

**WEBANIMAL**

# **Pruebas de software**

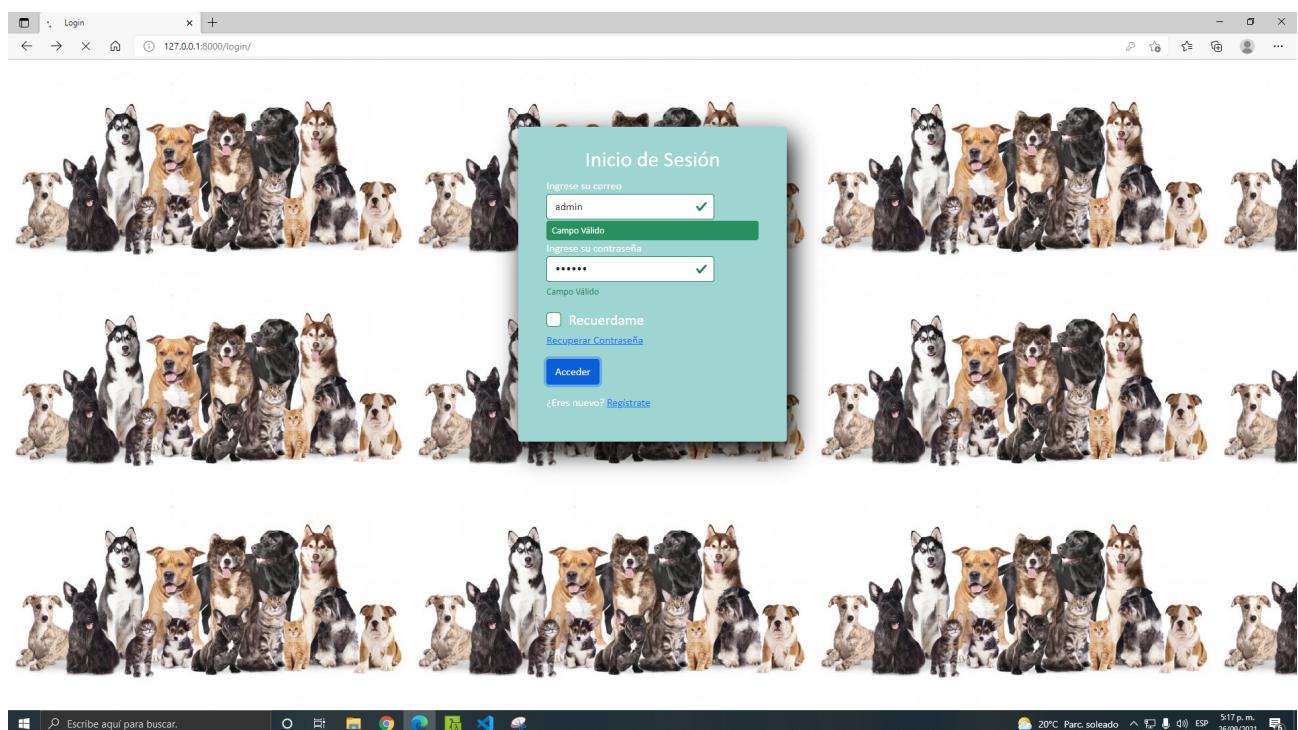
Son un conjunto de procesos con los que se pretende probar un sistema o aplicación en diferentes momentos para comprobar su correcto funcionamiento. Este tipo de pruebas abarca cualquier estadio del desarrollo del sistema, desde su creación hasta su puesta en producción.

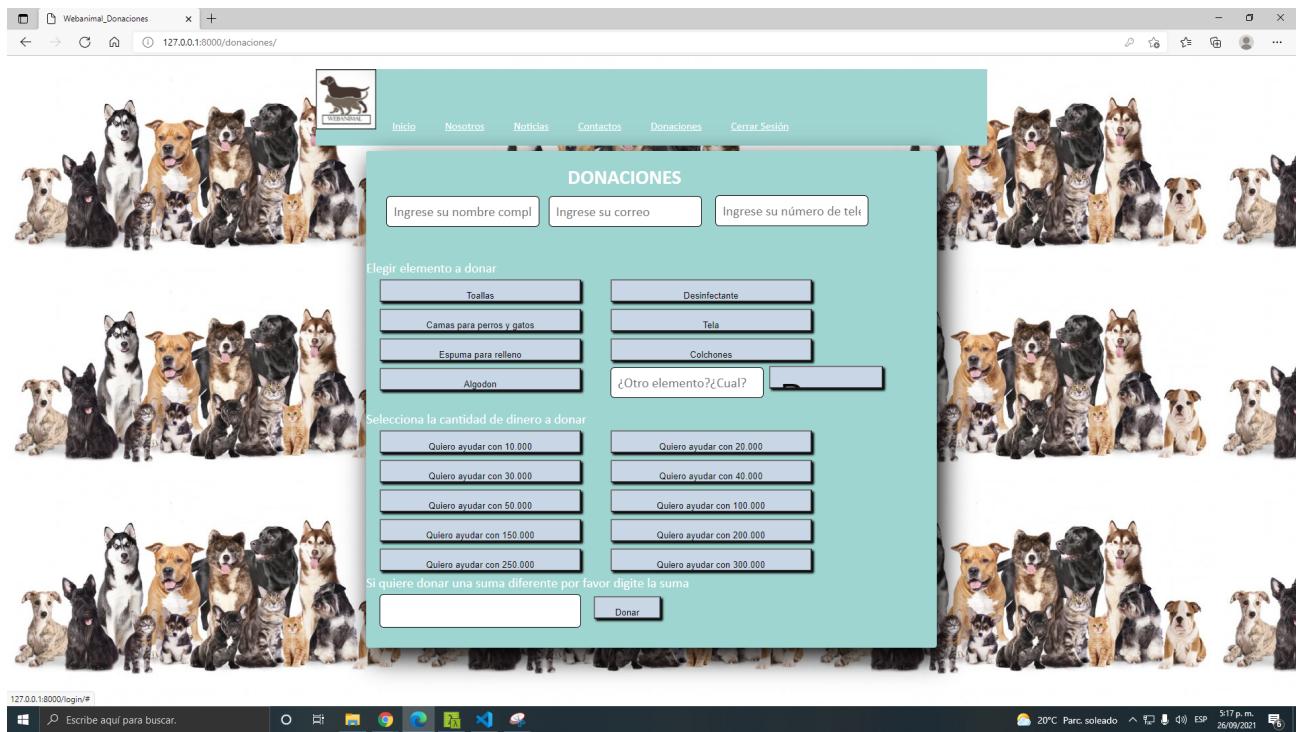
# Pruebas de caja negra

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 donaciones.

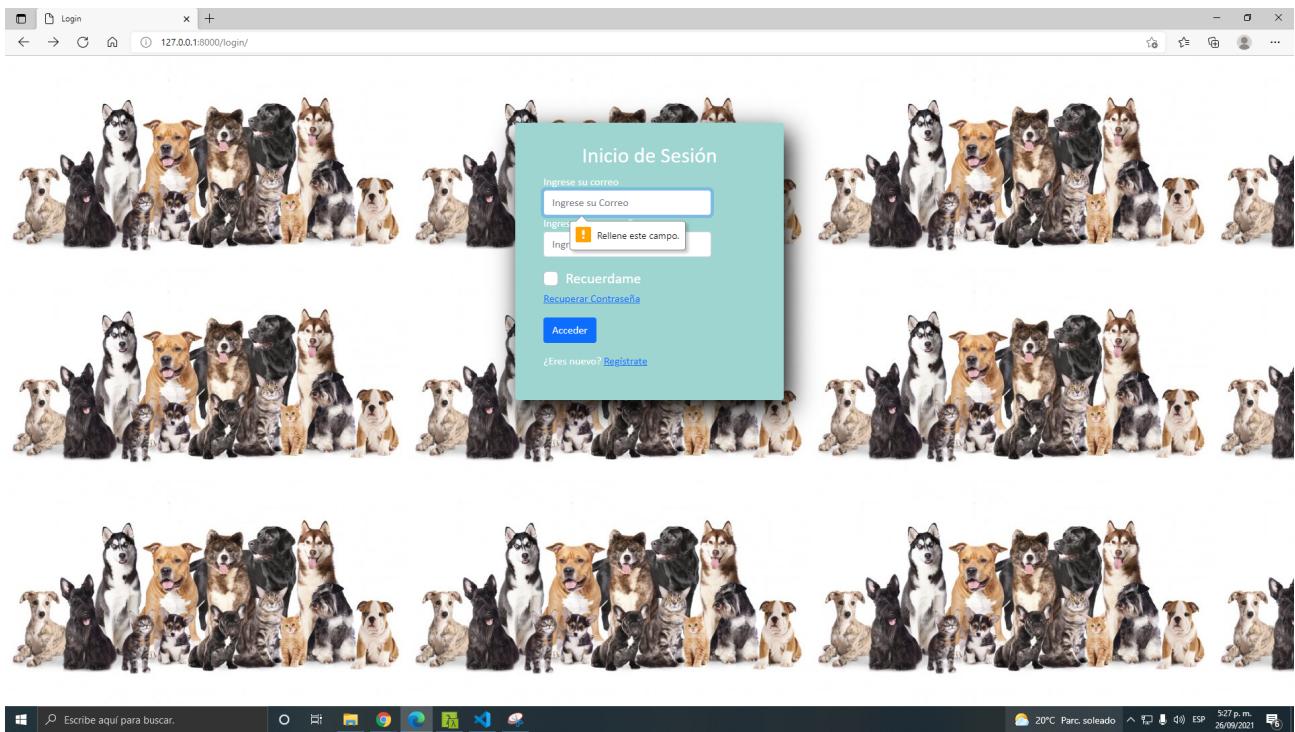




Podemos observar que el sistema reconoce al usuario ingresado y nos muestra un mensaje de confirmación antes de redirijirnos a la pagina de donaciones.

## Prueba 2

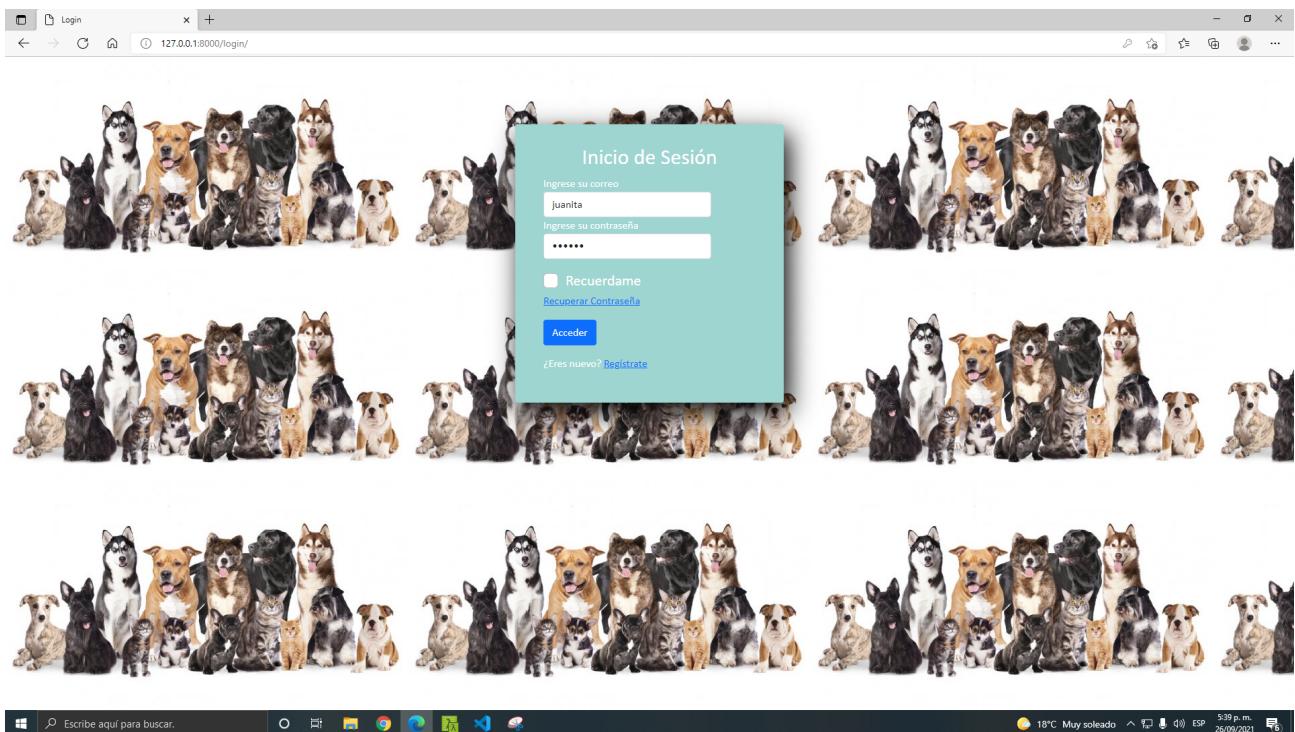
En esta prueba, se validara que pasa en caso de que se quiera iniciar sesión sin llenar los campos correspondientes.



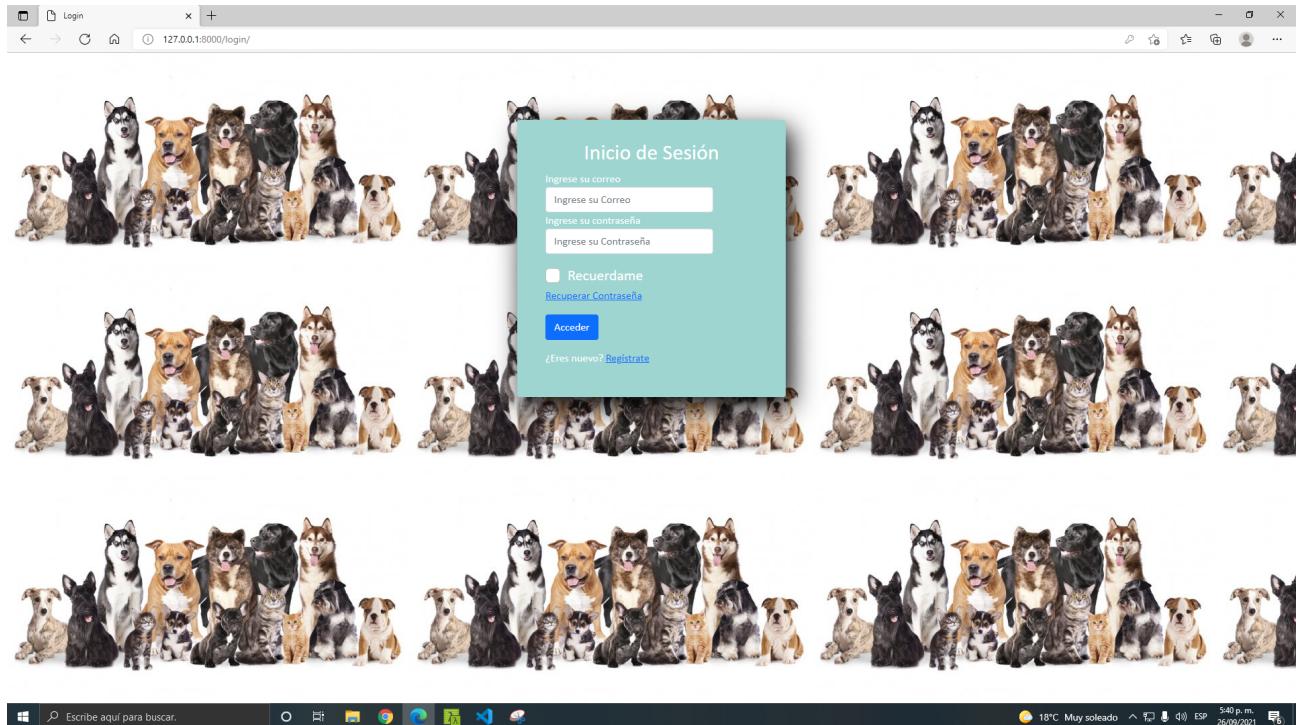
Podemos ver que nos aparece un cuadro emergente, diciendo que hay que llenar el campo.

## Prueba 3

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



Podemos observar que al dar clic en acceder nos lleva de nuevo a iniciar sesión con los campos vacíos indicandonos que las credenciales estan mal.



# 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 realizo ninguna modificación al código

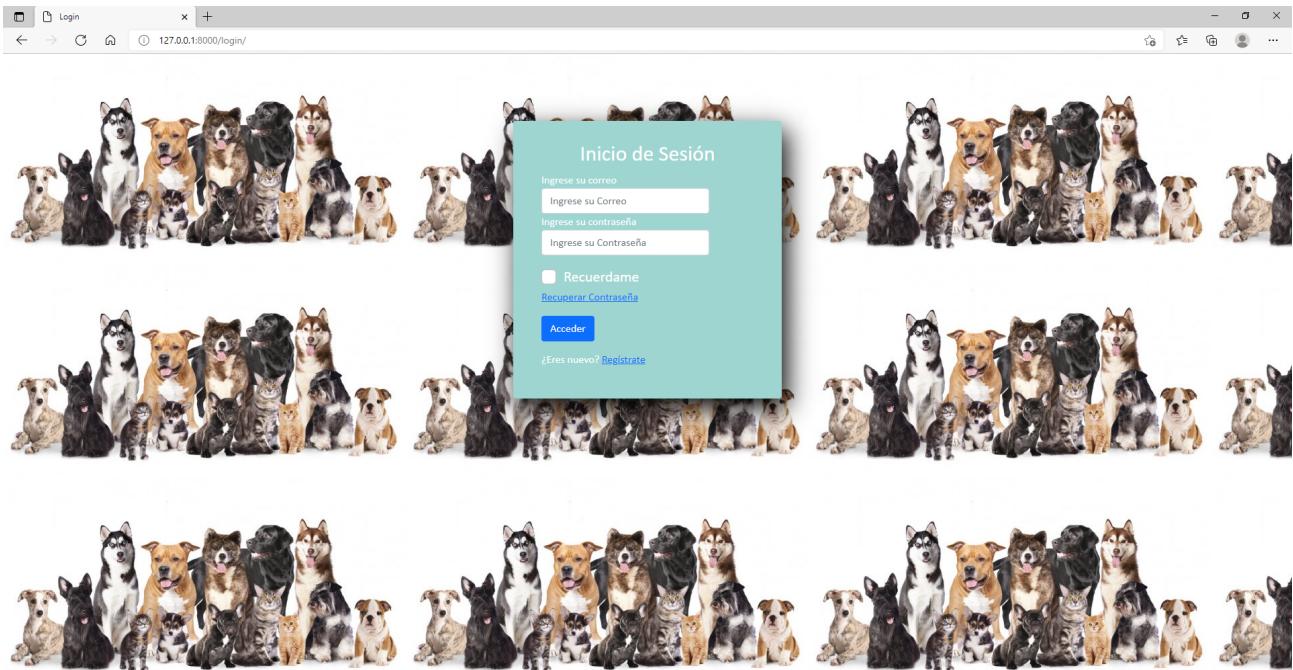
```
1 def login_view(request):
2     if request.method == "POST":
3         email = request.POST.get('email')
4         password = request.POST.get('password')
5         email = authenticate(username=email, password=password)
6         if email:
7             login(request, email)
8             messages.success(request, '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    })
```

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

Al correr el servidor, en el ambiente virtual activado, no presento ningun 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 linea para autenticar el usuario al ingresar las credenciales.

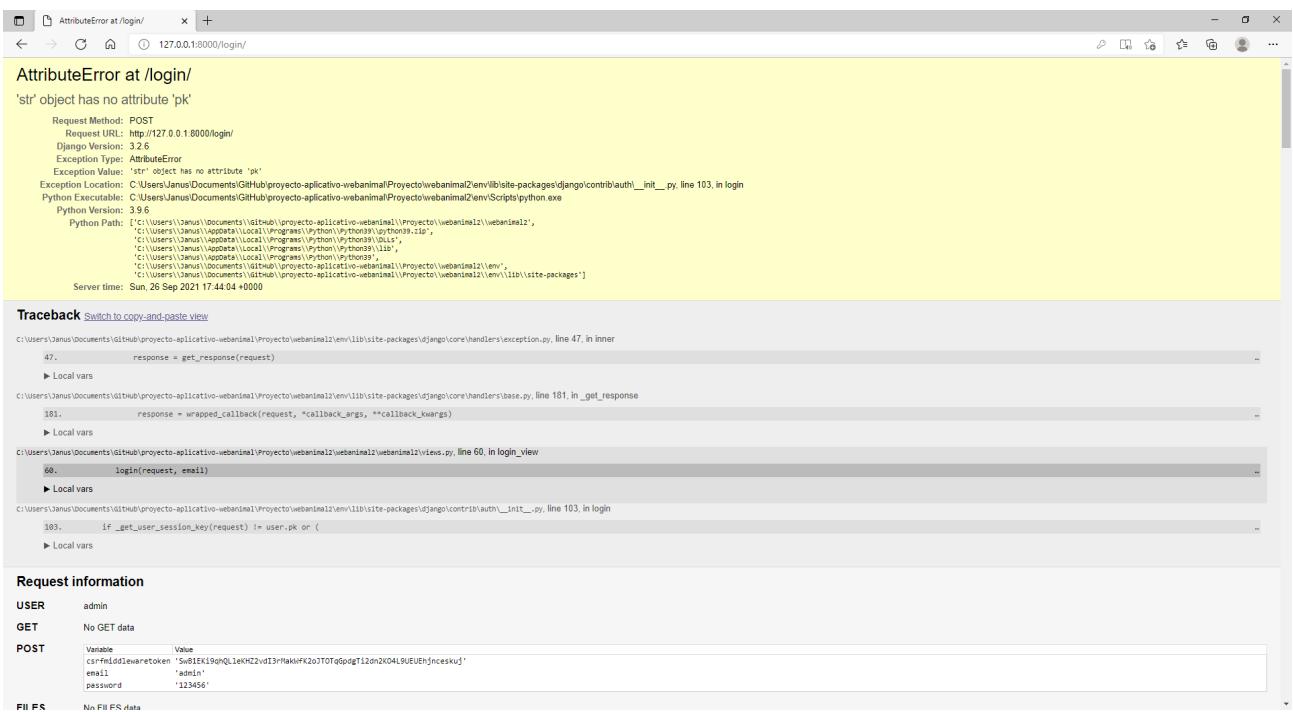
```
1 def login_view(request):
2     if request.method == "POST":
3         email = request.POST.get('email')
4         password = request.POST.get('password')
5
6         if email:
7             login(request, email)
8             messages.success(request, 'Bienvenido {}'.format(email.username))
9 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

Se corrio el servidor para testear si funciona correctamente y pudimos ver que nos presento el siguiente error, en consola.

```
Windows PowerShell
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
C:\Users\Janus\Documents\GitHub\proyecto-aplicativo-webanimal\Proyecto\webanimal2\webanimal2\views.py changed
, reloading.
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
September 26, 2021 - 13:18:50
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.
[26/Sep/2021 13:18:52] "GET /login/ HTTP/1.1" 200 3793
Internal Server Error: /login/
Traceback (most recent call last):
  File "C:\Users\Janus\Documents\GitHub\proyecto-aplicativo-webanimal\Proyecto\webanimal2\env\lib\site-packages\django\core\handlers\exception.py", line 47, in inner
    response = get_response(request)
  File "C:\Users\Janus\Documents\GitHub\proyecto-aplicativo-webanimal\Proyecto\webanimal2\env\lib\site-packages\django\core\handlers\base.py", line 181, in _get_response
    response = wrapped_callback(request, *callback_args, **callback_kwargs)
  File "C:\Users\Janus\Documents\GitHub\proyecto-aplicativo-webanimal\Proyecto\webanimal2\webanimal2\views.py", line 60, in login_view
    login(request, email)
  File "C:\Users\Janus\Documents\GitHub\proyecto-aplicativo-webanimal\Proyecto\webanimal2\env\lib\site-packages\django\contrib\auth\__init__.py", line 103, in login
    if _get_user_session_key(request) != user.pk or (
AttributeError: 'str' object has no attribute 'pk'
[26/Sep/2021 13:18:59] "POST /login/ HTTP/1.1" 500 70659
```

En este caso podemos ver que al ingresar las credencias para que el usuario pueda iniciar sesión, no se lo permitira ya que este aun no esta autenticado en el apartado de views.py.



## Prueba 3

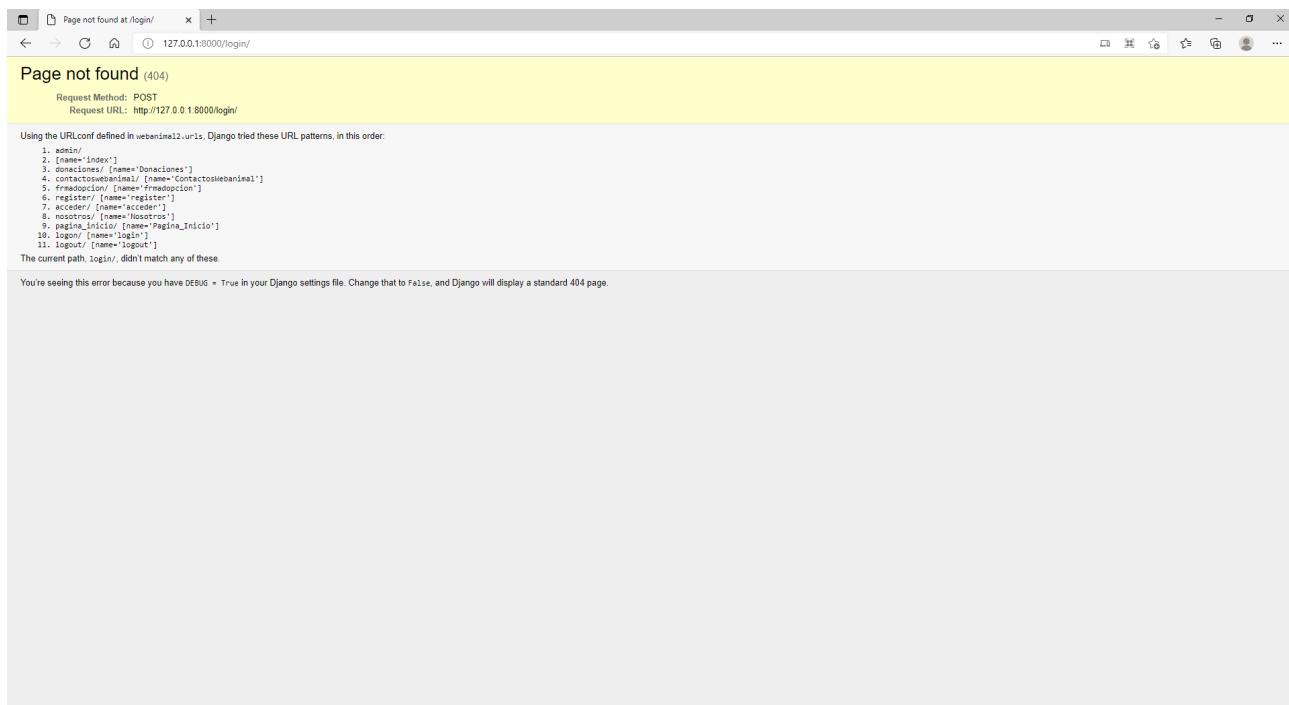
En el archivo urls.py se cambiara la ruta del login, poniendole otro nombre

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

Al correr el servidor nos mostrara el siguiente error.

```
System check identified no issues (0 silenced).
September 26, 2021 - 13:25:39
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.
Not Found: /login/
[26/Sep/2021 13:25:41] "POST /login/ HTTP/1.1" 404 3376
```

Podemos encontrar que al tener la ruta incorrecta como lo tenemos definido en el apartado de urls.py, no nos mostrara los formularios, y nos mostrara los apartados de las urls para verificar que si este correctamente.



# 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 ingresarán datos a la tabla Pet

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

Al ejecutar la prueba mostrará el siguiente mensaje para confirmar que la prueba se ejecutó correctamente.

```
C:\Users\Janus\Documents\GitHub\proyecto-aplicativo-webanimal\Proyecto\webanimal2\webanimal2 (master -> origin)
(env) λ pytest
=====
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 .
```

## Prueba 2

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

```
1 @pytest.mark.djangoproject
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 deberia 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) λ pytest
=====
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.djangoproject
2 def test_adopcion():
3     adopcion = Adopcion.objects.create(
4         name = 'Mini',
5         pet_age = '2',
6         breed = 'Criollo',
7         user_name = 'Juan',
8         noDocument = '7777777777',
9     )
10    assert adopcion.name == 'Mini'
```

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) λ pytest
=====
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 .
```