



### **Plan de calidad RUP**

David S. Madrid Cardozo, Laura V. Hidalgo Melo, Johan S. Zapata Talero & Javier M. Diaz  
Sanabria

Servicio Nacional de Aprendizaje (SENA)

Técnica  
Análisis y Desarrollo de Sistemas de Información

Ing. Miguel Angel López Cacho

29 de marzo del 2022

## INTRODUCCIÓN

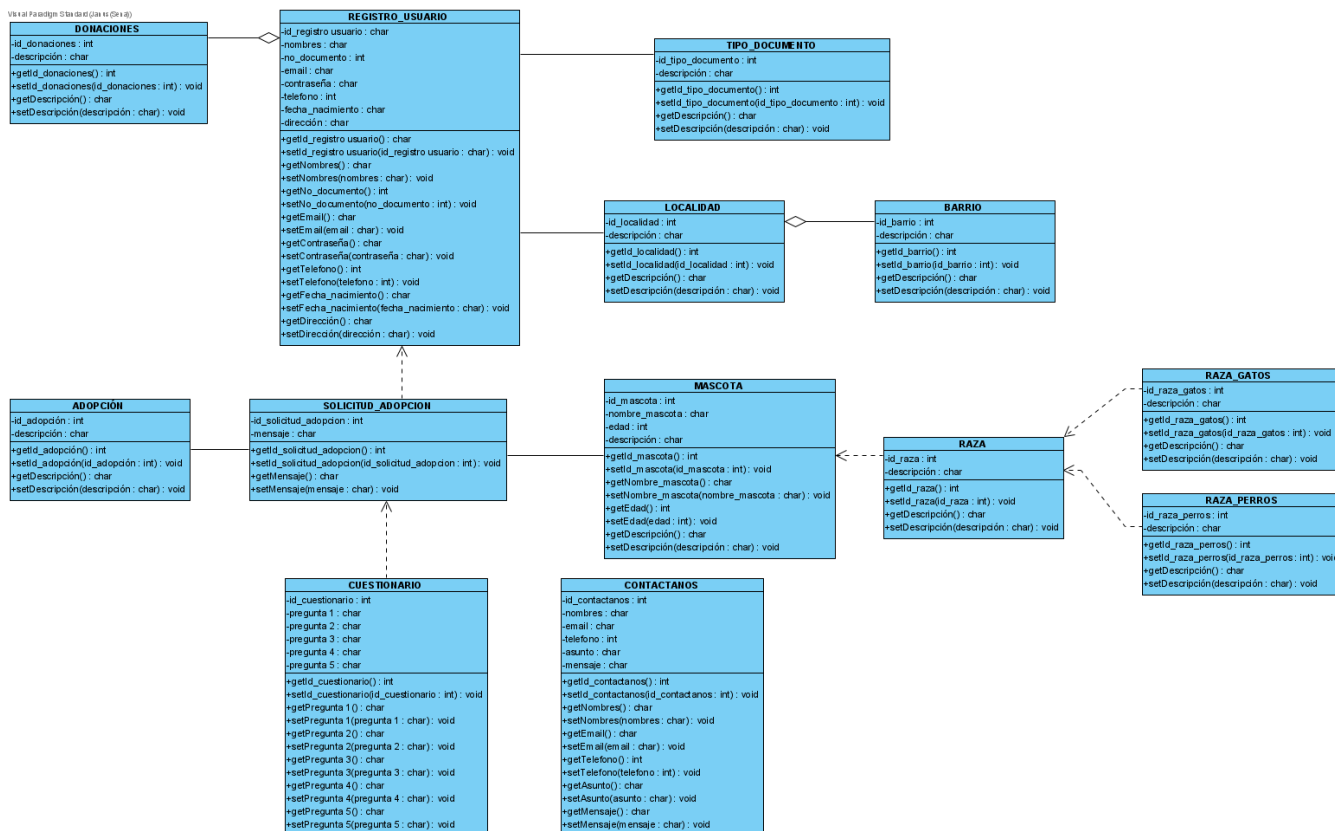
El objetivo de la ingeniería de software es realizar actividades relacionadas con el desarrollo, la operación y el mantenimiento de software de manera sistemática, disciplinada y medible. En otras palabras, el objetivo de la ingeniería de software es crear software de alta calidad a un costo estimado que sea fácil de desarrollar y mantener.

La primera fase ofertada corresponde a las fases Inicio y Elaboración en RUP (Racional Unified Process, la metodología que se propone utilizar). La siguiente fase que se entregara más adelante corresponde al RUP a las fases de construcción y de transición. Es importante recalcar esto puesto que utilizaremos la terminología RUP en este documento. Se incluirá el detalle para las fases de inicio, elaboración y adicionalmente se esbozará las fases posteriores de construcción y transición para dar una visión global de todo proceso.

El enfoque de desarrollo propuesto es una configuración del proceso RUP a partir de las características del proyecto, los roles de los participantes seleccionados, las actividades a realizar y los entregables que serán generados. Este documento es a su vez uno de los artefactos RUP.

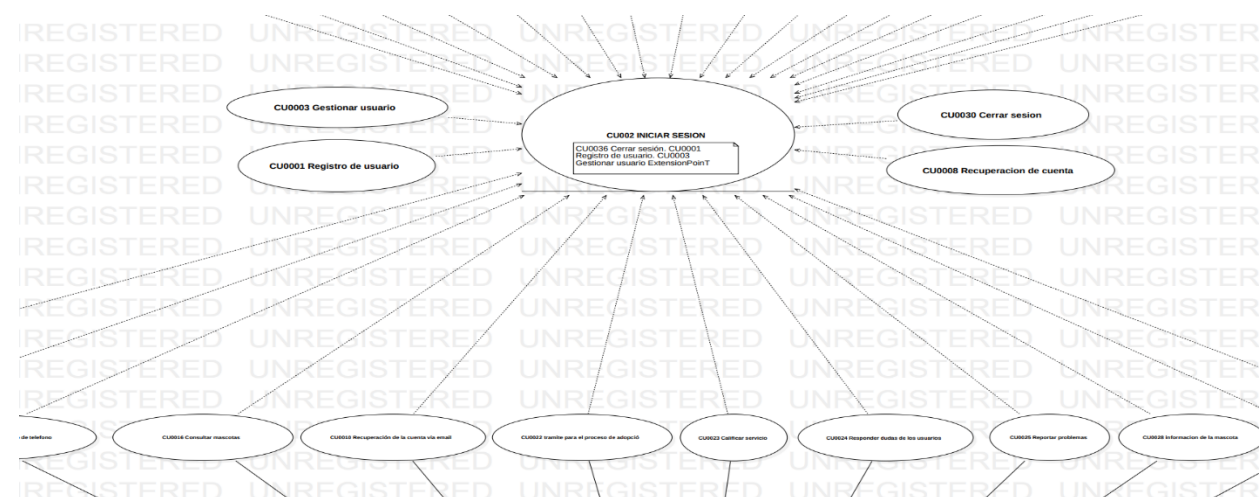
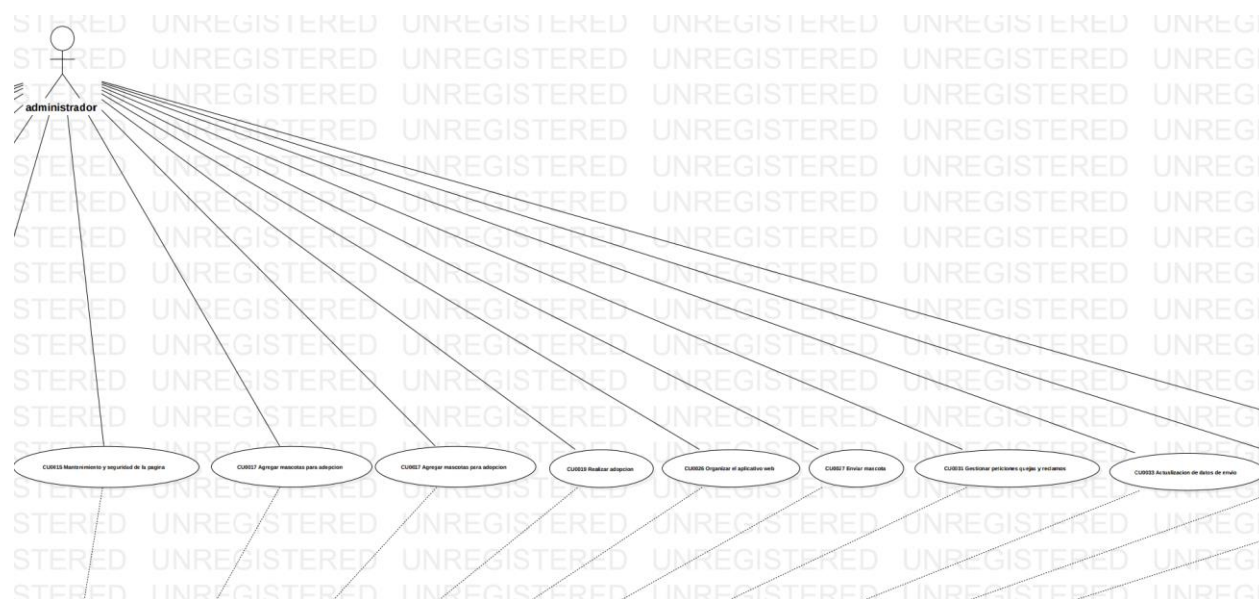
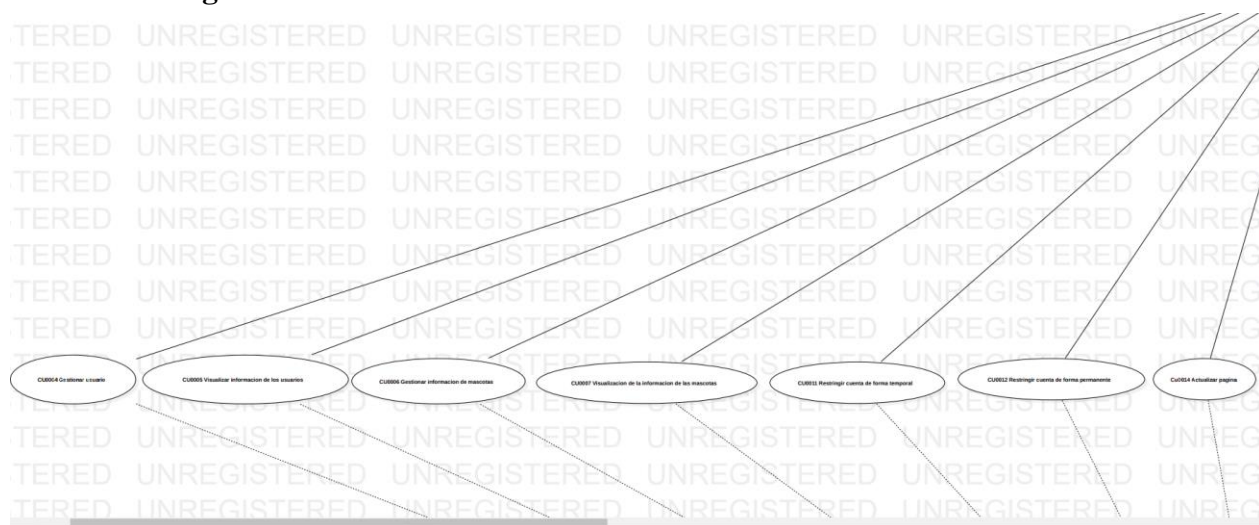
## CAPTURA DE REQUISITOS

### Modelo del dominio



El modelo del dominio es un tipo de diagrama de estructura estática que describe la estructura del sistema del aplicativo web, mostrando las clases de sistema, sus atributos, operaciones y las relaciones entre los objetos.

## Modelo de negocio



[https://drive.google.com/file/d/1E60DQs\\_LEc7dwQ\\_zhcQIxR6sC6Y9zWj8/view?usp=sharing](https://drive.google.com/file/d/1E60DQs_LEc7dwQ_zhcQIxR6sC6Y9zWj8/view?usp=sharing)

Tiene como fin representar y describir las diferentes acciones, procesos, sistemas y los actores que van a hacer parte del aplicativo, como lo son el usuario y el administrador, se puede encontrar la relación entre estos actores y los diferentes casos de uso.

### **Búsqueda de casos de uso a partir del modelo de negocio**

Enlace pdf de todos los casos de uso extendido: Los diagramas de caso de uso modelan la funcionalidad del sistema usando actores y casos de uso. Los casos de uso son servicios o funciones provistas por el sistema para sus usuarios. Un caso de uso es una descripción de las acciones de un sistema desde el punto de vista del usuario. Es una herramienta valiosa dado que es una técnica de aciertos y errores para obtener los requerimientos del sistema, justamente desde el punto de vista del

usuario. <https://drive.google.com/file/d/1espSe2oO8whqTNv37WRXAJIDwuNDb8G2/view?usp=sharing>

### **Requisitos adicionales**

Enlace pdf requerimiento funcionales y no funcionales: El presente documento tiene como propósito definir las especificaciones funcionales, no funcionales para el desarrollo de un sistema informativo web que permitirá gestionar el proceso de adopción de mascotas callejeras.

[https://drive.google.com/file/d/1ratkCIa\\_o5Ps7p9wr7f9teSG3gKclJPI/view?usp=sharing](https://drive.google.com/file/d/1ratkCIa_o5Ps7p9wr7f9teSG3gKclJPI/view?usp=sharing)

## **CAPTURA DE REQUISITOS COMO CASOS DE USO**

### **Trabajadores**

Nombre: Laura Valentina Hidalgo Melo

Rol: Administrador del proyecto

Categoría profesional: ADSI

Responsabilidades: Analizar y desarrollar sistemas de información

Nombre: Johan Sebastián Zapata Talero

Rol: Diseñador

Categoría profesional: ADSI

Responsabilidades: Analizar y desarrollar sistemas de información

Nombre: Javier Mauricio Diaz Sanabria

Rol: Analista

Categoría profesional: ADSI

Responsabilidades: Analizar y desarrollar sistemas de información

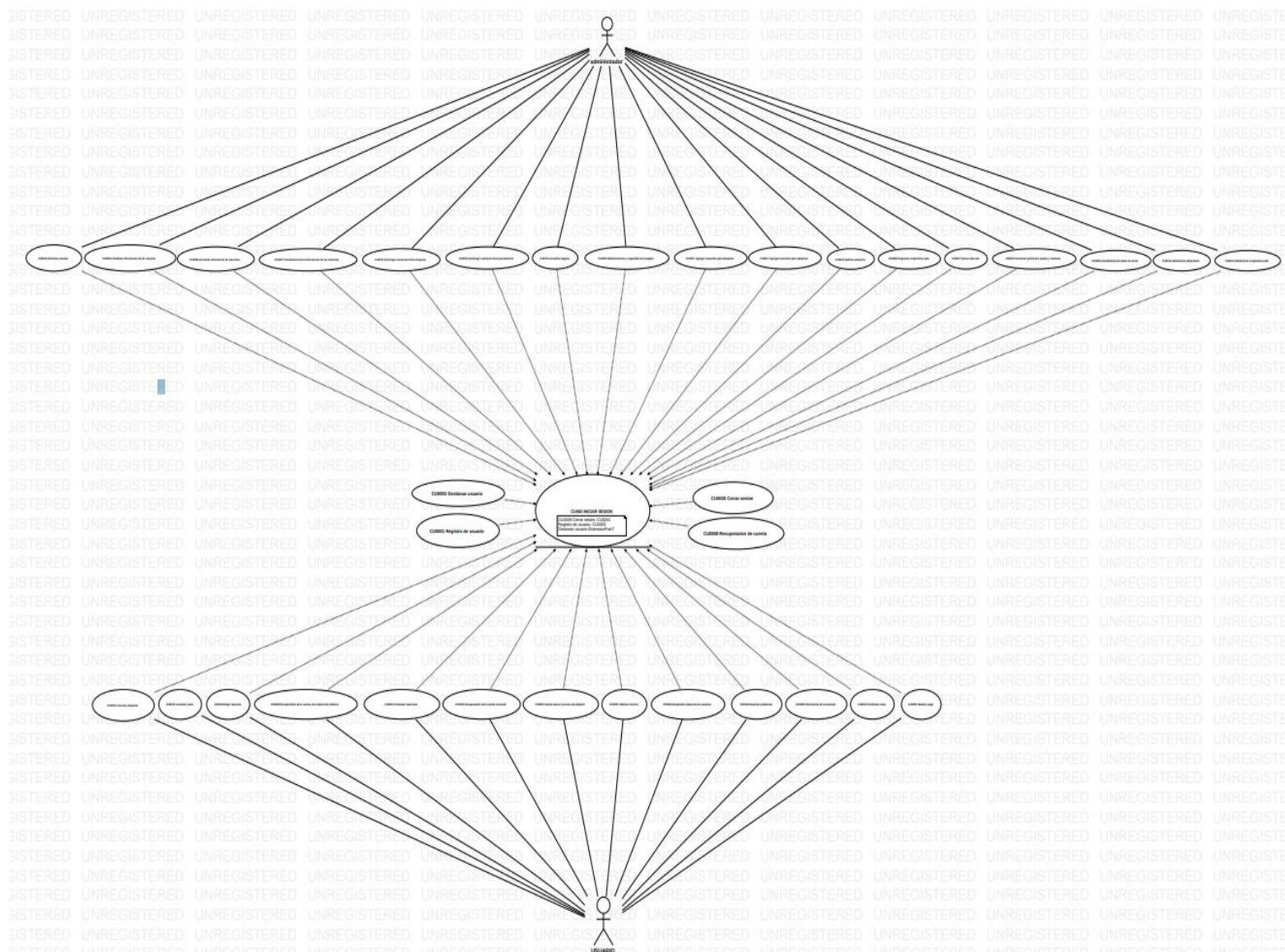
Nombre: David Santiago Madrid Cardozo

Rol: Programador

Categoría profesional: ADSI

Responsabilidades: Analizar y desarrollar sistemas de información

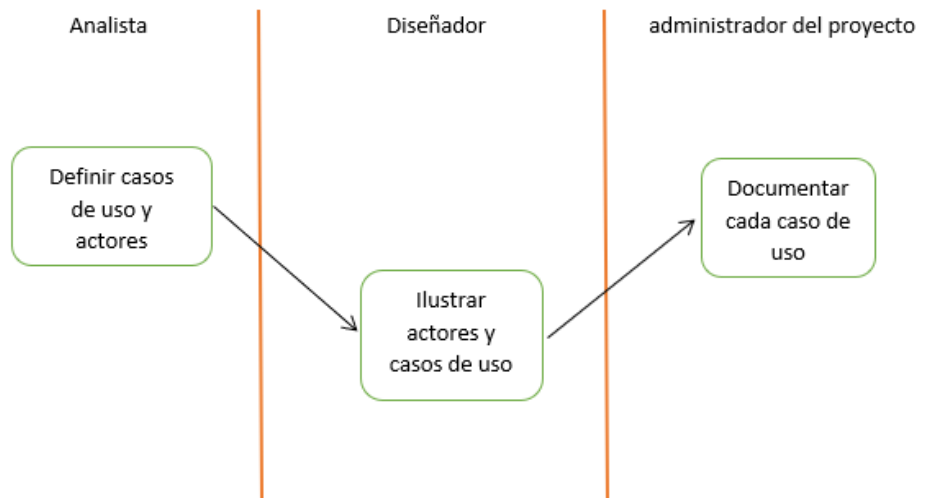
## Artefacto



[https://drive.google.com/file/d/1E60DQs\\_LEc7dwQ\\_zhcQIxR6sC6Y9zWj8/view?usp=sharing](https://drive.google.com/file/d/1E60DQs_LEc7dwQ_zhcQIxR6sC6Y9zWj8/view?usp=sharing)



## Flujo de trabajo



## ANÁLISIS

### Trabajadores

Nombre: Laura Valentina Hidalgo Melo

Rol: Analista de software

Categoría profesional: ADSI

Responsabilidades: Analizar y desarrollar sistemas de información

Nombre: Johan Sebastián Zapata Talero

Rol: Diseñador

Categoría profesional: ADSI

Responsabilidades: Analizar y desarrollar sistemas de información

Nombre: Javier Mauricio Diaz Sanabria

Rol: Desarrollador de software

Categoría profesional: ADSI

Responsabilidades: Analizar y desarrollar sistemas de información

Nombre: David Santiago Madrid Cardozo

Rol: Arquitecto de software

Categoría profesional: ADSI

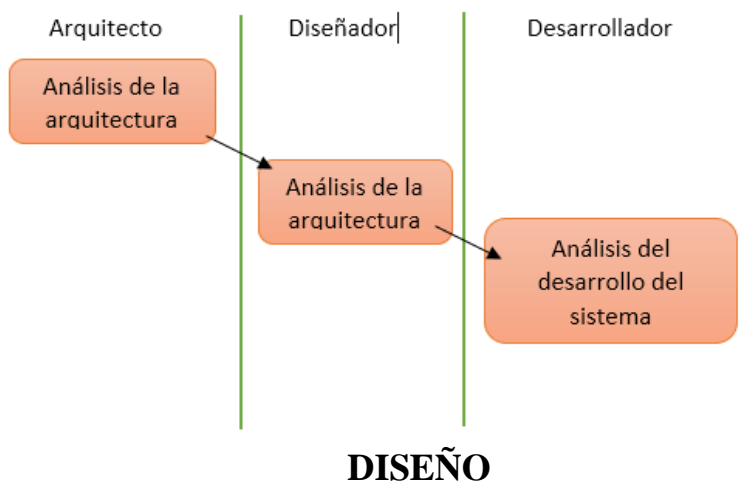
Responsabilidades: Analizar y desarrollar sistemas de información

### Artefactos

Enlace pdf de todos los casos de uso extendido:

<https://drive.google.com/file/d/1espSe2oO8whqTNv37WRXAJIDwuNDb8G2/view?usp=sharing>

## Flujo de trabajo



## Trabajadores

Nombre: Laura Valentina Hidalgo Melo

Rol: Analista de software

Categoría profesional: ADSI

Responsabilidades: Analizar y desarrollar sistemas de información

Nombre: Johan Sebastián Zapata Talero

Rol: Diseñador

Categoría profesional: ADSI

Responsabilidades: Analizar y desarrollar sistemas de información

Nombre: Javier Mauricio Diaz Sanabria

Rol: Desarrollador de software

Categoría profesional: ADSI

Responsabilidades: Analizar y desarrollar sistemas de información

Nombre: David Santiago Madrid Cardozo

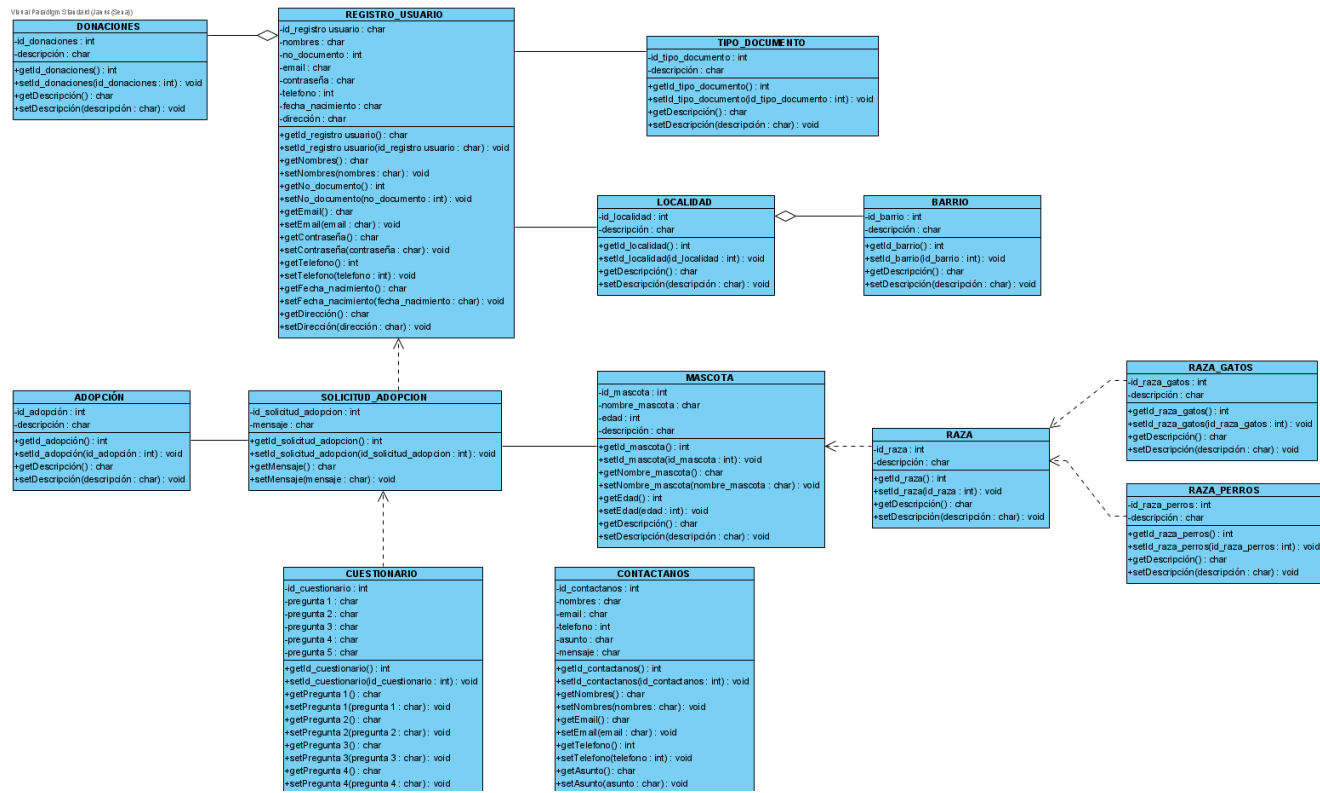
Rol: Arquitecto de software

Categoría profesional: ADSI

Responsabilidades: Analizar y desarrollar sistemas de información



## Artefactos



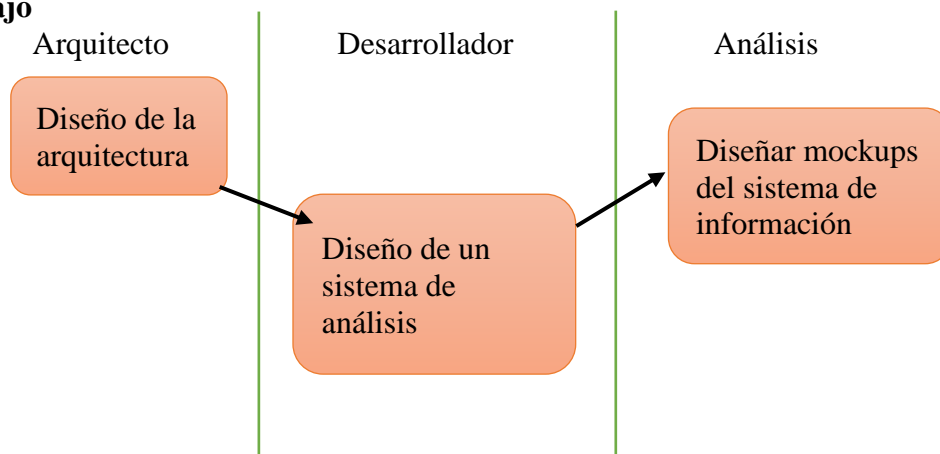
## Interfaz

Diseño de interfaz del sistema de información: pdf con el diseño de las interfaces

<https://drive.google.com/file/d/1x6IXRmTFri8hyRk1RJezcKkU65IPfIJH/view?usp=share>

ng

## Flujo de trabajo



## IMPLEMENTACIÓN

### Trabajadores

Nombre: Laura Valentina Hidalgo Melo

Rol: Analista de software

Categoría profesional: ADSI

Responsabilidades: Analizar y desarrollar sistemas de información

Nombre: Johan Sebastián Zapata Talero

Rol: Arquitecto de software

Categoría profesional: ADSI

Responsabilidades: Analizar y desarrollar sistemas de información

Nombre: Javier Mauricio Diaz Sanabria

Rol: Desarrollador de software

Categoría profesional: ADSI

Responsabilidades: Analizar y desarrollar sistemas de información

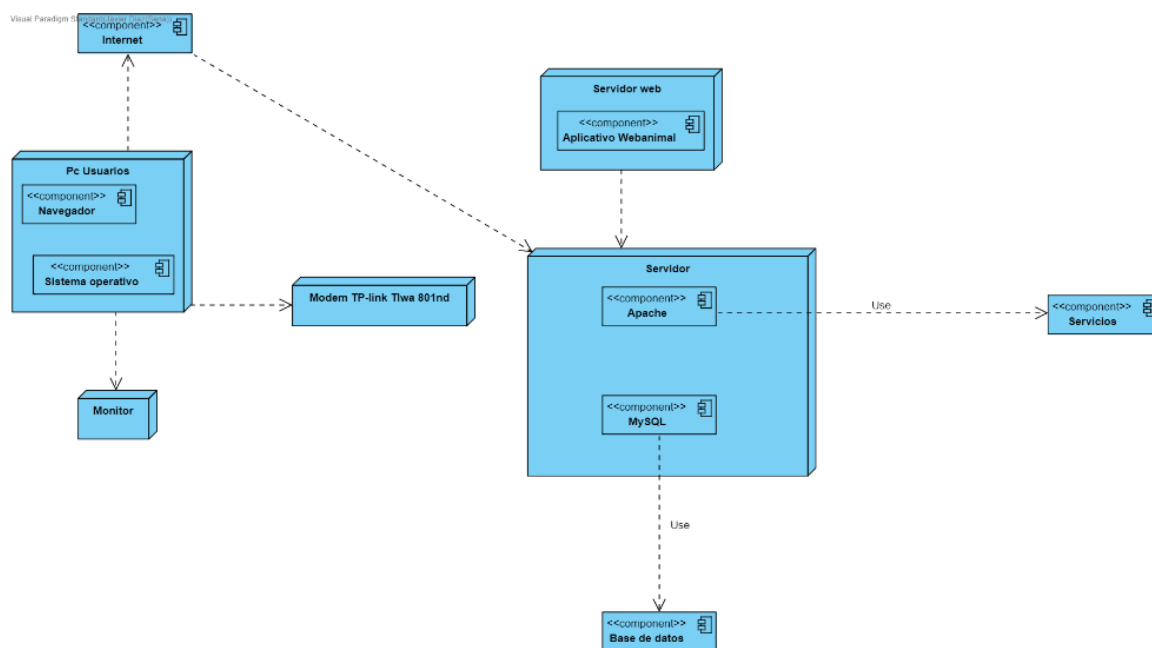
Nombre: David Santiago Madrid Cardozo

Rol: Tester

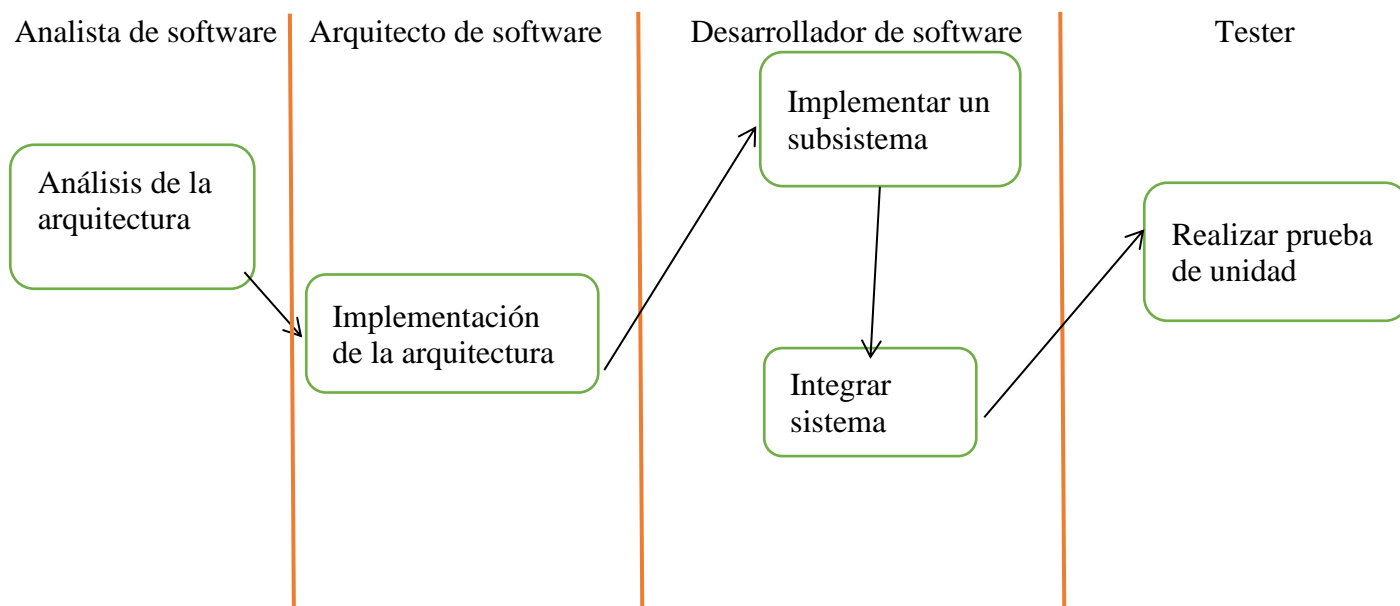
Categoría profesional: ADSI

Responsabilidades: Analizar y desarrollar sistemas de información

### Artefactos



## Flujo de trabajo



## Pruebas

### Trabajadores

Nombre: Laura Valentina Hidalgo Melo

Rol: Analista de software

Categoría profesional: ADSI

Responsabilidades: Analizar y desarrollar sistemas de información

Nombre: Johan Sebastián Zapata Talero

Rol: Diseñador

Categoría profesional: ADSI

Responsabilidades: Analizar y desarrollar sistemas de información

Nombre: Javier Mauricio Diaz Sanabria

Rol: Desarrollador de software

Categoría profesional: ADSI

Responsabilidades: Analizar y desarrollar sistemas de información

Nombre: David Santiago Madrid Cardozo

Rol: Arquitecto de software

Categoría profesional: ADSI

Responsabilidades: Analizar y desarrollar sistemas de información

## Artefacto

### Pruebas de caja negra

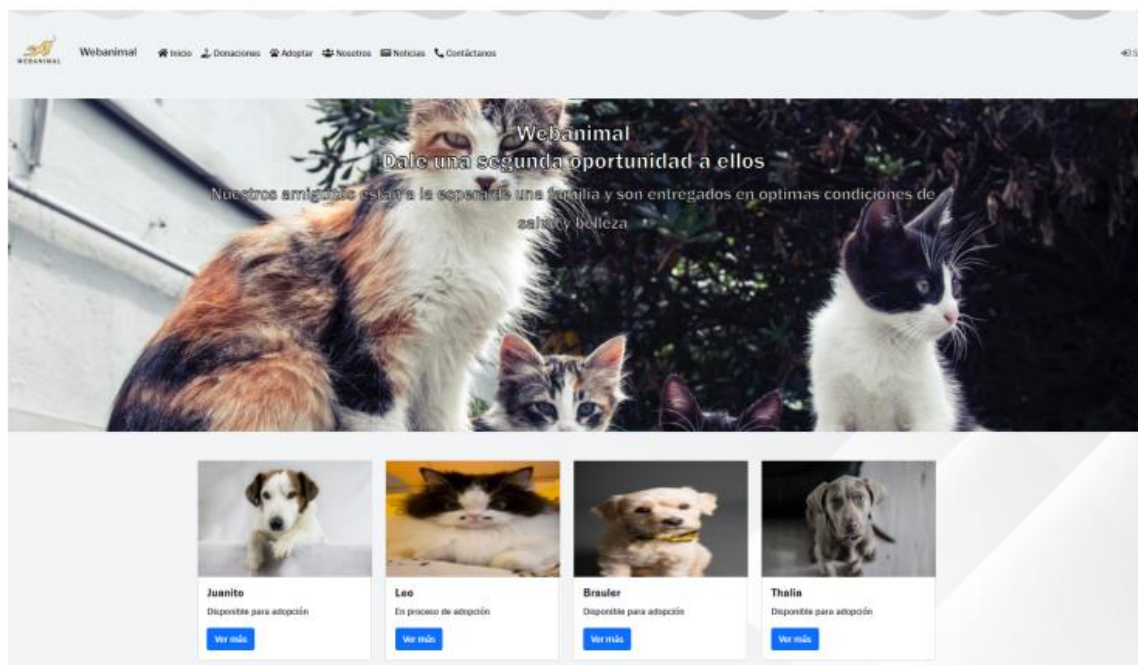
## Pruebas de caja negra

La pruebas de caja negra es una técnica de pruebas de software en la cual la funcionalidad se verifica sin tomar en cuenta la estructura interna de código, detalles de implementación o escenarios de ejecución internos en el software.

### Prueba 1

En esta ocasión se va a poner a prueba el Login del sistema de información, para validar que cuando el usuario ingrese la credenciales lo redirija a la pagina de adopciones.

The screenshot shows the Webanimal website's login interface. At the top, a navigation bar includes the Webanimal logo, a home icon, and links for 'Inicio', 'Donaciones', 'Adoptar', 'Nosotros', 'Noticias', and 'Contactanos'. On the right side of the bar are links for 'Iniciar Sesión' and 'Registrarse'. The main content area features a central login modal titled 'Inicio de Sesión'. This modal contains a 'Username' field with the text 'admin', a 'Contraseña' field with masked characters, a blue 'Ingresar' button, and a link '¿No tienes cuenta? Crear una'. Below the login form, the footer is divided into three columns: 'Adopta' with placeholder text, 'Noticias' with placeholder text, and 'Redes sociales' with icons for Facebook, Instagram, and Twitter.



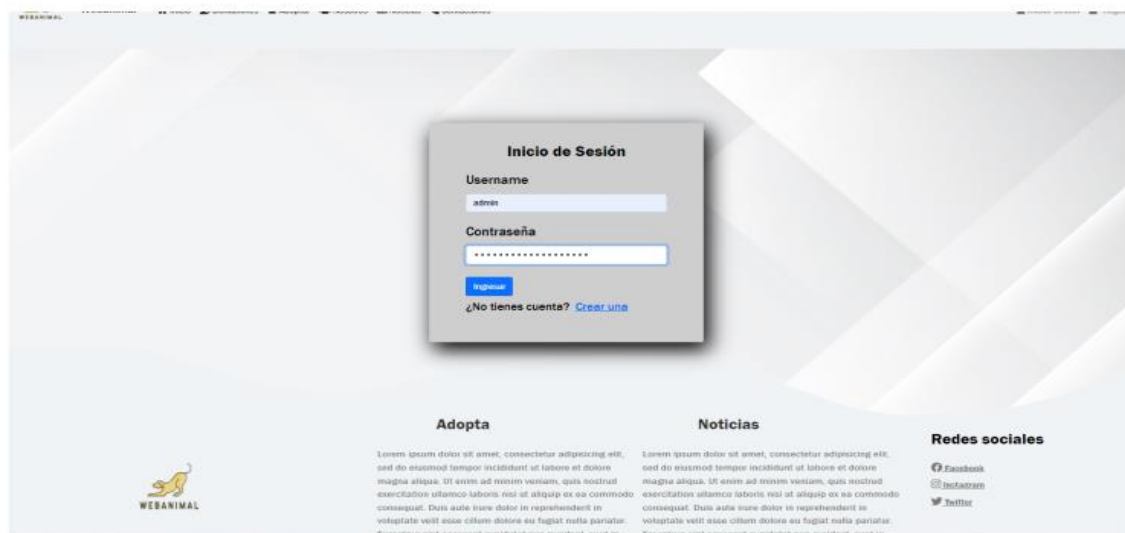
Podemos observar que el sistema reconoce al usuario ingresado y nos muestra un mensaje de confirmación antes de redirigirnos a la página de donaciones.

## Prueba 2

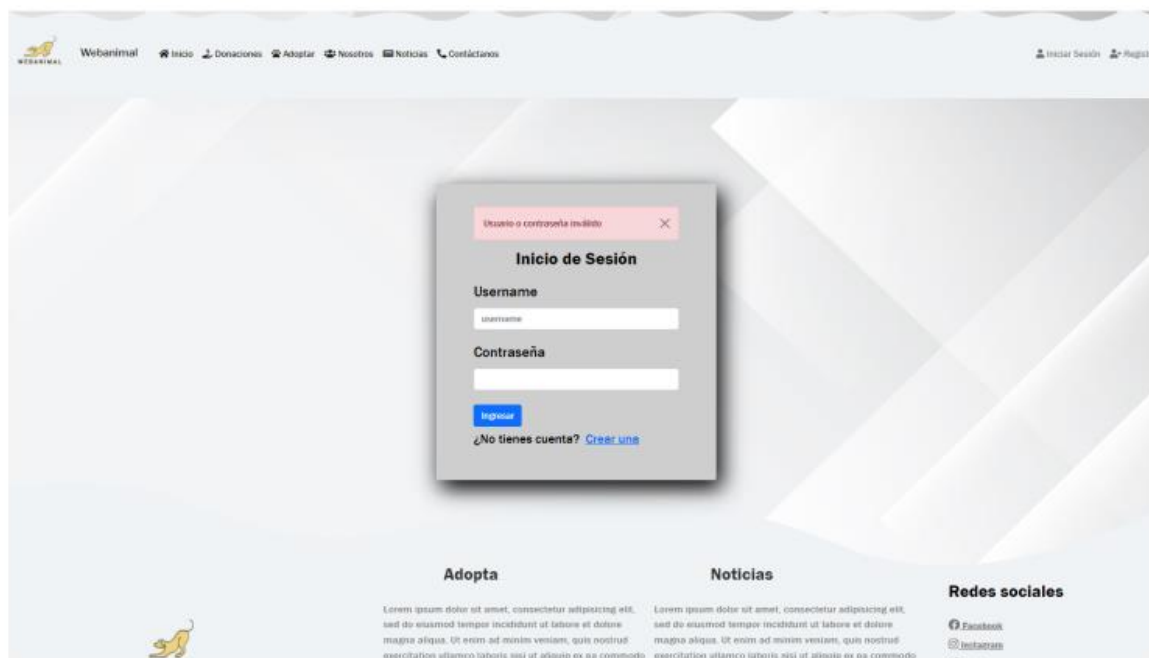
En esta prueba, se validará que pasa en caso de que se quiera iniciar sesión sin llenar los campos correspondientes.

### Prueba 3

En este caso probaremos que pasa en caso de que el usuario ingrese las credenciales de manera errónea.



Podemos observar que al dar clic en acceder nos llevara de nuevo a iniciar sesión con un mensaje emergente.



## Pruebas de caja blanca

### Pruebas de caja blanca

Estas pruebas tienen como objetivo probar la funcionalidad interna del sistema. Haremos las pruebas con el login del sistema.

#### Prueba 1

En este apartado no se realizó ninguna modificación al código

```
1 def login_view(request):
2     if request.method == "POST":
3         email = request.POST.get('email')
4         password = request.POST.get('password')
5         email = authenticate(username=email, password=password)
6         if email:
7             login(request, email)
8             messages.success(request, 'Bienvenido {}'.format(email.username))
9             return redirect('Donaciones') # Nombre url
10        else:
11            messages.error(request, 'Usuario o contraseña incorrecta')
12        return render(request, 'InicioSesion.html',{
13    })
```

```
1 path('login/', views.login_view, name='login'),
```

Al correr el servidor, en el ambiente virtual activado, no presento ningún error y podemos ver los formularios correctamente.

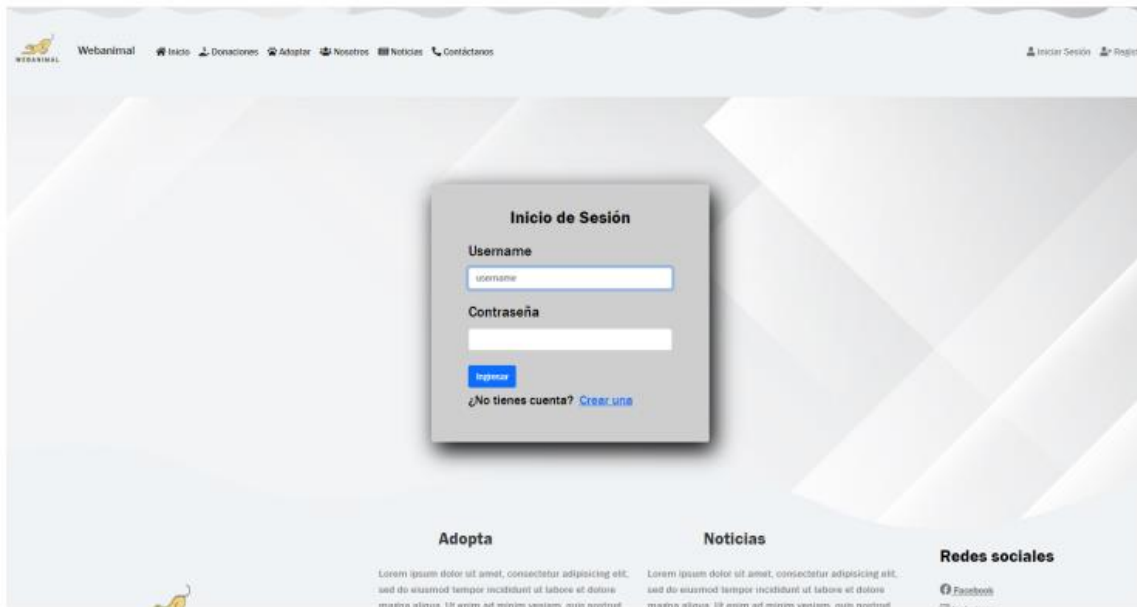


```

(env) PS C:\Users\Janus\Documents\GitHub\proyecto-aplicativo-webanimal\Proyecto\webanimal2\webanimal2> py manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
September 26, 2021 - 12:26:48
Django version 3.2.6, using settings 'webanimal2.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.

```



## Prueba 2

En el apartado de views.py, se borrara la línea para autenticar el usuario al ingresar las credenciales.

```

1 def login_view(request):
2     if request.method == "POST":
3         email = request.POST.get('email')
4         password = request.POST.get('password')
5
6         if email:
7             login(request, email)
8             messages.success(request, 'Bienvenido {}'.format(email.username))
9             return redirect('Donaciones') # Nombre url
10        else:
11            messages.error(request, 'Usuario o contraseña incorrecta')
12    return render(request, 'InicioSesion.html', {
13    })

```

Se corrió el servidor para testear si funciona correctamente y pudimos ver que nos presentó el siguiente error en consola:





## Prueba Pytest

### Pytest

Pytest proporciona documentación completa en línea, una gran cantidad de complementos de terceros y ayuda integrada, desde pruebas pequeñas hasta pruebas de gran escala como pruebas funcionales de aplicaciones y librerías.

#### Prueba 1

En esta primera prueba, se ingresaran datos a la tabla Pet

```
1 @pytest.mark.django_db
2 def test_pet():
3     pet = Pet.objects.create(
4         name = 'Mini',
5         pet_age = '2',
6         breed = 'Criollo',
7         description = 'Con manchas',
8     )
9     assert pet.name == 'Mini'
```

Al ejecuta la prueba mostrara el siguiente mensaje para confirmar que la prueba se ejecuto correctamente.

```
C:\Users\Janus\Documents\Github\projecto-aplicativo-webanimal\Projecto\webanimal2\webanimal2 (master -> origin)
(env) A pytest
===== test session starts =====
platform win32 -- Python 3.9.6, pytest-6.2.5, py-1.10.0, pluggy-1.0.0
django: settings: webanimal2.settings (from ini)
rootdir: C:\Users\Janus\Documents\Github\projecto-aplicativo-webanimal\Projecto\webanimal2\webanimal2, configfile: pytest.ini
plugins: django-4.4.0
collected 1 item

test\test_source.py .
```

#### Prueba 2

En esta ocasión se agregaran datos como la descripción de la tabla Donation

```
1 @pytest.mark.django_db
2 def test_donation():
```

```

3     donation = Donation.objects.create(
4         description = 'Espuma para camas',
5     )
6     assert donation.description == 'Espuma para camas'

```

En la consola nos debería mostrar el siguiente mensaje para confirmar que agrego el dato correctamente.

```

C:\Users\Janus\Documents\GitHub\proyecto-aplicativo-webanimal\Proyecto\webanimal2\webanimal2 (master -> origin)
(env) A pytest
===== test session starts =====
platform win32 -- Python 3.9.6, pytest-6.2.5, py-1.10.0, pluggy-1.0.0
django: settings: webanimal2.settings (from ini)
rootdir: C:\Users\Janus\Documents\GitHub\proyecto-aplicativo-webanimal\Proyecto\webanimal2\webanimal2, configfile: pytest.ini
plugins: django-4.4.0
collected 1 item

test\test_source.py .

```

### Pueba 3

Por ultimo, se realizara la prueba agregando datos a la tabla Adopción, como name, pet\_age, breed, user\_name y noDocument.

```

1 @pytest.mark.django_db
2 def test_adopcion():
3     adopcion = Adopcion.objects.create(
4         name = 'Mini',
5         pet_age = '2',
6         breed = 'Criollo',
7         user_name = 'Juan',
8         nuDocument = '77777777',
9     )
10    assert adopcion.name == 'Mini'

```

Página 12 de 13

Podemos ver en la siguiente imagen que se han agregando los datos correspondientes y la prueba fue ejecutada exitosamente.

```

C:\Users\Janus\Documents\GitHub\proyecto-aplicativo-webanimal\Proyecto\webanimal2\webanimal2 (master -> origin)
(env) A pytest
===== test session starts =====
platform win32 -- Python 3.9.6, pytest-6.2.5, py-1.10.0, pluggy-1.0.0
django: settings: webanimal2.settings (from ini)
rootdir: C:\Users\Janus\Documents\GitHub\proyecto-aplicativo-webanimal\Proyecto\webanimal2\webanimal2, configfile: pytest.ini
plugins: django-4.4.0
collected 1 item

test\test_source.py .

```

<https://drive.google.com/file/d/1Qx4kiRVdoGZgdX4QFoMxBex-85tBTWxX/view>

**Flujo de trabajo**