerTests.
        + Prueba E2E del método searchBeautyDatesId(…) de BeauticianController en el método testSearchBeautyDatesIdNoPermission() de BeauticianControllerTests.
        + Prueba unitaria del método showBeautyDates(…) de BeauticianController en el método testShowBeautyDatesNoPermission() de BeauticianControllerTests.
        + Prueba E2E del método showBeautyDates(…) de BeauticianController en el método testShowBeautyDatesNoPermission() de BeauticianControllerE2ETests.
  + La fecha introducida es una fecha pasada.
    - * + Prueba unitaria del método showBeautyDates(…) de BeauticianController en el método testShowBeautyDatesWrongDate() de BeauticianControllerTests.
        + Prueba E2E del método showBeautyDates(…) de BeauticianController en el método testShowBeautyDatesWrongDate() de BeauticianControllerE2ETests.

1. **Como** propietario de una mascota **quiero** visualizar los servicios disponibles filtrados por animales **para** ver las diversas opciones estéticas que tengo para mi animal.

Caso positivos:

* Con una sesión iniciada con rol Owner, introduzco un tipo de animal para realizar una búsqueda de productos según animal.
  + - * + Prueba unitaria del método searcheautyCenter(…) de OwnerController en el método testSearchBeautyCenter() de OwnerControllerTests.
* Con una sesión iniciada con rol Owner, muestro la lista de servicios asociados a un tipo de mascota.
  + - * + Prueba unitaria del método showBeautyDates(…) de OwnerController en el método testShowBeautyDate() de OwnerControllerTests.

1. **Como** propietario de una mascota **quiero** poder solicitar una cita de un servicio de un esteticista **para** que mi mascota lo reciba y tenga un mejor aspecto.

Casos positivos:

* Accedo al formulario de creación de la cita.
  + - * Prueba unitaria del método initCreationFormBeautyDate de la clase BeautyDateController en el método initCreationFormBeautyDateTest de la clase BeautyDateControllerTest.
      * Prueba E2E del método initCreationFormBeautyDate d la clase BeautyDateController en el método initCreationFormBeautyDateTest de la clase BeautyDateControllerE2ETest.
* Relleno correctamente el formulario de la cita con todos sus atributos.
  + - * Prueba unitaria del método processCreationFormBeautyDate de la clase BeautyDateController en el método testProcessCreateBeautyDateFormSuccess en la clase BeautyDateControllerTests.
      * Prueba de servicios e integración de la bd del método saveBeautyDate de la clase BeautyDateService en el método testSavingBeautyDateCorrect de la clase BeautyDateServiceTests.
      * Prueba E2E del método processCreationFormBeautyDate de la clase BeautyDateController en el método testProcessCreateBeautyDateFormSuccess en la clase BeautyDateControllerE2ETests.

Casos negativos:

* Relleno el formulario dejando la fecha vacía y salta error.
  + - * Prueba unitaria del método procesCreationFormBeautyDate de la clase BeautyDateController en el método testProcessCreateBeautyDateFormErrorEmptyDate de la clase BeautyDateControllerTests.
      * Prueba de servicios e integración de la bd del método saveBeautyDate de la clase BeautyDateService en el método testSavingBeautyDateIncorrectEmptyPet de la clase BeautyDateServiceTests.
      * Prueba E2E del método processCreationFormBeautyDate de la clase BeautyDateController en el método testProcessCreateBeautyDateFormErrorEmptyDate.
* Relleno el formulario dejando vacío el campo de mascota y salta un error.
  + - * Prueba unitaria del método procesCreationFormBeautyDare de la clase BeautyDateController en el método testProcessCreateBeautyDateFormErrorEmptyPet de la clase BeautyDateControllerTests.
      * Prueba de servicios e integración de la bd del método saveBeautyDate de la clase BeautyDateService en el método testSavingBeautyDateIncorrectEmptyPet de la clase BeautyDateServiceTests.
      * Prueba E2E del método processCreationFormBeautyDate de la clase BeautyDateController en el método testProcessPet.

1. **Como** propietario de una mascota **quiero** ver mis citas a los diversos servicios estéticos tanto pasadas como futuras **para** recordar cuando tenía que asistir o cuándo fue la última vez que asistió mi mascota.

Caso positivo:

* + El propietario lista correctamente las citas pasadas y futuras. Dicho caso se prueba en las siguientes clases:
    - Prueba unitaria del método de OwnerController showBeautyDates en OwnerControllerTests en el método testShowBeautyDate.
    - Prueba de interfaz de usuario del método de OwnerController showBeautyDates en la clase ListOwnerBeautyDatesSuccessUITest.

Caso negativo:

* + Un propietario intenta listar citas que no son suyas mediante la url y la aplicación le lanza un mensaje de error. Dicho caso se prueba en las siguientes clases:
    - Prueba unitaria del método de OwnerController showBeautyDates en OwnerControllerTests en el método testProcessDeleteBeautyDateFormWithErrors.
    - Prueba de interfaz de usuario del método de OwnerController showBeautyDates en la clase ListOwnerBeautyDatesAuthorErrorUITest.

1. **Como** propietario de una mascota **quiero** poder eliminar una cita aún no realizada **para** cancelarla por si no puedo asistir.

Caso positivo:

* Al seleccionar el listado de mis citas en la aplicación, aparece un listad con las citas que ha pedido el usuario y aún no se han efectuado y al seleccionar la opción eliminar se elimina de la base de datos.
* Prueba unitaria del método de OwnerController deleteBeautyDate en OwnerControllerTests en el método testProcessDeleteBeautyDateFormSuccess.

Caso negativo:

* + Un propietario accede a eliminar un servicio que no es suyo mediante la url y la aplicación le lanza un mensaje de error.
* Prueba unitaria del método de OwnerController deleteBeautyDate en OwnerControllerTests en el método testProcessDeleteBeautyDateFormWithErrors.

1. **Como** propietario de una mascota **quiero** poder modificar la fecha y hora de una cita aún no realizada a un horario nuevo disponible **para** cambiarla en caso de que me surja un contratiempo para asistir en dicho momento.

Casos positivos:

* Muestro por pantalla los datos iniciales de la cita.
  + - Prueba unitaria del método initUpdateBeautyDateForm de la clase BeautyDateController en el método initUpdateBeautyDateFormTest de la clase BeautyDateControllerTest.
    - Prueba E2E del método initUpdateBeautyDateForm de la clase BeautyDateController en el método initUpdateBeautyDateFormTest de la clase BeautyDateControllerE2ETest.
* Actualizo correctamente todos los atributos y los guardo.
* Prueba unitaria del método processUpdateBeautyDateForm de la clase BeautyDateController en el método testProcessUpdateBeautyDateFormSuccess de la clase BeautyDateControllerTests.
* Prueba de servicios e integración de la bd del método saveBeautyDate en el método testSavingBeautyDateCorrect de BeautyDateServiceTests.
* Prueba E2E del método processUpdateBeautyDateForm de la clase BeautyDateController en el método testProcessUpdateBeautyDateFormSuccess en la clase BeautyDateControllerE2ETests.
* Actualizo correctamente todos los atributos dejando en blanco la descripción.
* Prueba unitaria del método processUpdateBeautyDateForm de la clase BeautyDateController en el método testProcessUpdateBeautyDateNoDescriptionFormSuccess de la clase BeautyDateControllerTests.
* Prueba de servicios e integración de la bd del método saveBeautyDate en el método testSavingBeautyDateCorrectNoDescription de BeautyDateServiceTests.
* Prueba E2E del método processUpdateBeautyDateForm de la clase BeautyDateController en el método testProcessUpdateBeautyDateNoDescriptionFormSuccess en la clase BeautyDateControllerE2ETests.

Casos negativos:

* Intento acceder a las citas sin ser el autor de ellas.
* Prueba unitaria del método initCreationFormBeautyDate de la clase BeautyDateController en el método testBeautyDateNoAuthorForm en la clase BeautyDateControllerTests.
* Prueba E2E del método initCreationFormBeautyDate de la clase BeautyDateController en el método testBeautyDateNoAuthorForm en la clase BeautyDateControllerE2ETests.
* Actualizo la cita dejando la fecha en blanco y salta error.
* Prueba unitaria del método processUpdateBeautyDateForm de la clase BeautyDateController en el método testProcessUpdateBeautyDateEmptyDateFormError de la clase BeautyDateControllerTests.
* Prueba de servicios e integración de la bd del método saveBeautyDate en el método testSavingBeautyDateIncorrectEmptyDate de BeautyDateServiceTests.
* Prueba E2E del método processUpdateBeautyDateForm de la clase BeautyDateController en el método testProcessUpdateBeautyDateEmptyDateFormError en la clase BeautyDateControllerE2ETests.
* Actualizo la cita seleccionando una fecha en fin de semana y salta error.
* Prueba unitaria del método processUpdateBeautyDateForm de la clase BeautyDateController en el método testProcessUpdateBeautyDateWeekendDateFormError de la clase BeautyDateControllerTests.
* Prueba de servicios e integración de la bd del método saveBeautyDate en el método testSavingBeautyDateIncorrectWeekend de BeautyDateServiceTests.
* Prueba E2E del método processUpdateBeautyDateForm de la clase BeautyDateController en el método testProcessUpdateBeautyDateWeekendDateFormError en la clase BeautyDateControllerE2ETests.
* Actualizo la cita seleccionando una fecha que ya está cogida y salta error.
* Prueba unitaria del método processUpdateBeautyDateForm de la clase BeautyDateController en el método testProcessUpdateBeautyDateConcurrentDateFormError de la clase BeautyDateControllerTests.
* Prueba de servicios e integración de la bd del método saveBeautyDate en el método testSavingBeautyDateConcurrent de BeautyDateServiceTests.
* Prueba E2E del método processUpdateBeautyDateForm de la clase BeautyDateController en el método testProcessUpdateBeautyDateConcurrentDateFormError en la clase BeautyDateControllerE2ETests.
* Actualizo la cita seleccionando una hora que no está dentro del horario del Beautician y salta error.
* Prueba unitaria del método processUpdateBeautyDateForm de la clase BeautyDateController en el método testProcessUpdateBeautyDateIncorrectTimeFormError de la clase BeautyDateControllerTests.
* Prueba de servicios e integración de la bd del método saveBeautyDate en el método testSavingBeautyDateIncorrectTime de BeautyDateServiceTests.
* Prueba E2E del método processUpdateBeautyDateForm de la clase BeautyDateController en el método testProcessUpdateBeautyDateIncorrectTimeFormError en la clase BeautyDateControllerE2ETests.
* Actualizo la cita dejando el campo mascota vacío y salta error.
* Prueba unitaria del método processUpdateBeautyDateForm de la clase BeautyDateController en el método testProcessUpdateBeautyDateEmptyPetFormError de la clase BeautyDateControllerTests.
* Prueba de servicios e integración de la bd del método saveBeautyDate en el método testSavingBeautyDateEmptyPet de BeautyDateServiceTests.
* Prueba E2E del método processUpdateBeautyDateForm de la clase BeautyDateController en el método testProcessUpdateBeautyDateEmptyPetFormError en la clase BeautyDateControllerE2ETests.

1. **Como** propietario de una mascota **quiero** que exista la posibilidad de elegir los productos que quiero que usen en mi mascota en una cita solicitada sobre un servicio **para** que usen la mejor calidad sobre mi pequeño animal.

Caso positivo:

* + Selecciono los productos necesarios dentro del listado de productos disponibles para el servicio que he elegido para mi mascota. Dicho caso se prueba en las siguientes clases:
    - Prueba unitaria del método de BeautyDateController initCreationFormBeautyDate en BeautyDateControllerTests en el método testInitCreationForm.
    - Prueba unitaria del método de BeautyDateController processCreationFormBeautyDate en BeautyDateControllerTests en el método testProcessCreateBeautyDateFormSuccess.
    - Prueba de interfaz de usuario del método BeautyDateController processCreationFormBeautyDate en ChooseProductsInBeautyDateSuccessUITest.
    - Prueba E2E del método de BeautyDateController processCreationFormBeautyDate en BeautyDateControllerE2ETests en testProcessCreateBeautyDateFormSuccess.

Caso negativo:

* + A la hora de elegir un producto, escojo uno que no es correspondiente con el servicio seleccionado y se informa del error de que el producto no está disponible para el servicio.
    - Prueba unitaria del método de BeautyDateController processCreationFormBeautyDate en BeautyDateControllerTests en el método testProcessCreateBeautyDateFormErrorEmptyDate.
    - Prueba unitaria del método de BeautyDateController processCreationFormBeautyDate en BeautyDateControllerTests en el método testProcessCreateBeautyDateFormErrorEmptyPet.
    - Prueba E2E del método de BeautyDateController processCreationFormBeautyDate en BeautyDateControllerE2ETests en testProcessCreateBeautyDateFormErrorEmptyDate.
    - Prueba E2E del método de BeautyDateController processCreationFormBeautyDate en BeautyDateControllerE2ETests en testProcessCreateBeautyDateFormErrorEmptyPet.

1. **Como** usuario del sistema **quiero** poder publicar una petición de recogida de un animal, indicando el tipo de animal, su estado físico, un estado de la petición que indique si un veterinario la aceptado o aún no, una dirección de donde fue encontrado y una descripción **para** que la clínica se haga cargo de él.

Casos positivos:

* Accedo al formulario para crear una petición y se muestra.
* Prueba unitaria del método initCreatePickUpRequestForm de la clase PickUpRequestController en el método testInitCreationForm de la clase PickUpRequestControllerTests.
* Prueba E2E del método initCreatePickUpRequestForm de la clase PickUpRequestController en el método testInitCreationForm de la clase PickUpRequestControllerE2ETests.
* Creo una petición de recogida correctamente rellenando todos los atributos.
* Prueba unitaria del método processCreatePickUpRequestForm de la clase PickUpRequestController en el método testProcessCreationFormSuccess de la clase PickUpRequestControllerTests.
* Prueba de servicios e integración de la bd del método savePickUpRequest de la clase PickUpRequestService en el método testSavingPickUpRequestCorrect de la clase PickUpRequestServiceTests.
* Prueba E2E del método processCreatePickUpRequestForm de la clase PickUpRequestController en el método testProcessCreationFormSuccess de la clase PickUpRequestControllerE2ETests.
* Creo una petición de recogida correctamente sin descripción.
* Prueba unitaria del método processCreatePickUpRequestForm de la clase PickUpRequestController en el método testProcessCreationFormNoDescriptionSuccess de la clase PickUpRequestControllerTests.
* Prueba de servicios e integración de la bd del método savePickUpRequest de la clase PickUpRequestService en el método testSavingPickUpRequestNoDescriptionCorrect de la clase PickUpRequestServiceTests.
* Prueba E2E del método processCreatePickUpRequestForm de la clase PickUpRequestController en el método testProcessCreationFormNoDescriptionSuccess de la clase PickUpRequestControllerE2ETests.
* Creo una petición de recogida sin estado físico ni descripción.
* Prueba unitaria del método processCreatePickUpRequestForm de la clase PickUpRequestController en el método testProcessCreationFormNoDescriptionAndPhysicalStatusSuccess de la clase PickUpRequestControllerTests.
* Prueba de servicios e integración de la bd del método savePickUpRequest de la clase PickUpRequestService en el método testSavingPickUpRequestNoPhysicalStatusCorrect de la clase PickUpRequestServiceTests.
* Prueba E2E del método processCreatePickUpRequestForm de la clase PickUpRequestController en el método testProcessCreationFormNoDescriptionAndPhysicalStatusSuccess de la clase PickUpRequestControllerE2ETests.

Casos negativos:

* Intento acceder al formulario con un id que no es el de mi usuario.
* Prueba unitaria del método initCreatePickUpRequestForm de la clase PickUpRequestController en el método testInitCreationFormError de la clase PickUpRequestControllerTests.
* Prueba E2E del método initCreatePickUpRequestForm de la clase PickUpRequestController en el método testInitCreationFormError de la clase PickUpRequestControllerE2ETests.
* Relleno el formulario dejando en blanco la dirección y salta error.
* Prueba unitaria del método processCreatePickUpRequestForm de la clase PickUpRequestController en el método testProcessCreationFormNoAddressIncorrect de la clase PickUpRequestControllerTests.
* Prueba de servicios e integración de la bd del método savePickUpRequest de la clase PickUpRequestService en el método testSavingPickUpRequestNoAddressIncorrect de la clase PickUpRequestServiceTests.
* Prueba E2E del método processCreatePickUpRequestForm de la clase PickUpRequestController en el método testProcessCreationFormNoAddressIncorrect de la clase PickUpRequestControllerE2ETests.
* Relleno el formulario dejando el tipo de mascota vacío y salta error.
* Prueba unitaria del método processCreatePickUpRequestForm de la clase PickUpRequestController en el método testProcessCreationFormNoPetTypeIncorrect de la clase PickUpRequestControllerTests.
* Prueba de servicios e integración de la bd del método savePickUpRequest de la clase PickUpRequestService en el método testSavingPickUpRequestNoPetTypeIncorrect de la clase PickUpRequestServiceTests.
* Prueba E2E del método processCreatePickUpRequestForm de la clase PickUpRequestController en el método testProcessCreationFormNoPetTypeIncorrect de la clase PickUpRequestControllerE2ETests.
* Relleno el formulario dejando el campo aceptado vacío y salta error.
* Prueba unitaria del método processCreatePickUpRequestForm de la clase PickUpRequestController en el método testProcessCreationFormNoAcceptedIncorrect de la clase PickUpRequestControllerTests.
* Prueba de servicios e integración de la bd del método savePickUpRequest de la clase PickUpRequestService en el método testSavingPickUpRequestNoAcceptedIncorrect de la clase PickUpRequestServiceTests.
* Prueba E2E del método processCreatePickUpRequestForm de la clase PickUpRequestController en el método testProcessCreationFormNoAcceptedIncorrect de la clase PickUpRequestControllerE2ETests.
* Muestro el listado de peticiones en pantalla.
* Prueba unitaria del método showPickUpRequests de la clase PickUpRequestController en el método testInitShowForm de la clase PickUpRequestControllerTests.
* Prueba E2E del método showPickUpRequests de la clase PickUpRequestController en el método testInitShowForm de la clase PickUpRequestControllerE2ETests.
* Intento mostrar el listado de peticiones de un usuario que no es el mío y salta error.
* Prueba unitaria del método showPickUpRequests de la clase PickUpRequestController en el método testInitShowInvalidAccessForm de la clase PickUpRequestControllerTests.
* Prueba E2E del método showPickUpRequests de la clase PickUpRequestController en el método testInitShowInvalidAccessForm de la clase PickUpRequestControllerE2ETests.

1. **Como** usuario del sistema **quiero** poder eliminar la petición de recogida de un animal que aún no haya sido aceptado por un veterinario **para** poder quedarme con el animal por si cambio de opinión.

Caso positivo:

* + El usuario elimina correctamente la petición de recogida.
    - Prueba unitaria del método de PickUpRequestController deletePickUpRequest en PickUpRequestControllerTests mediante el método testProcessDeleteOwnerPickUpRequestSuccess.
    - Prueba E2E del método de PickUpRequestController deletePickUpRequest en PickUpRequestControllerE2ETests mediante el método testProcessDeleteOwnerPickUpRequestSuccess.

Caso negativo:

* + El usuario trata de eliminar la petición una vez que se ha aceptado por un veterinario, llevándole a una página de error.
    - Prueba unitaria del método de PickUpRequestController deletePickUpRequest en PickUpRequestControllerTests mediante el método testProcessDeleteOwnerPickUpRequestError.
    - Prueba E2E del método de PickUpRequestController deletePickUpRequest en PickUpRequestControllerE2ETests mediante el método testProcessDeleteOwnerPickUpRequestError.

1. **Como** veterinario **quiero** poder listar y mostrar las peticiones de recogida subidas por usuarios **para** valorar si me puedo responsabilizar de un animal abandonado mientras es adoptado.

Caso positivo:

* + Al seleccionar la opción de peticiones de recogida aparece un listado con todas las peticiones de recogida de animales subidas por los usuarios.
    - Prueba unitaria del método de PickUpRequestController en listPickUpRequets en PickUpRequestControllerTests en el método testInitShowForm.
    - Prueba de interfaz de usuario del método en PickUpRequestController listPickUpRequests en ListPickUpRequestsSuccessUITests.
    - Prueba E2E del método en PickUpRequestController listPickUpRequests en PickUpRequestControllerE2ETEsts en el método testInitShowForm.

Caso negativo:

* + Un propietario intenta listar citas que no son suyas mediante la url y la aplicación le lanza un mensaje de error.
    - Prueba unitaria del método de PickUpRequestController en listPickUpRequets en PickUpRequestControllerTests en el método testInitShowInvalidAccessForm.
    - Prueba de interfaz de usuario del método en PickUpRequestController listPickUpRequests en ListPickUpRequestsErrorUITests.
    - Prueba E2E del método en PickUpRequestController listPickUpRequests en PickUpRequestControllerE2ETEsts en el método testInitShowInvalidAccessForm.

1. **Como** veterinario **quiero** poder aceptar o rechazar peticiones de recogida de animales, añadiendo un campo de contacto **para** responsabilizarme de un animal abandonado y ponerlo en adopción.

Caso positivo:

* Accedo al formulario de una solicitud abierta.
  + - Prueba unitaria del método de PickUpRequestController initUpdatePickUpRequest en PickUpRequestControllerTests mediante el método testProcessUpdatePickUpRequestAsVetSuccess.
    - Prueba E2E del método de PickUpRequestController initUpdatePickUpRequest en PickUpRequestControllerE2ETests mediante el método testProcessUpdatePickUpRequestAsVetSuccess.
* Deniego una solicitud aún abierta.
  + - Prueba unitaria del método de PickUpRequestController processUpdatePickUpRequest en PickUpRequestControllerTests mediante el método testProcessUpdateFormSuccess.
    - Prueba E2E del método de PickUpRequestController processUpdatePickUpRequest en PickUpRequestControllerE2ETests mediante el método testProcessUpdateFormSuccess.

Caso negativo:

* Accedo al formulario de una solicitud cerrada.
  + - Prueba unitaria del método de PickUpRequestController initUpdatePickUpRequest en PickUpRequestControllerTests mediante el método testProcessUpdatePickUpRequestAsVetError.
    - Prueba E2E del método de PickUpRequestController initUpdatePickUpRequest en PickUpRequestControllerE2ETests mediante el método testProcessUpdatePickUpRequestAsVetError.
* Intento aceptar una solicitud cerrada.
  + - Prueba unitaria del método de PickUpRequestController processUpdatePickUpRequest en PickUpRequestControllerTests mediante el método testProcessUpdateFormError.
  + Prueba E2E del método de PickUpRequestController processUpdatePickUpRequest en PickUpRequestControllerE2ETests mediante el método testProcessUpdateFormError.

1. **Como** veterinario **quiero** poder eliminar las peticiones de recogidas aceptadas **para** no confundir a las familias de acogida en potencia sobre animales ya adoptados.

Caso positivo:

* + Un veterinario elimina una petición de recogida del sistema ya aceptada porque el animal ya ha sido adoptado.
    - * + Prueba unitaria del método deletePickUpRequest(…) de PickUpRequestController en el método testProcessDeletePickUpRequestAsVetSuccess() de PickUpRequestControllerTests.
        + Prueba E2E del método deletePickUpRequest(…) de PickUpRequestController en el método testProcessDeletePickUpRequestAsVetSuccess() de PickUpRequestControllerE2ETests.
        + Prueba de interfaz de usuario del método deletePickUpRequest(…) de PickUpRequestController en DeletePickUpRequestByVetSuccessUITest.

Caso negativo:

* + Un usuario del sistema con un rol diferente al de veterinario intenta eliminar una petición de recogida.
    - * + Prueba de interfaz de usuario del método deletePickUpRequest(…) de PickUpRequestController en DeletePickUpRequestByVetNoPermissionTest.

1. **Planificación de los diferentes sprint**

Para el reparto de tareas se ha planteado centrarse en implementar las historias de usuario relacionadas con el centro de estética para mascotas, denominado BeautyPet, e implementar las historias de usuario más básicas del servicio de acogida. Por ello la planificación resultante sería la siguiente:

* + Sprint 2: Para este sprint, el equipo ha decidido implementar las siguientes historias de usuario:
* Las historias de usuario 1,3,5,7,10,11 que serán implementadas por Carlos Santos Tirado y Mario David Jiménez Hartman.
* Las historias de usuario 2,4,6,8,12,13 que serán implementadas por Andrés Fernández Rodríguez y Gonzalo Guerrero Luna.
* Se realizarán las pruebas unitarias de las historias de usuario mencionadas.
* Se realizarán las pruebas Travis CI de las historias de usuario implementadas.
  + Sprint 3: Para este sprint, el equipo ha decidido implementar las siguientes historias de usuario:
* Las historias de usuario 9,14,16,20, que serán implementadas por Carlos Santos Tirado y Mario David Jiménez Hartman.
* Las historias de usuario 15,17,18,19, que serán implementadas por Andrés Fernández Rodríguez y Gonzalo Guerrero Luna.
* Con ello se realizará el 100% de implementación de historias de usuario, además se implementarán todas las pruebas unitarias y de integración correspondientes, y 24 pruebas de interfaz de usuario de 12 historias (un caso positivo y negativo por cada una).
* Se realizará la implementación de la API Rest de Spotify con nuestra aplicación, además de sus pruebas correspondientes.
  + Sprint 4: Para este sprint, el equipo ha decidido implementar las siguientes historias de usuario: