Pylint Report datacube

**Historial de versiones**

|  |  |  |  |
| --- | --- | --- | --- |
| **Fecha** | **Versión** | **Responsable** | **Descripción** |
| 14/05/2010 | 1.0 | Angelo Candia | Resultados aplicación de Pylint al repositorio “sag\_model\_datacube” con commit: “\_add time check for updating Datacube” |

Contenido

[**1.** **Introducción** 4](#_Toc40371870)

[**2.** **Herramientas utilizadas** 4](#_Toc40371871)

[**6.** **Resultado general: antes y después** 11](#_Toc40371872)

[**7.** **Resultado detallado: antes y después** 14](#_Toc40371873)

[4.1 run\_update\_datacube.py 14](#_Toc40371874)

[4.2 parameters.py 14](#_Toc40371875)

[4.3 table\_block\_model.py 14](#_Toc40371876)

[4.4 table\_dispatch.py 15](#_Toc40371877)

[4.5 table\_ramp\_up.py 16](#_Toc40371878)

[4.6 table\_tags.py 16](#_Toc40371879)

[4.7 00\_generate\_tags\_database.py 17](#_Toc40371880)

[4.8 01\_assign\_clusters\_block\_model.py 18](#_Toc40371881)

[4.9 02\_generate\_data\_prestockpile.py 19](#_Toc40371882)

[4.10 03\_estimate\_stockpile\_residency\_time.py 19](#_Toc40371883)

[4.11 04\_join\_block\_model\_dispatch.py 20](#_Toc40371884)

[4.12 05\_generate\_mineralogy\_database.py 21](#_Toc40371885)

[4.13 06\_join\_tags\_mineralogy.py 22](#_Toc40371886)

[4.14 07\_clean\_engineer\_tile.py 22](#_Toc40371887)

[4.15 Train\_stockpile\_model.py 24](#_Toc40371888)

[**8.** **Riesgos** 25](#_Toc40371889)

[**9.** **Conclusión** 25](#_Toc40371890)

# **Introducción**

El siguiente documento, describe los resultados antes y después de aplicar la herramienta “Pylint” al proyecto datacube, aplicando estándar de programación de estilo PEP8.

El objetivo principal es estandarizar el código que conforma el datacube, con el cual, se pueden hacer pruebas de mayor calidad, generar documentación exacta con herramientas como Sphinx o similares y, finalmente, permite que terceros del proyecto puedan comprender de mejor manera el proyecto al estar estandarizado.

# **Herramientas utilizadas**

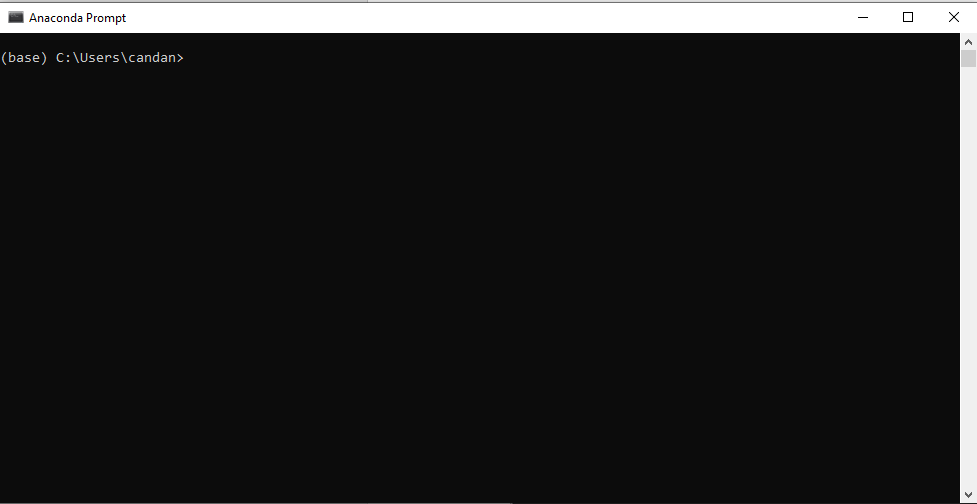
Se lista las herramientas utilizadas para realizar el análisis al proyecto:

* Herramientas informáticas en relación a Pylint:
  + **Anaconda 1.6.14:** Sistema de gestión de paquetes, incluye Spyder y Python.
  + **Python 3.6.5:** Incluido en Anaconda.
  + **Spyder 3.6:** Entorno de desarrollo, incluido en Anaconda.
  + **Módulo Pylint:** Herramienta verificador de código fuente, error y calidad para el lenguaje de programación Python. Incluido en Anaconda, integrado en Spyder.
* Herramientas gestión
  + **Word:** Usado para generar el presente documento.
  + **Excel:** Usado para generar resumen resultados.
* Otros:
  + **Git:** Usado para obtener el repositorio del datacube.

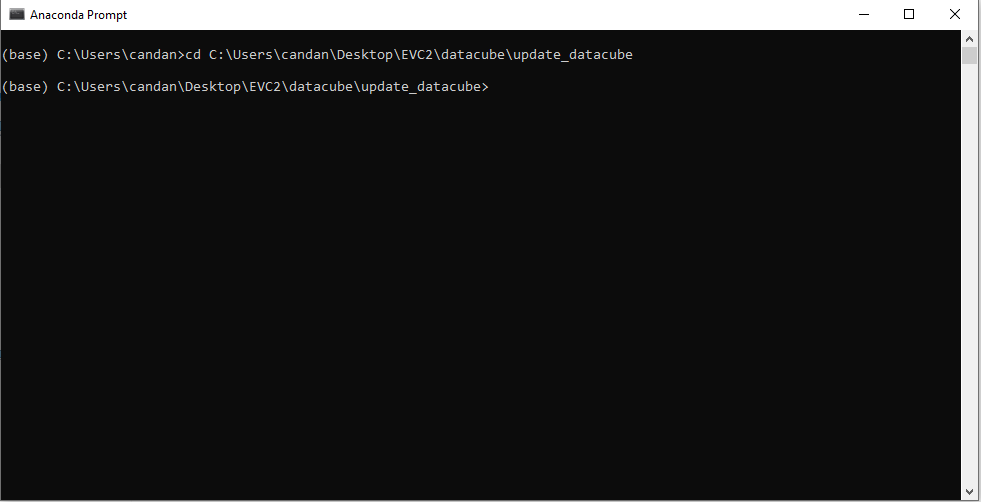
A continuación, se procede a demostrar las 2 formas de realizar el análisis de Pylint al datacube:

2.1 Ejecución Mediante CMD

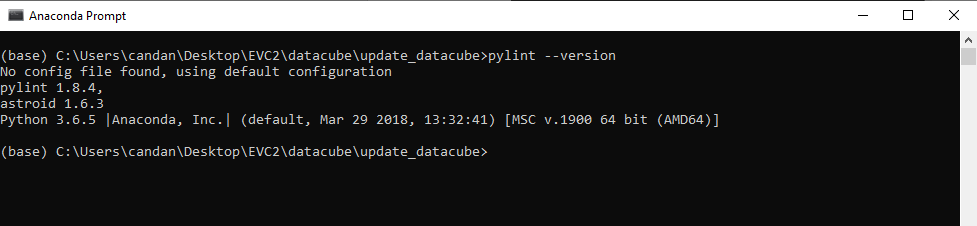
1. Abrir Anaconda Prompt:



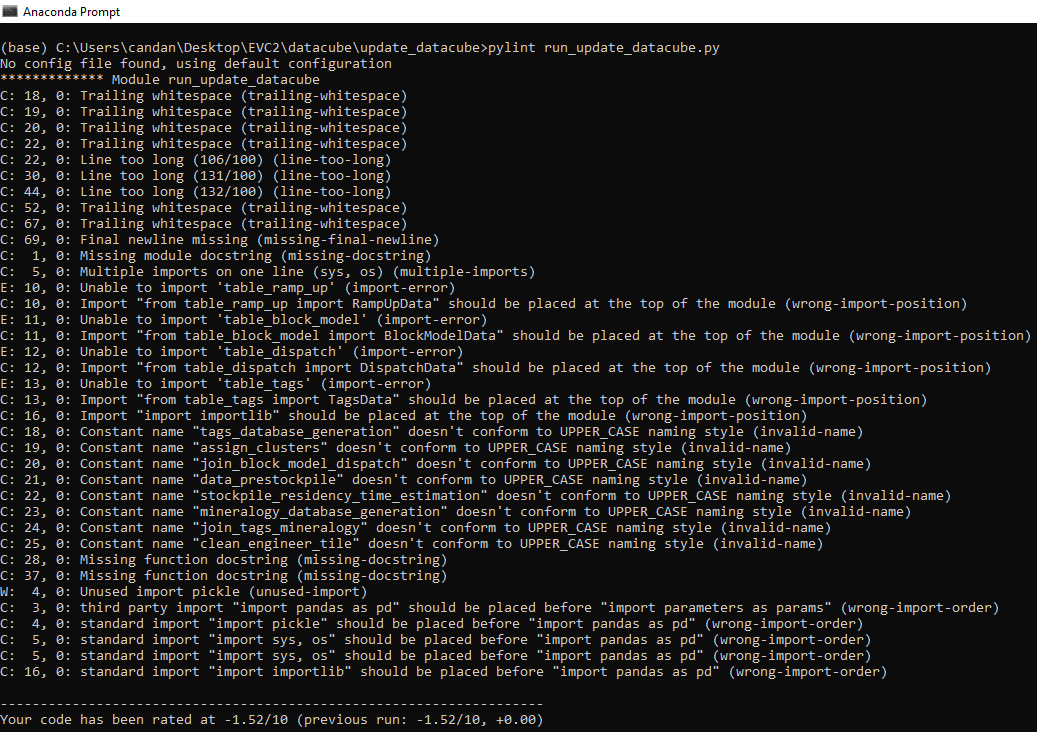
1. Ir al directorio del datacube:



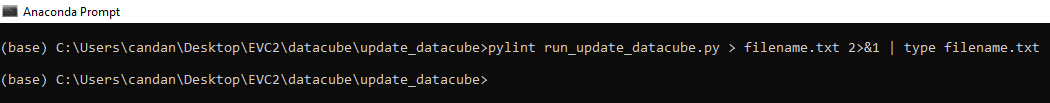
1. Ver versión de Pylint con el siguiente comando: “pylint –version”



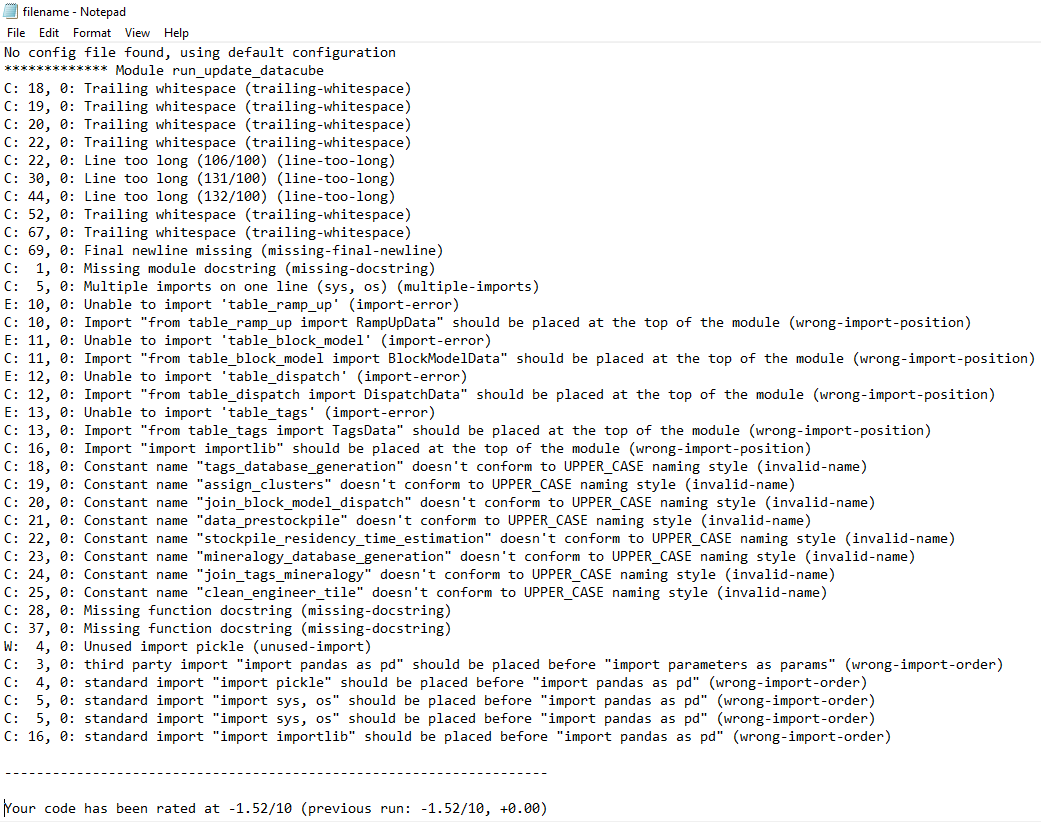
1. Ejecutar comando: “pylint run\_update\_datacube.py”. Si los pasos 1 a 3 están correctos, debería salir como output en la consola. El resultado del análisis:



1. Opcionalmente, es posible exportar estos resultados a un archivo TXT, mediante el siguiente comando:



Se obtiene:



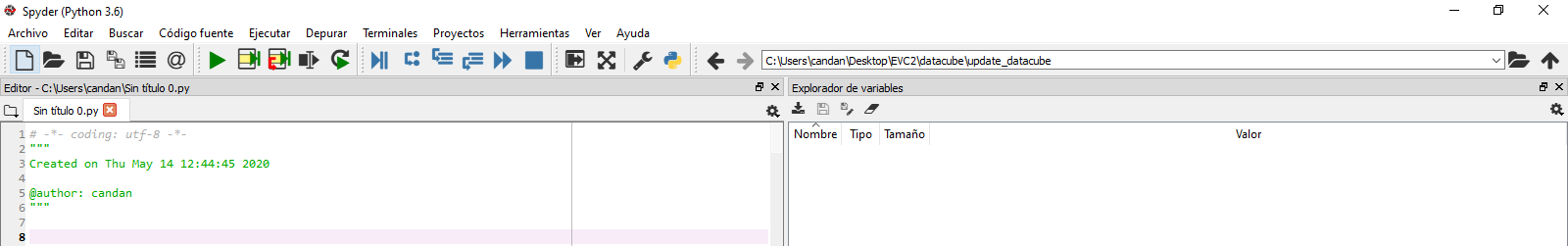
**Nota:** A partir de la version 2.0 de Pylint, el comando esta eliminado.

Para ejecutar con el resto de archivo, cambiar “run\_update\_datacube.py” por cualquier archivo “.py” del proyecto, incluir carpeta si aplica.

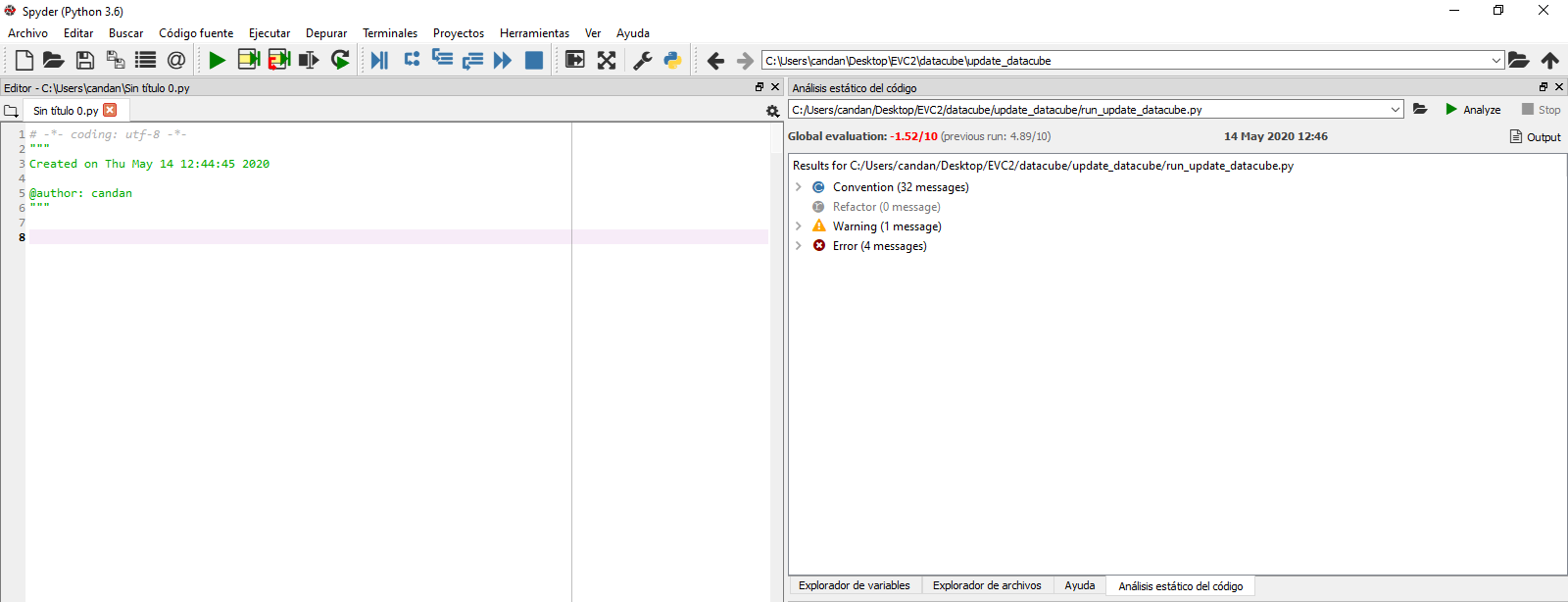
2.2 Ejecución Mediante Spyder

1. Abrir Aplicación Spyder
2. Seleccione la carpeta de trabajo (Works pace):

En este caso: C:\Users\candan\Desktop\EVC2\datacube\update\_datacube



1. Seleccione la opción “Código Fuente” y “Realizar análisis estático del código”:
2. En la parte derecha habla una pestaña “Análisis estático del código”
3. Para este ejemplo se analizará el archivo “run\_update\_datacube.py”:



Se muestran, al igual que en el caso anterior, las fallas al PEP8 y una evaluación del código de 0 a 10.

**Nota:** En la parte derecha, está el resultado. Para cambiar de archivo, clic en la carpeta a la izquierda del botón “Analize”. Seleccionar cualquier archivo “.py” del proyecto y si compila correctamente, entregara el resultado esperado. Este proceso suele demorar 1 minuto.

Al hacer clic en el botón “Output”, Spyder devuelve el mismo resultado mostrado en el proceso utilizando “Anaconda Prompt”.

**Observaciones:** Aunque es posible realizar el análisis al proyecto completo, es decir todos los archivos en un solo análisis.

Es recomendable realizar los análisis por cada archivo, por separado.

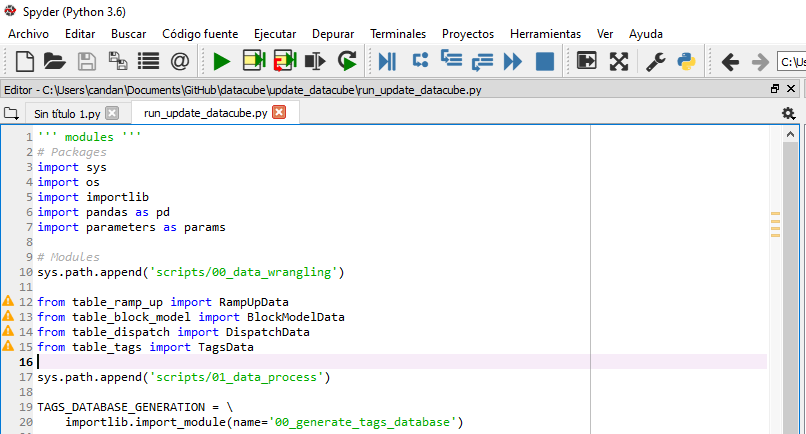
Esto permite corregir de manera más eficaz el archivo y así, proseguir al siguiente.

Antes utilizar “Pylint”, existe otra forma de implementar PEP8 en Spyder. Esta manera no entrega un resultado o evaluación. Pero, permite por otra parte, detectar inmediatamente cuando se está fallando en una regla al PEP8 en el momento de la codificación:

En el mismo Spyder ir a las opciones:

* Herramientas
* Preferencias
* Editor
* Análisis e introspección de código
* Seleccionar: Análisis de estilo del código en el editor

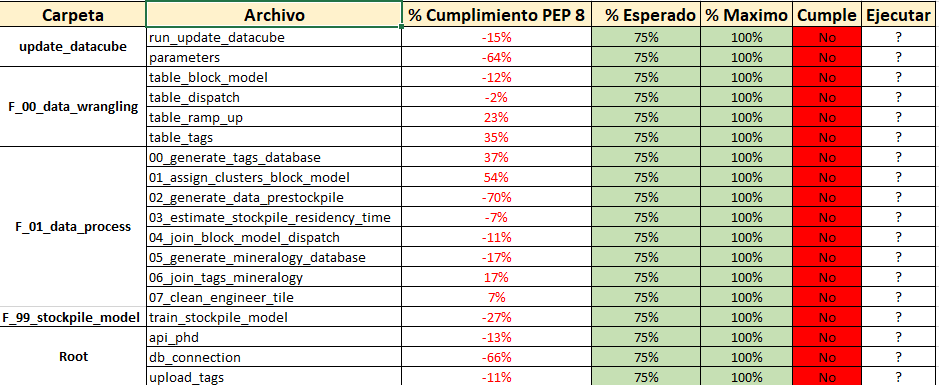
Con esto, cada vez que el desarrollador escriba un código que no cumpla con el PEP8, se mostrara un icono de tipo alert, en el número de línea y al poner el cursor en la alerta, mostrara, la falta a PEP8:

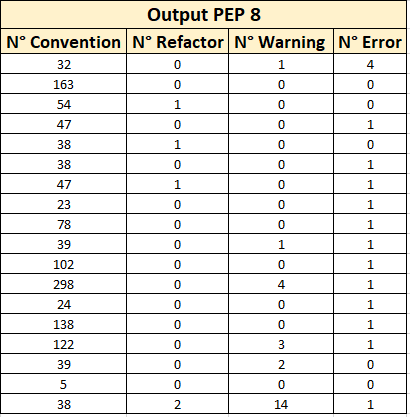


# **Resultado general: antes y después**

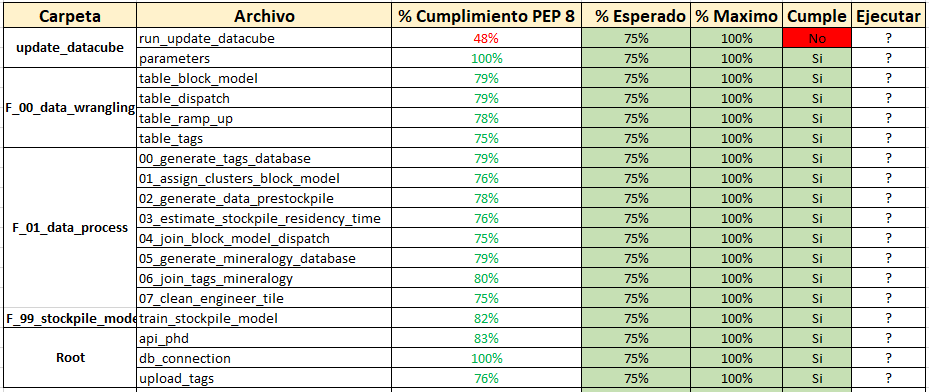
A continuación, se lista los resultados a nivel general, antes y después del análisis con la herramienta Pylint, dicho resultado, se obtuvo con la ejecución mediante Spyder:

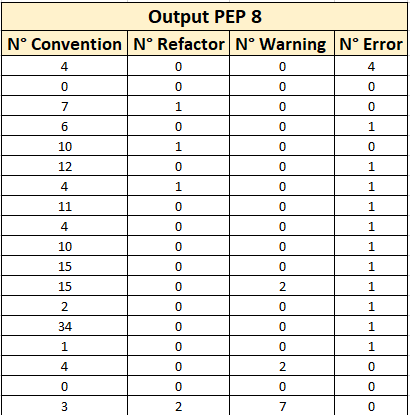
Antes de realizar modificaciones en base a Pylint:





Después de realizar modificaciones en base a Pylint:





**Observaciones:** Como se puede observar, el archivo “run\_update\_datacube.py”, solo cumple en un 48% el estándar de PEP8 en base a Pylint.

Esto se debe a que los 4 errores que se muestran en la tabla “Output” Bajan en gran medida la evaluación del archivo.

Dichos errores son de error al importar librerías de terceros, cuando en la ejecución del datacube, estos import funcionan correctamente.

El problema radica en la forma en que se importan, utilizando el código “sys.path.append()”. Pylint no reconoce esa ruta, por lo que las librerías de tercero las considera como fallidas.

La solución ante este problema, radica en cambiar la configuración de Pylint (No recomendable) o cambiar la forma de importación de dichos archivos.

Otro punto a considerar, Pylint considera como máximo líneas de 100 caracteres, superando ese número. Ya es considera una Falta a la nomenclatura de PEP8.

Oficialmente PEP8 como estándar, considera un máximo de 80 caracteres por línea.

Durante este análisis se consideró el máximo que Pylint considera, 100 caracteres como máximo.

# **Resultado detallado: antes y después**

A continuación, se lista los puntos de mejoras e implementación de PEP8 por cada archivo .py que conforma el proyecto datacube. Siendo los siguientes:

## 4.1 run\_update\_datacube.py

No config file found, using default configuration

\*\*\*\*\*\*\*\*\*\*\*\*\* Module run\_update\_datacube

E: 12, 0: Unable to import 'table\_ramp\_up' (import-error)

C: 12, 0: Import "from table\_ramp\_up import RampUpData" should be placed at the top of the module (wrong-import-position)

E: 13, 0: Unable to import 'table\_block\_model' (import-error)

C: 13, 0: Import "from table\_block\_model import BlockModelData" should be placed at the top of the module (wrong-import-position)

E: 14, 0: Unable to import 'table\_dispatch' (import-error)

C: 14, 0: Import "from table\_dispatch import DispatchData" should be placed at the top of the module (wrong-import-position)

E: 15, 0: Unable to import 'table\_tags' (import-error)

C: 15, 0: Import "from table\_tags import TagsData" should be placed at the top of the module (wrong-import-position)

-------------------------------------------------------------------

Your code has been rated at 4.89/10

## 4.2 parameters.py

No config file found, using default configuration

----------------------------------------------------------------------

Your code has been rated at 10.00/10

## 4.3 table\_block\_model.py

No config file found, using default configuration

\*\*\*\*\*\*\*\*\*\*\*\*\* Module table\_block\_model

C: 50, 4: Argument name "bm" doesn't conform to snake\_case naming style (invalid-name)

C: 59, 8: Variable name "tr" doesn't conform to snake\_case naming style (invalid-name)

C: 60, 8: Variable name "BM" doesn't conform to snake\_case naming style (invalid-name)

R: 50, 4: Method could be a function (no-self-use)

C: 67, 4: Argument name "BM" doesn't conform to snake\_case naming style (invalid-name)

C: 74,60: Variable name "f" doesn't conform to snake\_case naming style (invalid-name)

C: 82, 8: Variable name "BM" doesn't conform to snake\_case naming style (invalid-name)

C: 83, 8: Variable name "BM" doesn't conform to snake\_case naming style (invalid-name)

-------------------------------------------------------------------

Your code has been rated at 7.95/10

## 4.4 table\_dispatch.py

No config file found, using default configuration

\*\*\*\*\*\*\*\*\*\*\*\*\* Module table\_dispatch

E: 10, 0: Unable to import 'db\_connection' (import-error)

C: 10, 0: Import "from db\_connection import engine\_tr" should be placed at the top of the module (wrong-import-position)

C: 43,15: Variable name "dt" doesn't conform to snake\_case naming style (invalid-name)

C: 55,16: Variable name "MT\_retraining" doesn't conform to snake\_case naming style (invalid-name)

C: 57,16: Variable name "df" doesn't conform to snake\_case naming style (invalid-name)

C: 58,16: Variable name "MT\_retraining" doesn't conform to snake\_case naming style (invalid-name)

C:123,78: Variable name "f" doesn't conform to snake\_case naming style (invalid-name)

-------------------------------------------------------------------

Your code has been rated at 7.92/10

## 4.5 table\_ramp\_up.py

No config file found, using default configuration

\*\*\*\*\*\*\*\*\*\*\*\*\* Module table\_ramp\_up

C: 24, 0: Wrong continued indentation (remove 13 spaces).

delimiter=';', encoding='latin-1')

| ^ (bad-continuation)

C: 41, 0: Line too long (118/100) (line-too-long)

C: 50, 0: Trailing whitespace (trailing-whitespace)

C: 65, 0: Trailing whitespace (trailing-whitespace)

C: 65, 0: No space allowed around keyword argument assignment

camp\_dates[start + datetime.timedelta(days = i)] = (key, i + 1)

^ (bad-whitespace)

C: 73, 0: Trailing whitespace (trailing-whitespace)

C: 74, 0: Trailing whitespace (trailing-whitespace)

C: 75, 0: Trailing whitespace (trailing-whitespace)

C: 76, 0: Trailing whitespace (trailing-whitespace)

C: 78, 0: Trailing whitespace (trailing-whitespace)

R: 31, 4: Too many local variables (16/15) (too-many-locals)

------------------------------------------------------------------

Your code has been rated at 7.88/10

## 4.6 table\_tags.py

No config file found, using default configuration

\*\*\*\*\*\*\*\*\*\*\*\*\* Module table\_tags

C: 40, 0: No space allowed after bracket

list\_tags = ", ".join( repr(e) for e in self.list\_tags)

^ (bad-whitespace)

C: 66, 0: Trailing whitespace (trailing-whitespace)

C: 73, 0: Line too long (113/100) (line-too-long)

C: 95, 0: Line too long (134/100) (line-too-long)

C:117, 0: Line too long (118/100) (line-too-long)

E: 10, 0: Unable to import 'db\_connection' (import-error)

C: 10, 0: Import "from db\_connection import engine\_sag\_model" should be placed at the top of the module (wrong-import-position)

C: 50,15: Variable name "dt" doesn't conform to snake\_case naming style (invalid-name)

C: 65,16: Variable name "df" doesn't conform to snake\_case naming style (invalid-name)

C: 95,132: Variable name "f" doesn't conform to snake\_case naming style (invalid-name)

C:102,12: Variable name "m" doesn't conform to snake\_case naming style (invalid-name)

C:113, 4: Missing method docstring (missing-docstring)

C:117,116: Variable name "f" doesn't conform to snake\_case naming style (invalid-name)

------------------------------------------------------------------

Your code has been rated at 7.50/10

## 4.7 00\_generate\_tags\_database.py

No config file found, using default configuration

\*\*\*\*\*\*\*\*\*\*\*\*\* Module 00\_generate\_tags\_database

C: 1, 0: Module name "00\_generate\_tags\_database" doesn't conform to snake\_case naming style (invalid-name)

C: 10, 0: Import "import parameters as params" should be placed at the top of the module (wrong-import-position)

C: 23,97: Variable name "f" doesn't conform to snake\_case naming style (invalid-name)

C: 29,81: Variable name "f" doesn't conform to snake\_case naming style (invalid-name)

R:198, 0: Too many arguments (6/5) (too-many-arguments)

------------------------------------------------------------------

Your code has been rated at 9.43/10

## 4.8 01\_assign\_clusters\_block\_model.py

No config file found, using default configuration

\*\*\*\*\*\*\*\*\*\*\*\*\* Module 01\_assign\_clusters\_block\_model

C: 1, 0: Module name "01\_assign\_clusters\_block\_model" doesn't conform to snake\_case naming style (invalid-name)

C: 15, 0: Import "import parameters as params" should be placed at the top of the module (wrong-import-position)

C: 25, 8: Variable name "bm" doesn't conform to snake\_case naming style (invalid-name)

C: 73,82: Variable name "f" doesn't conform to snake\_case naming style (invalid-name)

C: 74, 8: Variable name "KN" doesn't conform to snake\_case naming style (invalid-name)

C: 89, 0: Argument name "bm" doesn't conform to snake\_case naming style (invalid-name)

C:109, 0: Argument name "bm" doesn't conform to snake\_case naming style (invalid-name)

C:122, 0: Argument name "bm" doesn't conform to snake\_case naming style (invalid-name)

C:128,94: Variable name "f" doesn't conform to snake\_case naming style (invalid-name)

C:134, 4: Variable name "bm" doesn't conform to snake\_case naming style (invalid-name)

C:137, 4: Variable name "bm" doesn't conform to snake\_case naming style (invalid-name)

------------------------------------------------------------------

Your code has been rated at 8.36/10

## 4.9 02\_generate\_data\_prestockpile.py

No config file found, using default configuration

\*\*\*\*\*\*\*\*\*\*\*\*\* Module 02\_generate\_data\_prestockpile

C: 1, 0: Module name "02\_generate\_data\_prestockpile" doesn't conform to snake\_case naming style (invalid-name)

C: 10, 0: Import "import parameters as params" should be placed at the top of the module (wrong-import-position)

C: 43,94: Variable name "f" doesn't conform to snake\_case naming style (invalid-name)

C: 98, 0: Function name "run\_data\_prestockpile\_generation" doesn't conform to snake\_case naming style (invalid-name)

------------------------------------------------------------------

Your code has been rated at 9.02/10

## 4.10 03\_estimate\_stockpile\_residency\_time.py

No config file found, using default configuration

\*\*\*\*\*\*\*\*\*\*\*\*\* Module 03\_estimate\_stockpile\_residency\_time

C: 1, 0: Module name "03\_estimate\_stockpile\_residency\_time" doesn't conform to snake\_case naming style (invalid-name)

C: 11, 0: Import "import parameters as params" should be placed at the top of the module (wrong-import-position)

C: 25,82: Variable name "f" doesn't conform to snake\_case naming style (invalid-name)

C: 32,73: Variable name "f" doesn't conform to snake\_case naming style (invalid-name)

C: 86, 4: Variable name "X\_to\_predict" doesn't conform to snake\_case naming style (invalid-name)

C: 91, 0: Argument name "X\_to\_predict" doesn't conform to snake\_case naming style (invalid-name)

C:101, 4: Variable name "df\_topredict\_MB" doesn't conform to snake\_case naming style (invalid-name)

C:104, 4: Variable name "df\_topredict\_MB" doesn't conform to snake\_case naming style (invalid-name)

C:173, 0: Function name "run\_stockpile\_residency\_time\_estimation" doesn't conform to snake\_case naming style (invalid-name)

C:177,27: Variable name "X\_to\_predict" doesn't conform to snake\_case naming style (invalid-name)

------------------------------------------------------------------

Your code has been rated at 8.46/10

## 4.11 04\_join\_block\_model\_dispatch.py

No config file found, using default configuration

\*\*\*\*\*\*\*\*\*\*\*\*\* Module 04\_join\_block\_model\_dispatch

C: 1, 0: Module name "04\_join\_block\_model\_dispatch" doesn't conform to snake\_case naming style (invalid-name)

C: 11, 0: Import "import parameters as params" should be placed at the top of the module (wrong-import-position)

C: 23,94: Variable name "f" doesn't conform to snake\_case naming style (invalid-name)

C: 24, 8: Variable name "BM" doesn't conform to snake\_case naming style (invalid-name)

C: 29,73: Variable name "f" doesn't conform to snake\_case naming style (invalid-name)

C: 35, 0: Argument name "BM" doesn't conform to snake\_case naming style (invalid-name)

C: 45, 4: Variable name "BM\_D" doesn't conform to snake\_case naming style (invalid-name)

C: 65, 4: Variable name "BM\_D" doesn't conform to snake\_case naming style (invalid-name)

C: 72, 0: Argument name "BM\_D" doesn't conform to snake\_case naming style (invalid-name)

C: 78,16: Comparison to False should be 'not expr' or 'expr is False' (singleton-comparison)

C: 82,93: Variable name "f" doesn't conform to snake\_case naming style (invalid-name)

C: 88, 4: Variable name "BM" doesn't conform to snake\_case naming style (invalid-name)

C: 89, 4: Variable name "BM\_D" doesn't conform to snake\_case naming style (invalid-name)

C: 96, 0: Argument name "BM\_D" doesn't conform to snake\_case naming style (invalid-name)

C:127,30: Comparison to True should be just 'expr' or 'expr is True' (singleton-comparison)

------------------------------------------------------------------

Your code has been rated at 7.00/10

## 4.12 05\_generate\_mineralogy\_database.py

No config file found, using default configuration

\*\*\*\*\*\*\*\*\*\*\*\*\* Module 05\_generate\_mineralogy\_database

C: 48, 0: Trailing whitespace (trailing-whitespace)

C: 1, 0: Module name "05\_generate\_mineralogy\_database" doesn't conform to snake\_case naming style (invalid-name)

C: 12, 0: Import "import parameters as params" should be placed at the top of the module (wrong-import-position)

C: 79, 0: Function name "run\_mineralogy\_database\_generation" doesn't conform to snake\_case naming style (invalid-name)

C:141, 4: Variable name "SPI\_by\_type\_df\_pct" doesn't conform to snake\_case naming style (invalid-name)

C:188, 0: Function name "create\_SPI\_buckets" doesn't conform to snake\_case naming style (invalid-name)

C:195, 4: Variable name "SPI" doesn't conform to snake\_case naming style (invalid-name)

C:202, 4: Variable name "SPI\_by\_type" doesn't conform to snake\_case naming style (invalid-name)

C:204, 4: Variable name "SPI\_by\_type" doesn't conform to snake\_case naming style (invalid-name)

C:209, 4: Variable name "SPI\_by\_type\_df" doesn't conform to snake\_case naming style (invalid-name)

C:217, 4: Variable name "SPI\_by\_type\_df" doesn't conform to snake\_case naming style (invalid-name)

C:231, 0: Argument name "x" doesn't conform to snake\_case naming style (invalid-name)

C:353, 4: Variable name "Ton\_sag" doesn't conform to snake\_case naming style (invalid-name)

W:361,53: Cell variable i defined in loop (cell-var-from-loop)

W:350,39: Unused argument 'origin\_flag' (unused-argument)

C:418, 4: Variable name "pivotH" doesn't conform to snake\_case naming style (invalid-name)

C:452, 4: Variable name "pnullH" doesn't conform to snake\_case naming style (invalid-name)

---------------------------------------------------------------------

Your code has been rated at 8.44/10

## 4.13 06\_join\_tags\_mineralogy.py

No config file found, using default configuration

\*\*\*\*\*\*\*\*\*\*\*\*\* Module 06\_join\_tags\_mineralogy

C: 1, 0: Module name "06\_join\_tags\_mineralogy" doesn't conform to snake\_case naming style (invalid-name)

C: 9, 0: Import "import parameters as params" should be placed at the top of the module (wrong-import-position)

------------------------------------------------------------------

Your code has been rated at 9.44/10

## 4.14 07\_clean\_engineer\_tile.py

No config file found, using default configuration

\*\*\*\*\*\*\*\*\*\*\*\*\* Module 07\_clean\_engineer\_tile

C: 58, 0: Line too long (140/100) (line-too-long)

C: 71, 0: Line too long (200/100) (line-too-long)

C:107, 0: Line too long (127/100) (line-too-long)

C:126, 0: Line too long (127/100) (line-too-long)

C:175, 0: Line too long (232/100) (line-too-long)

C:223, 0: Line too long (111/100) (line-too-long)

C:243, 0: Line too long (142/100) (line-too-long)

C: 1, 0: Module name "07\_clean\_engineer\_tile" doesn't conform to snake\_case naming style (invalid-name)

C: 13, 0: Import "import parameters as params" should be placed at the top of the module (wrong-import-position)

C:140, 0: Argument name "df" doesn't conform to snake\_case naming style (invalid-name)

C:170, 0: Argument name "df" doesn't conform to snake\_case naming style (invalid-name)

C:179, 0: Argument name "df" doesn't conform to snake\_case naming style (invalid-name)

C:190, 0: Argument name "df" doesn't conform to snake\_case naming style (invalid-name)

C:198, 4: Variable name "X" doesn't conform to snake\_case naming style (invalid-name)

C:199, 4: Variable name "y" doesn't conform to snake\_case naming style (invalid-name)

C:223,109: Variable name "f" doesn't conform to snake\_case naming style (invalid-name)

C:229, 0: Argument name "df" doesn't conform to snake\_case naming style (invalid-name)

C:240, 4: Variable name "VOIDAGE" doesn't conform to snake\_case naming style (invalid-name)

C:250, 0: Function name "aggregate\_coarse\_and\_medium\_feed" doesn't conform to snake\_case naming style (invalid-name)

C:250, 0: Argument name "df" doesn't conform to snake\_case naming style (invalid-name)

C:260, 0: Argument name "df" doesn't conform to snake\_case naming style (invalid-name)

C:277, 0: Argument name "df" doesn't conform to snake\_case naming style (invalid-name)

C:287, 0: Argument name "df" doesn't conform to snake\_case naming style (invalid-name)

C:304, 0: Argument name "df" doesn't conform to snake\_case naming style (invalid-name)

C:309, 4: Variable name "td" doesn't conform to snake\_case naming style (invalid-name)

C:317, 0: Function name "create\_grouping\_by\_tile\_duration" doesn't conform to snake\_case naming style (invalid-name)

C:317, 0: Argument name "df" doesn't conform to snake\_case naming style (invalid-name)

C:328, 0: Argument name "df" doesn't conform to snake\_case naming style (invalid-name)

C:336, 0: Argument name "df" doesn't conform to snake\_case naming style (invalid-name)

C:347, 0: Argument name "df" doesn't conform to snake\_case naming style (invalid-name)

C:355, 0: Argument name "df" doesn't conform to snake\_case naming style (invalid-name)

C:367, 0: Argument name "df" doesn't conform to snake\_case naming style (invalid-name)

C:380, 0: Argument name "df" doesn't conform to snake\_case naming style (invalid-name)

C:393, 0: Argument name "df" doesn't conform to snake\_case naming style (invalid-name)

------------------------------------------------------------------

Your code has been rated at 7.82/10

## 4.15 Train\_stockpile\_model.py

No config file found, using default configuration

\*\*\*\*\*\*\*\*\*\*\*\*\* Module train\_stockpile\_model

C: 14, 0: Import "import parameters as params" should be placed at the top of the module (wrong-import-position)

---------------------------------------------------------------------

Your code has been rated at 9.71/10

Debido al tamaño de las primeras ejecuciones, los resultados al ejecutar Pylint por primera vez, están adjuntos en la carpeta “Resultados”, separados en dos carpetas “Antes” y “Después”.

# **Riesgos**

Este documento es sensible ante cualquier modificación del datacube. Puesto que, al realizar cambios al código como:

* Actualización de código
* Corrección de código
* Añadir nuevos trozos de código

Puede provocar que al volver a ejecutar Pylint en los archivos modificados, la evaluación, así como el total de faltas a PEP8 puedan aumentar considerablemente y así vez, provoque que no se cumpla con el mínimo porcentaje de cumplimiento que actualmente es del 70%

Es recomendable, que, al finalizar alguna modificación o cambios importantes al proyecto, o al finalizar una etapa del proyecto. Volver a realizar el análisis con Pylint, para actualizar el porcentaje de cumplimiento con PEP8

# **Conclusión**

Implementar pylint al datacube ha permitido que el código sea mejor leíble y comprendido por terceros desarrolladores que puedan entrar en la necesidad de ejecutar el proceso.

A su vez, con el código corregido con PEP8 se puede lograr ejecutar pruebas de mayor calidad de tipo unitario o integral y, finalmente, realizar documentación automática utilizando herramientas como Sphinx