

# Buenas prácticas de desarrollo en Python



Erick Castillo B.  
@ecastillob | 1p.cl | ecastillo

Noviembre - 2021

# Indice

## 1 Intro

- Intro
- Alternativas

## 2 Gestión de dependencias

- Entornos

## 3 Formateadores

- Formateadores
- Black
- isort

## 4 Linters

- Linters
- flake8
- pylint

## 5 Aspectos de seguridad

- Intro
- safety

## Contexto



## Situaciones

- Tengo mi propia identación
  - Prefiero usar comillas simples a dobles
  - Me da lo mismo el orden de importar los módulos
  - Resuelvo conflicto de dependencias con pip “a mano”
  - Uso paquetes útiles, pero no me aseguro de verificar si tienen problemas de seguridad



## Alternativas a problemática



## Alternativas a problemática (cont.)



## Alternativas a problemática (cont.)



# Alternativas a problemática (cont.)

(Loading Video...)

# Alternativas a problemática (cont.)



# Gestión de dependencias

# Python, versiones, paquetes y gestores

Versiones de Python	Gestor de paquetes
pyenv	*pip
	pipx
	pipenv
	poetry
	conda

# Gestores de paquetes - versiones

	<b>pip</b>	<b>pipx</b>	<b>pipenv</b>	<b>poetry</b>	<b>conda</b>
primera versión	28-10-2008	27-07-2017	20-01-2017	28-02-2018	06-09-2012
versión >=1.0	04-04-2011	-	24-01-2017	12-12-2019	06-09-2012
versión actual	21.3 (22-10-2021)	0.16.4 (05-08-2021)	11.10.4 (08-05-2018)  2021.5.29 (29-05-2021)	1.1.11 (04-10-2021)	4.10.3 (30-06-2021)

# Gestores de paquetes - características *out-of-the-box*

	<b>pip</b>	<b>pipx</b>	<b>pipenv</b>	<b>poetry</b>	<b>conda</b>
inicializar	-	-	-	init	-
instalar	install	install	add + install	install	
desinstalar	uninstall	-	uninstall	remove	remove
listar	freeze	?	-	-	list
información	show	?	show	-	
verificación de conflictos	check	o	o	o	
verificación de seguridad	-	-	check	-	-
entorno virtual	-	-	shell	shell	create
generación	-	-	-	build	-
publicación	-	-	-	publish	-
separación de ambientes dev/no-dev	-	?	o	o	-
carga auto de .env	-	-	o	-	-
archivo de dependencias	requirements.txt	requirements.txt	Pipfile	pyproject.toml	requirements.txt

# Gestores de paquetes - Poetry vs Pipenv

1 2 3 4	poetry	pipenv	pip
resolución de conflictos de dependencias	muy buena	buenas	pobre
velocidad de instalación de nuevos paquetes	lenta	muy lenta	rápida
velocidad de instalación de todos los paquetes	muy lenta	lenta	rápida
bloqueo de dependencias	buenas	buenas	nula
mantenibilidad	buenas	cuestionable <sup>5</sup>	buenas

<sup>1</sup> <https://johnfraney.ca/posts/2019/03/06/pipenv-poetry-benchmarks-ergonomics/>

<sup>2</sup> <https://johnfraney.ca/posts/2019/11/19/pipenv-poetry-benchmarks-ergonomics-2/>

<sup>3</sup> <https://pythonspeed.com/articles/pipenv-docker/>

<sup>4</sup> <https://dev.to/frostming/a-review-pipenv-vs-poetry-vs-pdm-39b4>

<sup>5</sup> no tuvo cambios por año y medio, entre 2018 y 2019

# Formateadores

# Formateadores

- Paquetes de Python
- Formatea el código fuente en base a una guía de estilo (como la PEP 8)
- Configurable
- Ejemplos:
  - **Black**
  - **isort**
  - **autopep8**
  - **YAPF**

# black

- sitios web:
  - paquete en PyPI: <https://pypi.org/project/black/>
  - repositorio en Github: <https://github.com/psf/black>
  - documentación: <https://black.readthedocs.io/en/stable/index.html>
- *fun fact*: está en beta
- configuración:
  - por defecto usa un archivo llamado `pyproject.toml`<sup>6</sup>, si no lo encuentra en directorio actual, sigue buscando directorios hacia arriba hasta detectar un repositorio
  - si no encuentra el archivo busca el archivo `.black` ubicado típicamente en la carpeta del usuario
  - es posible utilizarlo sin poseer ambos archivos de configuración
- quién lo utiliza:
  - paquetes: pytest, Pyramid, SQLAlchemy, Poetry, pandas, Pillow
  - organizaciones / empresas: Facebook, Dropbox, Mozilla, Quora, Duolingo

<sup>6</sup>Introducido primeramente en la PEP 518 y expandido en la PEP PEP 517, PEP 621 y PEP 660

black - ejemplo 1: sin archivo de config (antes de aplicar)

```
def my_function( first_names,last_names):
    names = []
    for first_name,last_name in zip(first_names,last_names):
        name = f"{first_name} {last_name}"
        names.append(name)
    return names

def run():
    first_names = ["Gon", "Killua", "Leorio", "Zeno", "Alluka", "Mito", "Ging"]
    last_names = ["Freecss", "Zoldyck", "Paladiknight", "Zoldyck", "Zoldyck", "Freecss", "Freecss"]
    names = my_function( first_names,last_names)
    print(names)

if __name__ == "__main__":
    run()
```

# black - ejemplo 1: sin archivo de config (después de aplicar)

```
● ● ●

def my_function(first_names, last_names):
    names = []
    for first_name, last_name in zip(first_names, last_names):
        name = f"{first_name} {last_name}"
        names.append(name)
    return names


def run():
    first_names = ["Gon", "Killua", "Leorio", "Zeno", "Alluka", "Mito", "Ging"]
    last_names = [
        "Freecss",
        "Zoldyck",
        "Paladiknight",
        "Zoldyck",
        "Freecss",
        "Freecss",
    ]
    names = my_function(first_names, last_names)
    print(names)


if __name__ == "__main__":
    run()
```

# black - ejemplo 2: con archivo de config (antes de aplicar)

```
● ● ●

import math
from random import choice as choice_random
import datetime

def my_function( fields):
    last_field = fields[-1]
    counter=0
    while True:
        counter += 1
        field = choice_random(fields)
        if field==last_field:
            break
    return counter

def run():
    fields = ["best_brand", "best_model", "min_year", "max_year", "budget_min", "budget_max"]
    now = datetime.datetime.now()
    n_attempts = my_function(fields)
    total_time = str(datetime.datetime.now() - now)
    print(f"{n_attempts} in {total_time}", "\t", f"log_2 of {n_attempts} is ", math.log2(n_attempts))

if __name__ == "__main__":
    run()
```

# black - ejemplo 2: con archivo de config

archivo de configuración `pyproject.toml`

```
● ● ●

[project]
name = "Examples for black formatter"
version = "0.1.0"
description = "A simple description"
authors = [
    "Erick Castillo <myemail@mydomain.cl>"
]
readme = "README.md"
python = "^3.8"

[tool.black]
line-length = 120
```

# black - ejemplo 2: con archivo de config (después de aplicar)

```
● ● ●

import math
from random import choice as choice_random
import datetime

def my_function(fields):
    last_field = fields[-1]
    counter = 0
    while True:
        counter += 1
        field = choice_random(fields)
        if field == last_field:
            break
    return counter

def run():
    fields = ["best_brand", "best_model", "min_year", "max_year", "budget_min", "budget_max"]
    now = datetime.datetime.now()
    n_attempts = my_function(fields)
    total_time = str(datetime.datetime.now() - now)
    print(f"{n_attempts} in {total_time}", "\t", f"log_2 of {n_attempts} is ", math.log2(n_attempts))

if __name__ == "__main__":
    run()
```

# isort

- sitios web:
  - paquete en PyPI: <https://pypi.org/project/isort/>
  - repositorio en github: <https://github.com/pycqa/isort>
  - documentación: <https://pycqa.github.io/isort/>
- Mantenido por PyCQA<sup>7</sup> (Python Code Quality Authority)
- Última versión estable: 5.10.0 (3-Nov-2021)

# isort - ejemplo 1: sin archivo de config (antes de aplicar)

```
● ○ ●

import math
from random import randint
import datetime

def run():
    now = datetime.datetime.now()
    number = randint(1000, 9999)
    total_time = str(datetime.datetime.now() - now)
    log2_number = math.log2(number)
    print(f"log_2 of {number} is {log2_number} calculated in {total_time}")

if __name__ == "__main__":
    run()
```

# isort - ejemplo 1: sin archivo de config (después de aplicar)

```
● ○ ●

import datetime
import math
from random import randint

def run():
    now = datetime.datetime.now()
    number = randint(1000, 9999)
    total_time = str(datetime.datetime.now() - now)
    log2_number = math.log2(number)
    print(f"log_2 of {number} is {log2_number} calculated in {total_time}")

if __name__ == "__main__":
    run()
```

## isort - ejemplo 2: archivo de config



```
[project]
name = "Examples for show use of black"
version = "0.1.0"
description = "A simple description"
authors = [
    "Erick Castillo <mymail@mydomain.cl>"
]
readme = "README.md"
python = "^3.8"

[tool.black]
line-length = 120
```

pyproject.toml



```
[settings]
profile = black
multi_line_output = 3
include_trailing_comma = True
force_grid_wrap = 0
combine_as_imports = True
line_length = 80
```

.isort.cfg

## isort - ejemplo 2: con archivo de config (antes de aplicar)

```
● ● ●  
  
from math import cos, acos, sin, asin, tan, atan, log2, log10, sqrt, trunc, factorial  
  
# ...
```

# isort - ejemplo 2: con archivo de config (después de aplicar)

```
● ● ●  
  
from math import (  
    acos,  
    asin,  
    atan,  
    cos,  
    factorial,  
    log2,  
    log10,  
    sin,  
    sqrt,  
    tan,  
    trunc,  
)  
  
# ...
```

# Linters

# Linters

- análisis estático del código fuente (no realizan modificaciones)
- generan alertas:
  - errores
  - advertencias
  - sugerencias de mejora
- configurables

# Linters (Cont.)

¿ Y qué tiene que ver esto con el programa número uno de la televisión humorística ?



# Linters (Cont.)

- pycodestyle
- pyflakes
- flake8
- pylint
- pylama

# flake8

- sitios web:
  - PyPI: <https://pypi.org/project/flake8/>
  - Github: <https://github.com/pycqa/flake8>
  - documentación: <https://flake8.pycqa.org/en/latest/>
- instalación:
  - pip install flake8
- ejecución:
  - ① abrir la consola y ubicarse en la raíz de la carpeta que contiene el código fuente
  - ② escribir: flake8
- ignorar líneas:
  - de forma local
    - ignorar cualquier advertencia de una línea: # noqa flake8
    - ignorar advertencia específica: # noqa flake8: E231
  - de forma global
    - archivo de configuración llamado .flake8
    - archivo debe estar en la raíz de la carpeta del proyecto

# flake8 - ejemplo 1: sin archivo de config.

```
● ● ●

def my_function( first_names,last_names): # noqa: E201,E231
    names = []
    for first_name,last_name in zip(first_names,last_names): # noqa flake8
        name = f"{first_name} {last_name}"
        names.append(name)
    return names

def run():
    first_names = ["Gon", "Killua"]
    last_names = ["Freecss", "Zoldyck"]
    names = my_function( first_names,last_names )
    print(names)

if __name__ == "__main__":
    run()
```

# flake8 - ejemplo 1: sin archivo de config (cont.)

```
C:\Windows\System32\cmd.exe
(match) C:\PythonProjects\PyCON\flake8-example-1>flake8
.\main.py:12:25: E201 whitespace after '('
.\main.py:12:37: E231 missing whitespace after ','
```

# flake8 - ejemplo 2: con archivo de config.

```
● ● ●

[flake8]
max-line-length = 120
max-complexity = 15

# E203: whitespace before ':'
# W503: line break before binary operator
ignore = E203, W503

select = E,W
include = '\.pyi?$'
exclude = ''
/(
    \.git
    | build
    | dist
)/
...
```

# flake8 - plugins

- **mccabe:**

- mide la complejidad ciclomática
- instalación: pip install mccabe
- link al paquete en PyPI: <https://pypi.org/project/mccabe/>
- link al repo en Github: <https://github.com/pycqa/mccabe>

- compendio de plugins:

<https://github.com/DmytroLitvinov/awesome-flake8-extensions>

# flake8 - documentación de referencia

- parámetros en línea de comandos:  
<https://flake8.pycqa.org/en/latest/user/options.html>
- listado de errores y advertencias de flake8:  
<https://flake8.pycqa.org/en/latest/user/error-codes.html>
- listado de errores y advertencias de pycodestyle:  
<https://pycodestyle.pycqa.org/en/latest/intro.html#error-codes>

# pylint

- sitios web:
  - PyPI: <https://pypi.org/project/pylint/>
  - Github: <https://github.com/PyCQA/pylint/>
  - documentación: <https://pylint.pycqa.org/en/latest/>
- instalación:
  - pip install pylint
- ejecución:
  - ① abrir la consola y ubicarse en la raíz de la carpeta que contiene el código fuente
  - ② escribir: `pylint file_or_folder`
- ignorar líneas:
  - de forma local
    - ignorar cualquier advertencia de una línea: `# noqa pylint`
    - ignorar advertencia específica: `# noqa pylint: disable=arguments-differ`
  - de forma global
    - archivo de configuración llamado `.pylintrc`
    - archivo debe estar en la raíz de la carpeta del proyecto

# pylint - tipos de mensajes

Tipos<sup>8</sup>:

I nformational messages

R efactor

C onvention

W arning

E rror

F atal

Ejemplo:

```
filename.py:18:23: R1729: Use a generator instead
'any(value for value in range(1, 50 + 1) if is_prime(value))',
(use-a-generator)
```

# pylint vs flake8

- similitudes:

- poseen archivo de configuración
- pueden escanear directorios/subdirectorios de forma recursiva

- diferencias<sup>9</sup>:

pylint	flake8
actua a nivel de bloque (aunque puede actuar a nivel de sentencia)	actua a nivel de sentencia
genera métrica de calificación	no genera métrica
se debe ejecutar sobre archivos o carpetas	se puede ejecutar sobre directorio actual, carpeta o un archivo

<sup>9</sup>10.0 - ((float(5 \* error + warning + refactor + convention) / statement) \* 10)

# pylint - ejemplo 1: sin archivo de config

```
● ● ●

def is_prime(value):
    if value < 2:
        return False
    elif value % 2 == 0:
        return value == 2
    limit = value//2 + 1
    is_ok = True
    for current_value in range(3, limit, 2):
        if value % current_value == 0:
            is_ok = False
            break
    return is_ok


def run():
    any_prime_number = any([
        value
        for value in range(1, 50 + 1)
        if is_prime(value)
    ])
    return any_prime_number


if __name__ == "__main__":
    a = 123
    run()
```

# pylint - ejemplo 1: sin archivo de config (cont.)

```
C:\Windows\System32\cmd.exe

(match) C:\PythonProjects\PyCON\ pylint-example-1> pylint --fail-under=9.75 main.py
*****
Module main
main.py:1:0: C0114: Missing module docstring (missing-module-docstring)
main.py:1:0: C0116: Missing function or method docstring (missing-function-docstring)
main.py:2:4: R1705: Unnecessary "elif" after "return" (no-else-return)
main.py:15:0: C0116: Missing function or method docstring (missing-function-docstring)
main.py:16:23: R1729: Use a generator instead 'any(value for value in range(1, 50 + 1) if is_prime(value))'
(use-a-generator)
main.py:25:4: C0103: Constant name "a" doesn't conform to UPPER_CASE naming style (invalid-name)

-----
Your code has been rated at 6.67/10 (previous run: 6.67/10, +0.00)
```

# pylint - ejemplo 2: sin archivo de config

```
● ● ●

def is_prime(value):
    # noqa pylint: disable=no-else-return,missing-function-docstring
    if value < 2:
        return False
    elif value % 2 == 0:
        return value == 2
    limit = value//2 + 1
    is_ok = True
    for current_value in range(3, limit, 2):
        if value % current_value == 0:
            is_ok = False
            break
    return is_ok


def run():
    # noqa pylint: disable=use-a-generator
    any_prime_number = any([
        value
        for value in range(1, 50 + 1)
        if is_prime(value)
    ])
    # noqa pylint: enable=use-a-generator
    return any_prime_number


if __name__ == "__main__":
    a = 123 # noqa pylint: disable=invalid-name
    run()
```

# pylint - ejemplo 2: sin archivo de config (cont.)

```
C:\Windows\System32\cmd.exe

(match) C:\PythonProjects\PyCON\ pylint-example-2> pylint --fail-under=9.75 main.py
***** Module main
main.py:1:0: C0114: Missing module docstring (missing-module-docstring)
main.py:16:0: C0116: Missing function or method docstring (missing-function-docstring)

-----
Your code has been rated at 8.89/10 (previous run: 8.89/10, +0.00)
```

# pylint - ejemplo 3: con archivo de config.

```
● ● ● [MASTER]  
  
# Add files or directories to the blacklist. They should be base names, not paths.  
ignore=migrations  
  
# List of plugins (as comma separated values of python modules names) to load,  
# usually to register additional checkers.  
#load-plugins=pylint_django  
  
# In order to access some of the internal Django features to improve pylint inspections  
# you should also provide a Django settings module appropriate to your project  
#django-settings-module=settings  
  
[MESSAGES CONTROL]  
  
# C0114 : Missing module docstring (missing-module-docstring)  
# C0115 : Missing class docstring (missing-class-docstring)  
# C0116 : Missing function or method docstring (missing-function-docstring)  
# W0613 : Unused argument 'argument_name' (unused-argument)  
disable=C0114,C0115,C0116,W0613  
  
[FORMAT]  
  
# Maximum number of characters on a single line.  
max-line-length=120  
  
[DESIGN]  
  
# Maximum number of arguments for function / method.  
max-args=5  
  
# Minimum number of public methods for a class (see R0903).  
min-public-methods=1
```

# pylint - ejemplo 3: con archivo de config (cont.)

```
C:\Windows\System32\cmd.exe
(mach) C:\PythonProjects\PyCON\ pylint-example-3>pylint --fail-under=9.75 main.py
-----
Your code has been rated at 10.00/10 (previous run: 10.00/10, +0.00)
```

# pylint - plugins

- **pylint-django:**

- instalación: pip install pylint-django
- link al paquete en PyPI: <https://pypi.org/project/pylint-django/>
- link al repo en Github: <https://github.com/PyCQA/pylint-django>

- **pylint-flask:**

- instalación: pip install pylint-flask
- link al paquete en PyPI: <https://pypi.org/project/pylint-flask/>
- link al repo en Github: <https://github.com/jschaf/pylint-flask>

# pylint - documentación de referencia

- ejecución y archivos de configuración:

[https://pylint.pycqa.org/en/latest/user\\_guide/run.html](https://pylint.pycqa.org/en/latest/user_guide/run.html)

- ejemplos de ignorar alertas:

[https://pylint.pycqa.org/en/latest/user\\_guide/message-control.html](https://pylint.pycqa.org/en/latest/user_guide/message-control.html)

- tipos de mensajes, reportes y formatos de salida:

[https://pylint.pycqa.org/en/latest/user\\_guide/output.html](https://pylint.pycqa.org/en/latest/user_guide/output.html)

# Aspectos de seguridad

# Aspectos de seguridad

- paquetes
- código fuente

# safety

- sitios web:

- PyPI: <https://pypi.org/project/safety/>
- Github: <https://github.com/pyupio/safety>
- documentación: <https://pyup.io/docs/safety/what-is-safety/>

- instalación:

- pip install safety

- ejecución:

- ① abrir la consola y ubicarse en la raíz de la carpeta que contiene el código fuente
- ② escribir:

- para verificar las dependencias del entorno actual: `safety check`
- para verificar las dependencias en archivos de listado de dependencias:  
`safety check -r requirements.txt`

## safety - ejemplo 1: dependencias no vulnerables

# safety - ejemplo 2: dependencias vulnerables

```
+-----+
|          /$$$$$$$           /$$
|         /$$__  $$           | $$
|        /$$$$$$/ | $$ \_ / /$$$$$$ | /$$
|       /$$ _ \| $$ | $$$$_/ | $$ _/ | $$ | $$
|      | $$$_/ | $$ | $$/_/ | $$$$_/ | $$ | $$
|     \____/ | $$ | $$ | $$ | $$ | $$ | $$
|    /$$$$$/ | $$ | $$ | $$ | $$ | $$ | $$
|   /$$$$$/ | $$ | $$ | $$ | $$ | $$ | $$
|  /$$$$$/ | ____/ | ____/ | \____/ | \____/ | $$
| /$$$$$/ | \____/ | \____/ | \____/ | \____/ | $$
+-----+
| by pyup.io
+-----+
| REPORT
+-----+
| package           | installed   | affected      | source
+-----+
| insecure-package | 0.1.0       | <0.2.0        | changelog
+-----+
```

# bandit

- sitios web:

- PyPI: <https://pypi.org/project/bandit/>
- Github: <https://github.com/PyCQA/bandit>
- documentación: <https://bandit.readthedocs.io/en/latest/>

- instalación:

- pip install bandit

- ejecución:

- ① abrir la consola y ubicarse en la raíz de la carpeta que contiene el código fuente
- ② escribir:
  - para verificar 1 archivo: `bandit example1.py`
  - para verificar varios archivos: `bandit example1.py example2.py example3.py`
  - para verificar 1 carpeta y todos sus carpetas y archivos<sup>10</sup>: `bandit --ini .bandit`

- para ignorar sentencias:

- escribir `# nosec` al final de la sentencia

<sup>10</sup>debe usar archivo de configuración

# bandit - ejemplo 1: código sin problemas de seguridad

```
C:\Windows\System32\cmd.exe

(match) C:\PythonProjects\PyCON\bandit-1>bandit example1-1.py
[main]  INFO profile include tests: None
[main]  INFO profile exclude tests: None
[main]  INFO cli include tests: None
[main]  INFO cli exclude tests: None
[main]  INFO running on Python 3.8.5
[node_visitor] INFO Unable to find qualified name for module: example1-1.py
Run started:2021-11-05 07:01:33.119297

Test results:
    No issues identified.

Code scanned:
    Total lines of code: 13
    Total lines skipped (#nosec): 0

Run metrics:
    Total issues (by severity):
        Undefined: 0.0
        Low: 0.0
        Medium: 0.0
        High: 0.0
    Total issues (by confidence):
        Undefined: 0.0
        Low: 0.0
        Medium: 0.0
        High: 0.0
Files skipped (0):
```

# bandit - ejemplo 2: código con problemas de seguridad

```
--> Issue: [B608:hardcoded_sql_expressions] Possible SQL injection vector through string-based query construction.  
Severity: Medium Confidence: Low  
Location: scripts\rss.py:360  
More Info: https://bandit.readthedocs.io/en/latest/plugins/b608\_hardcoded\_sql\_expressions.html  
359         with conn.cursor() as cur:  
360             query = f"SELECT name FROM news_source WHERE id = {source_id}"  
361             cur.execute(query)  
  
--> Issue: [B101:assert_used] Use of assert detected. The enclosed code will be removed when compiling to optimised byte code.  
Severity: Low Confidence: High  
Location: scripts\rss.py:445  
More Info: https://bandit.readthedocs.io/en/latest/plugins/b101\_assert\_used.html  
444     def run(source_type: str = "both", inheritance_type: str = "both", verify_updated_time_of_source_news: bool = True):  
445         assert source_type in {"both", "national", "international"}  
446         assert inheritance_type in {"both", "single", "multiple"}  
  
-----  
  
Code scanned:  
    Total lines of code: 2314  
    Total lines skipped (#nosec): 0  
  
Run metrics:  
    Total issues (by severity):  
        Undefined: 0.0  
        Low: 4.0  
        Medium: 7.0  
        High: 0.0  
    Total issues (by confidence):  
        Undefined: 0.0  
        Low: 7.0  
        Medium: 0.0  
        High: 4.0  
Files skipped (0):
```

# bandit - ejemplo 3: código sin problemas de seguridad (archivo de config).

```
[bandit]
# runs bandit with: bandit --ini .bandit
targets: ./my_folder
recursive: true
format: html
output: bandit.html
```

# bandit - ejemplo 3: código sin problemas de seguridad (usando config).

```
C:\Windows\system32\cmd.exe

(match) C:\PythonProjects\██████████>bandit --ini .bandit
[main] INFO  Using command line arg for excluded paths
[main] INFO  Using ini file for selected targets
[main] INFO  Using ini file for recursive scan
[main] INFO  Using command line arg for aggregate output type
[main] INFO  Using command line arg for max code lines output for issue
[main] INFO  Using command line arg for severity level
[main] INFO  Using command line arg for confidence level
[main] INFO  Using command line arg for output format
[main] INFO  Using command line arg for output file
[main] INFO  profile include tests: None
[main] INFO  profile exclude tests: None
[main] INFO  cli include tests: None
[main] INFO  cli exclude tests: None
[main] INFO  running on Python 3.8.5
59 [0.. 50.. ]
Run started:2021-11-05 07:15:06.902731

Test results:
    No issues identified.

Code scanned:
    Total lines of code: 3173
    Total lines skipped (#nosec): 1

Run metrics:
    Total issues (by severity):
        Undefined: 0.0
        Low: 0.0
        Medium: 0.0
        High: 0.0
    Total issues (by confidence):
        Undefined: 0.0
        Low: 0.0
        Medium: 0.0
        High: 0.0
Files skipped (0):
```



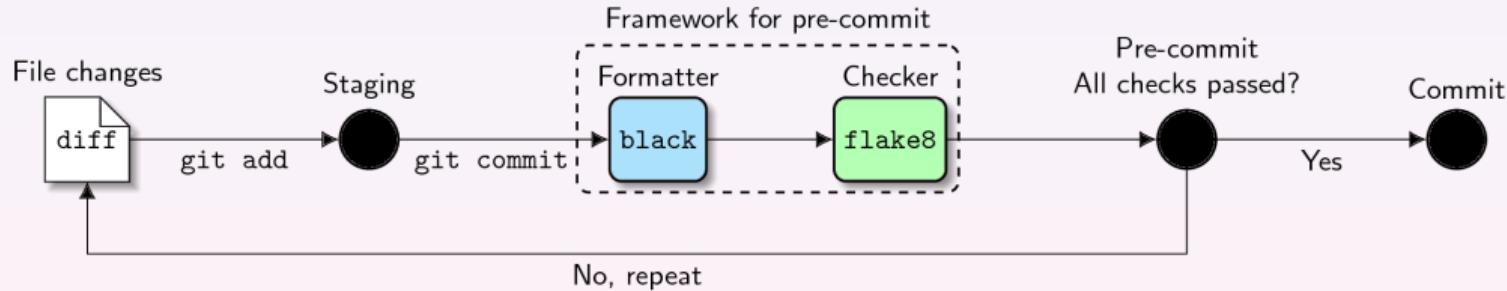
¿Existirá alguna forma de aplicar todo lo visto hasta ahora de forma automatizada?

# Hook en Git: pre-commit

# Hooks en Git

-  applypatch-msg.sample
-  commit-msg.sample
-  fsmonitor-watchman.sample
-  post-update.sample
-  pre-applypatch.sample
-  pre-commit.sample
-  pre-merge-commit.sample
-  prepare-commit-msg.sample
-  pre-push.sample
-  pre-rebase.sample
-  pre-receive.sample
-  push-to-checkout.sample
-  update.sample

# pre-commit - pipeline<sup>11</sup>



<sup>11</sup> <https://ljvmiranda921.github.io/notebook/2018/06/21/precommits-using-black-and-flake8/>

# pre-commit

- paquete de Python
- se instala con: `pip install pre-commit`
- debe configurarse mediante un archivo llamado `.pre-commit-config.yaml`
- sitios web:
  - paquete en PyPI: <https://pypi.org/project/pre-commit/>
  - repositorio en Github: <https://github.com/pre-commit/pre-commit>
  - documentación: <https://pre-commit.com/>

# pre-commit - paquetes a incorporar en el hook

Paquete	Licencia	Última versión	Fecha	Mantenedor	Estrellas	Lo usan
<a href="#">isort</a>	MIT	5.10.0	Nov 3, 2021	PyCQA	4.3k	199k
<a href="#">black</a>	MIT	21.10b0	Nov 1, 2021	PSF	22.9k	88.4k
<a href="#">flake8</a>	MIT	4.0.1	Oct 11, 2021	PyCQA	1.6k	197k
<a href="#">pylint</a>	GPL-2.0	2.11.1	Sep 16, 2021	PyCQA	3.6k	?
<a href="#">safety</a>	MIT	1.10.3	Ene 15, 2021	PyUp	1.1k	?
<a href="#">bandit</a>	Apache-2.0	1.7.0	Dic 13, 2020	PyCQA	3.6k	8.4k

# pre-commit - archivo de configuración (ejemplo 1)

```
● ● ●  
  
repos:  
- repo: https://github.com/pre-commit/pre-commit-hooks  
  rev: v2.3.0  
  hooks:  
    - id: check-yaml  
    - id: end-of-file-fixer  
    - id: trailing whitespace  
- repo: https://github.com/psf/black  
  rev: 19.3b0  
  hooks:  
    - id: black
```

.pre-commit-config.yaml

pre-commit - archivo de configuración (ejemplo 2)

```
fall_fast: true
repos:
  - repo: local
    hooks:
      - id: lsort
        name: lsort
        files: \.py$
        stages: [commit]
        language: system
        entry: lsort .
        types: [python]
        pass_filenames: false

      - id: black
        name: black
        stages: [commit]
        language: system
        entry: black .
        types: [python]

      - id: flake8
        name: flake8
        stages: [commit]
        language: system
        entry: flake8
        types: [python]

      - id: pylint
        name: pylint
        stages: [commit]
        language: system
        entry: pylint --fail-under=9.99 apps
        types: [python]
        pass_filenames: false

      - id: bandit
        name: bandit
        stages: [commit]
        language: system
        entry: bandit --ini .bandit
        types: [python]

      - id: safety
        name: safety
        stages: [commit]
        language: system
        entry: safety check -r requirements.txt -r requirements-dev.txt --full-report
        pass_filenames: false
```

# pre-commit - funcionamiento

- ① instalamos el script del hook mediante el comando `pre-commit install`
- ② probamos el hook mediante:
  - `precommit run` : aplica el hook sobre los archivos nuevos y/o modificados
  - `pre-commit run -all-files` : aplica el hook sobre todos los archivos

# pre-commit - ejecución manual: cuando falla

```
$ pre-commit run --all-files
[INFO] Initializing environment for https://github.com/pre-commit/pre-commit-hooks.
[INFO] Initializing environment for https://github.com/psf/black.
[INFO] Installing environment for https://github.com/pre-commit/pre-commit-hooks.
[INFO] Once installed this environment will be reused.
[INFO] This may take a few minutes...
[INFO] Installing environment for https://github.com/psf/black.
[INFO] Once installed this environment will be reused.
[INFO] This may take a few minutes...
Check Yaml.....Passed
Fix End of Files.....Passed
Trim Trailing Whitespace.....Failed
- hook id: trailing-whitespace
- exit code: 1

Files were modified by this hook. Additional output:

Fixing sample.py

black.....Passed
```

# pre-commit - ejecución manual: cuando no falla

```
C:\Windows\System32\cmd.exe
(match) C:\PythonProjects\██████████>pre-commit run
isort.....Passed
black.....Passed
flake8...Passed
pylint...Passed
bandit...Passed
safety...Passed
```

# pre-commit - ejecución automática

```
C:\Windows\System32\cmd.exe

(match) C:\PythonProjects\[REDACTED] >git commit -m "add JWT auth"
isort.....Passed
black.....Passed
flake8.....Passed
pylint.....Passed
bandit.....Passed
safety.....Passed
[temporal-branch 616d0f5] add JWT auth
 1 file changed, 54 insertions(+), 1 deletion(-)
```

Casos reales

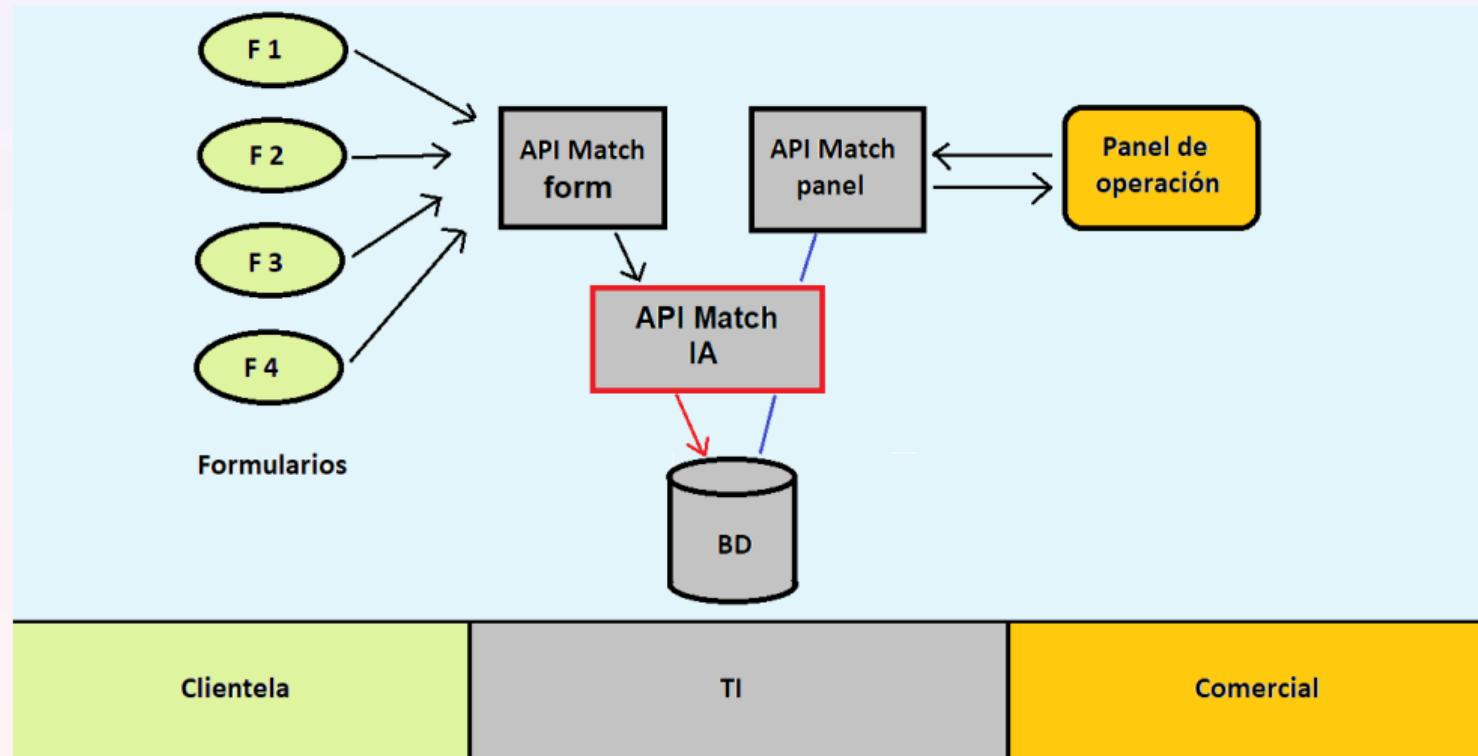
# Casos reales

- Match, un servicio comercial de Autofact
  - contratación del servicio: <https://www.autofact.cl/comparador-autos-match>
  - artículo en blog: <https://www.autofact.cl/blog/comprar-auto/tramites/tramites-para-comprar-un-auto-usado>
- Página personal de noticias
  - noticias de Chile: <https://1p.cl/webapps/news/national.html>
  - noticias internacionales: <https://1p.cl/webapps/news/international.html>
  - noticias de Hacker News: <https://1p.cl/webapps/news/hacker-news.html>

# Caso real laboral: Match - qué es

The screenshot shows a web browser window for <https://www.autofact.cl/comparador-autos-match>. The page features a blue header with the Autofact logo and navigation links: EL INFORME, PRECIO USADOS, TRANSFERENCIAS, MATCH (highlighted in orange), BLOG, and PEDIR MI INFORME. Below the header, there's a large graphic for 'AUTOMATCH'. It shows a hand holding a smartphone displaying a small blue car icon. To the right of the phone is a larger blue car with yellow dots around it, suggesting connectivity or data points. The text 'AUTOMATCH' is prominently displayed above the phone, with 'AUTO' in blue and 'MATCH' in orange. Below it, the text reads: 'La forma más fácil, moderna y segura para encontrar un auto usado'. In the bottom left corner of the main content area, there's a small decorative graphic of blue dots arranged in a cluster. The PyCon CHILE 2021 logo is visible in the bottom right corner.

# Caso real laboral: Match - microservicios



# Caso real laboral: Match - tecnologías

- backend:
  - microservicios que no realizan IA: Node.js, Express, Sequelize, pre-commit.
  - microservicio que realizan IA: Python, Django, DjangoRestFramework, pre-commit<sup>12</sup>, coverage, entre otros.
- frontend: Vue, Nuxt, Axios
- infra: AWS

<sup>12</sup>incorpora isort, black, flake8, safety, bandit

# Caso real personal: sitio web de noticias

- backend (Python):

- API: Django, DjangoRestFramework
- Webscrapping: requests, beautifulsoup4
- análisis de datos: pandas
- testing: unittest, coverage
- variables de entorno: python-decouple
- caching: python-memcached
- alertas (monitoreo): slack-sdk
- consola interactiva: ipython
- pre-commit: pre-commit<sup>13</sup>

- frontend: Vue, Buefy, Axios
- infra: Digital Ocean

<sup>13</sup>incorpora isort, black, flake8, pylint, safety, bandit

# Caso real personal: sitio web de noticias - nacionales (de Chile)

The screenshot shows a web browser window displaying a news website. The URL in the address bar is <https://1p.cl/webapps/news/national.html>. The page has a dark theme with a light-colored header and a white content area.

At the top of the page, there are several input fields and buttons:

- "Período de tiempo": "Ayer" (Yesterday) dropdown.
- "UTC": "-3" dropdown.
- "Buscar" button.
- "Entrar" button.
- Checkboxes for filtering: "ID" (unchecked), "Título" (checked), "Autor(es)" (unchecked), "Fecha" (checked), and "Fuente" (checked).
- "Items por página": "10" dropdown.
- "Ver fuentes" button.

The main content area displays a table of news articles:

Fuente	Título	Fecha
CHV Noticias	Harboe renuncia al PPD para "no traicionar sus principios": Pdta. del partido reconoce crisis de confianza	5/11/2021 23:59:54
El Ciudadano	Los principios de corrupción que rigen al neoliberalismo	5/11/2021 23:57:47
CHV Noticias	¿A quiénes les corresponde recibir el aguinaldo de Navidad 2021 y cuáles son sus montos?	5/11/2021 23:57:12
El Ciudadano	Mantienen apoyos en San Pablo Xochimehuacan: Eduardo Rivera	5/11/2021 23:53:07
CNN Chile	Confirman a Xavi Hernández como nuevo entrenador del Barcelona	5/11/2021 23:52:43
La Tercera	La columna de Garello: Dueño de nada	5/11/2021 23:49:24
FM2	¡Enamorados! Adele llevó su nueva relación al siguiente nivel	5/11/2021 23:48:06
CNN Chile	"Ya cumplió su ciclo": Harboe y más de 30 militantes renunciaron al PPD	5/11/2021 23:45:04
La Red	Jugamos al "Yo nunca nunca" con Juan Falcón y Betsy Camino	5/11/2021 23:42:39
Cooperativa	Tribunal Electoral destituyó a alcalde de Dalcahue, condenado por conducir ebrio un vehículo municipal	5/11/2021 23:41:00

At the bottom of the page, there are navigation links for pages 1 through 151, followed by arrows for more pages.

# Caso real personal: sitio web de noticias - nacionales (de Chile)

The screenshot shows a web browser displaying a news search results page. The URL is <https://1p.cl/webapps/news/national.html>. The search bar contains the query "candida". The search filters are set to "Título" checked, "Fecha" checked, and "Fuente" checked. The date range is "Ayer" and the time zone is "UTC -3". The results table has columns for "Fuente", "Título", and "Fecha". There are 10 items per page.

Fuente	Título	Fecha
el ciu	candida	
El Ciudadano	Esteban Arévalo candidato a senador por Biobio introduce acción legal contra Estado de Excepción en Wallmapu: «La pena se transforma en la necesidad de luchar»	5/11/2021 18:55:16
El Ciudadano	Caterina Simoncelli (DA), candidata a senadora por Región de Coquimbo: «La artesanía es manifestación de nuestra identidad cultural y parte de nuestro patrimonio»	5/11/2021 17:02:54
El Ciudadano	Candidata a diputada por el D23 Onésima Lienqueo: "Somos quienes llevamos años haciendo la pega que el Estado no hace"	5/11/2021 14:32:16
El Ciudadano	Rodrigo Urzúa, candidato a senador por RM: «Es necesario que nos demos cuenta que las malas decisiones políticas traen muerte»	5/11/2021 14:07:15
El Ciudadano	Candidato Gabriel Boric contagiado con variante Delta del Covid-19: Período de cuarentena podría prolongarse	5/11/2021 13:46:24
El Ciudadano	Matías Toledo, candidato a Diputado por el D-12 : Es importante irrumpir en el escenario político para darle bajada a la nueva Constitución con justicia social	5/11/2021 12:46:37

Pagination controls at the bottom: 1 < >

# Caso real personal: sitio web de noticias - nacionales (de Chile)

Fuentes de noticias			
Nº	Nombre	Etiquetas	Actualizado
1	24 Horas	[ "nacional", "televisión" ]	6/11/2021 3:15:01
2	ADN Radio	[ "nacional", "radio" ]	6/11/2021 3:15:01
3	CHV Noticias	[ "nacional", "televisión" ]	6/11/2021 3:15:01
4	CIPER Chile	[ "nacional", "fundación" ]	6/11/2021 3:15:01
5	CNN Chile	[ "nacional", "televisión" ]	6/11/2021 3:15:01
6	Cooperativa	[ "nacional", "radio" ]	6/11/2021 3:15:01
7	Diario Constitucional	[ "nacional", "diario" ]	6/11/2021 3:15:01
8	Diario El Día	[ "nacional", "Región de Coquimbo", "diario" ]	6/11/2021 3:15:01
9	Diario La Región	[ "nacional", "Región de Coquimbo", "diario" ]	6/11/2021 3:15:01
10	El Ciudadano	[ "nacional", "periódico" ]	6/11/2021 3:15:01
11	El Clarín	[ "nacional", "otro" ]	6/11/2021 3:15:01
12	El Informador Chile	[ "nacional", "otro" ]	6/11/2021 3:15:01
13	El Mostrador	[ "nacional", "otro" ]	6/11/2021 3:15:01
14	El Observatodo	[ "nacional", "otro", "Región de Coquimbo" ]	6/11/2021 3:15:01
15	El Periscopio	[ "nacional", "otro" ]	6/11/2021 3:15:01
16	El Serenense	[ "nacional", "otro", "Región de Coquimbo" ]	6/11/2021 3:15:01
17	FM2	[ "nacional", "radio" ]	6/11/2021 3:15:01
18	Futuro 360	[ "nacional", "televisión" ]	6/11/2021 3:15:01
19	Ilustrado	[ "nacional", "otro" ]	6/11/2021 3:15:01
20	Infogate	[ "nacional", "otro" ]	6/11/2021 3:15:01

# Caso real personal: sitio web de noticias - internacionales

The screenshot shows a web browser displaying a news website at <https://1p.cl/webapps/news/international.html>. The interface includes a header with search filters (Periodo de tiempo: Ayer, UTC: -4, Buscar), user authentication (Entrar), and sorting options (Items por página: 10, Ver fuentes). The main content area lists news articles from various sources:

Fuente	Título	Fecha
Xataka	GPD XP Game Console: una consola Android modular y portátil con 7.000 mAh de batería y mandos intercambiables	5/11/2021 10:00:52
Xataka Android	Xiaomi promete enfriar el doble los móviles: así es su nueva tecnología Loop LiquidCool	5/11/2021 8:16:52
Xataka Android	Tu wallpaper puede delatarte, así es como el fondo de pantalla identifica a tu teléfono	5/11/2021 7:30:53
Xataka Android	Google Arts & Culture te dice a qué obras de arte se parece tu mascota con una foto	5/11/2021 5:00:49
Xataka Ciencia	El problema de la obesidad solo está en sus inicios: para 2060 podría alcanzar un coste promedio del 3,6% del PIB de un país	5/11/2021 4:17:44
Xataka Home	Alexa celebra su cumpleaños con grandes descuentos, Días Sin IVA en MediaMarkt, televisores, sonido y más: Cazando Gangas	5/11/2021 8:08:53
Xataka Home	Las nuevas cerraduras Smart Locks 3.0 de Nuki quieren convertir tu móvil en la llave inteligente del hogar	5/11/2021 7:08:52
Xataka Móvil	Apple Watch Series 7, análisis: la pantalla más grande de la familia no es lo único que le hace destacar	5/11/2021 12:00:53
Xataka Móvil	Telefónica se posiciona en la lucha por los derechos del fútbol: le interesa, pero analizará si es rentable	5/11/2021 9:37:20
Xataka Móvil	Sonidos de fondo en iOS 15: así puedes relajarte con el móvil y sin aplicaciones de terceros	5/11/2021 8:34:51

Pagination controls at the bottom: 1, 2, 3, 4, <, >.

# Caso real personal: sitio web de noticias - Hacker News

The screenshot shows a custom-built interface for a news website, likely a clone of Hacker News. At the top, there's a header with a back button, forward button, refresh button, and a URL bar showing <https://1p.cl/webapps/news/hacker-news.html>. Below the header are search and filter controls. The search bar contains "Buscar". There are dropdowns for "Periodo de tiempo" (set to "Ayer") and "UTC" (-4). A "Entrar" button is on the right. Below these are several checkboxes: "ID" (unchecked), "Título" (checked), "Autor(a)" (unchecked), "Comentarios" (checked), "Puntaje" (checked), "Sitio web" (unchecked), and "Fecha" (checked). A dropdown for "Items por página" is set to 15. The main content area displays a table of news items. The columns are "Puntaje" (sorted by score), "Título" (title), "Comentarios" (number of comments), and "Fecha" (date). Each item row has a small green box with its ID (e.g., 860, 797, 462, etc.) on the left. The titles are hyperlinks. The table has a dark background with light-colored rows. At the bottom, there are navigation links for page 1, 2, ..., 17, <, and >.

Puntaje	Título	Comentarios	Fecha
860	Show HN: My wife is pregnant; naturally I made a baby-name app to prepare	325	5/11/2021 11:00:07
797	Pfizer's oral Covid-19 antiviral cuts hospitalization, death by 85%	1035	5/11/2021 10:30:06
462	Credit cards make money (by patio11)	310	5/11/2021 14:15:07
445	Sam Altman is investing in Helion for Fusion power	209	5/11/2021 11:45:06
318	So you want to Scrape like the Big Boys?	151	5/11/2021 00:05:09
256	Nuclear Power Is the Best Climate-Change Solution by Far	230	5/11/2021 20:30:03
244	IAB Europe consent pop-ups used by Google et al. are unlawful and infringe GDPR	240	5/11/2021 14:15:07
218	Ask HN: Coping with Cancer in the Family	128	5/11/2021 12:30:06
217	Things to say when you're losing a technical argument (2001)	117	5/11/2021 14:30:08
211	Orca – esoteric programming language, designed to create procedural sequencers	35	5/11/2021 15:00:20
209	Never Update Anything	144	5/11/2021 18:45:08
209	React Aria: a headless UI component library	71	5/11/2021 03:30:06
201	Why SQLite Does Not Use Git	259	5/11/2021 21:00:06
196	University of Florida reverses course, allows professors to testify	37	5/11/2021 16:00:11
191	An inside look into the illicit ad industry	83	5/11/2021 12:15:05



Fin