

Entrega 6 (linter)

La herramienta utilizada fue **Ruff**, un linter y formateador para Python diseñado para ser rápido y compatible con proyectos modernos como Django. Permite analizar el código, detectar errores comunes, advertencias, malas prácticas, importes no utilizados y problemas de estilo.

⚙️ Configuración aplicada

```
1 [tool.ruff]
2 target-version = "py313"
3 line-length = 100
4 extend-include = ["*.py", "*.pyi"]
5
6 [tool.ruff.lint]
7 select = ["E", "F", "W", "B", "I", "UP", "DJ"]
8 ignore = ["DJ001", "DJ008"]
9 fixable = ["ALL"]
10
11 [tool.ruff.lint.per-file-ignores]
12 */migrations/*.py" = ["ALL"]
13
14 [tool.ruff.format]
15 quote-style = "double"
16 indent-style = "space"
17 docstring-code-format = true
```

Se configuró Ruff dentro del archivo `pyproject.toml` con los siguientes ajustes principales:

-  **Versión objetivo:** `py313`
-  **Longitud máxima de línea:** 100 caracteres
-  **Archivos incluidos:** `*.py, *.pyi`

- **Reglas activadas:** E, F, W, B, I, UP, DJ
- **Reglas ignoradas:** DJ001, DJ008
- **Correcciones automáticas habilitadas:** ALL
- **Exclusiones por archivo:** todo en migrations/ ignora todas las reglas
- **Formato aplicado:**
 - Estilo de comillas → "double"
 - Indentación → espacios
 - Formateo de docstrings → habilitado

Estas configuraciones permiten que Ruff actue tanto como verificador de errores como formateador automático.



Resultados obtenidos (errores o advertencias relevantes)

Los principales problemas detectados fueron:

- **Imports no utilizados (F401)** Ruff detectó múltiples imports innecesarios, por ejemplo:
 - from django.test import TestCase
 - from django.shortcuts import render
 - from django.contrib import admin
 - from django.db import models
- Esto ocurre cuando se importan módulos que nunca se usan en el archivo.
- **Imports desorganizados** Ruff reordenó los imports según buenas prácticas:
 - Orden alfabético
 - Separación entre estándar, terceros y locales
- **Problemas menores de estilo** Incluyendo espacios innecesarios, saltos de línea incorrectos y desalineaciones.

```
F401 [*] `django.test.TestCase` imported but unused
--> costs\tests.py:1:25
|
1 | from django.test import TestCase
|          ^^^^^^^^
2 |
3 | # Create your tests here.
|
help: Remove unused import: `django.test.TestCase`


F401 [*] `django.shortcuts.render` imported but unused
--> costs\views.py:1:30
|
1 | from django.shortcuts import render
|          ^^^^^^^^
2 |
3 | # Create your views here.
|
help: Remove unused import: `django.shortcuts.render`


F401 [*] `django.contrib.admin` imported but unused
--> dashboard\admin.py:1:28
|
1 | from django.contrib import admin
|          ^^^^^^
2 |
3 | # Register your models here.
|
help: Remove unused import: `django.contrib.admin`


F401 [*] `django.db.models` imported but unused
--> dashboard\models.py:1:23
|
1 | from django.db import models
|          ^^^^^^^^
2 |
3 | # Create your models here.
|
help: Remove unused import: `django.db.models`


F401 [*] `django.test.TestCase` imported but unused
--> dashboard\tests.py:1:25
```

```
help: Remove unused import: `django.test.TestCase`  
  
W292 [*] No newline at end of file  
--> dashboard\urls.py:9:2  
|  
7 |     urlpatterns = [  
8 |         path("", DashboardHomeView.as_view(), name="home"),  
9 |     ]  
|  
  
help: Add trailing newline  
  
I001 [*] Import block is un-sorted or un-formatted  
--> dashboard\views.py:1:1  
|  
1 | / from django.contrib.auth.mixins import LoginRequiredMixin  
2 | | from django.views.generic import TemplateView  
| |  
3 | |  
4 | class DashboardHomeView(LoginRequiredMixin, TemplateView):  
|  
help: Organize imports  
  
F401 [*] `django.contrib.admin` imported but unused  
--> tracking\admin.py:1:28  
|  
1 | from django.contrib import admin  
| ^^^^^^  
2 |  
3 | # Register your models here.  
|  
help: Remove unused import: `django.contrib.admin`  
  
F401 [*] `django.db.models` imported but unused  
--> tracking\models.py:1:23  
|  
1 | from django.db import models  
| ^^^^^^  
2 |  
3 | # Create your models here.  
|  
help: Remove unused import: `django.db.models`
```

Evidencia de su ejecución

Durante la ejecución, Ruff mostró advertencias detalladas indicando:

- Archivo afectado
 - Línea del error
 - Tipo de problema (`imported but unused`)
 - Sugerencia automática ✨
-

```
Found 139 errors.  
[*] 135 fixable with the `--fix` option.  
● C:\Users\ADMINTN\Documents\GitHub\PTOS
```

🟡 Evidencia de su corrección

Tras ejecutar `ruff --fix`:

- Se eliminaron todos los imports no utilizados ✓
- Se organizaron correctamente los bloques de imports ✓
- El estilo quedó formateado automáticamente ✓

```
\ proyecto> ruff check . --fix
```

El reporte final ya no presenta advertencias relevantes.

```
Found 140 errors (136 fixed, 4 remaining).  
● C:\Users\ADMINTN\Documents\GitHub\PTOS-T
```