

Práctica 1

Desarrollo de Software - Gr 89



Adam Kowalczyk Holtsova - 100525023

Sergio Diaz Sanchez - 100550128

Introduction of Code regulations

Coding standards are very important among programmers, which results in optimized, easy readable and proper codes for each programmer that will read them, specially for projects that will be shared among a team of programmers.

For this project, we have to update many grammatical inconsistencies and some improper code.

Coding standards

Class naming convention

Classes shall use PascalCase naming convention.

Wrong:

```
class enterprise_management_exception(Exception):
```

Right:

```
class EnterpriseManagementException(Exception):
```

Function naming convention

Function names shall use snake_case naming convention.

Wrong:

Right:

```
def ValidateCif(self, cif):
```

```
def validate_cif(self, cif):
```

Constants naming convention

Constants shall use UPPER_CASE naming convention.

Wrong:

```
letters = string.ascii_letters + string.punctuation + string.digits
```

Right:

```
LETTERS = string.ascii_letters + string.punctuation + string.digits
```

Variables naming convention

Variables shall use snake_case naming convention.

Wrong:

```
org_letter = cif[0].upper()
central_body = cif[1:8]
control_char = cif[8]
```

Right:

```
ORG_LETTER = cif[0].upper()
CentralBody = cif[1:8]
controlChar = cif[8]
```

Modified rules

Python version

We change the python version according to the project's statement.

```
# MODIFIED RULE
# Minimum Python version to use for version dependent checks.
# py-version=3.14
py-version=3.13
```

Docstring minimum length

We set a minimum length for docstrings so that they are concise, but meaningful.

```
# MODIFIED RULE
# Minimum line length for functions/classes that require docstrings, shorter
# ones are exempt. Requires docstrings to be meaningful (at least 10 chars).
# docstring-min-length=-1
docstring-min-length=10
```

Good names

Good names are variable names which pylint doesn't warn you about, such as, "i", "j", which are used for loops. We add "cif" and "id".

```
# MODIFIED RULE
# Good variable names which should always be accepted, separated by a comma.
# Adds project-specific short names: 'cif' (Tax ID used in UC3MConsulting) and
# 'id' (common identifier) to the default accepted short names.
# good-names=i,j,k,ex,Run,_
good-names=i,
    j,
    k,
    ex,
    Run,
    _,
    cif,
    id
```

Maximum number of arguments for a function

We increase the number of arguments to 6 so that we have more flexibility when creating functions and assigning arguments.

```
# MODIFIED RULE
# Maximum number of arguments for function / method.
# Increased from 5 to 6 to allow slightly more flexibility in method signatures
# while still encouraging concise interfaces.
# max-args=5
max-args=6
```

Maximum number of public methods for a class

We increase the number of public methods so that we can set more extensive interfaces.

```
# MODIFIED RULE
# Maximum number of public methods for a class (see R0904).
# Increased from 20 to 25 to allow service classes to expose a extensive interface.
# max-public-methods=20
max-public-methods=25
```

Maximum number of returns for function

For validation logic, we use many returns, we extend the rule to 8.

```
# MODIFIED RULE
# Maximum number of return / yield for function / method body.
# Increased from 6 to 8 to support validation functions that return early
# at multiple guard clauses (e.g. CIF validation logic).
# max-returns=6
max-returns=8
```

Minimum number of public methods for a class

We reduce the minimum number of public methods to 1, so that a validator class with only validate() won't trigger any warnings.

```
# MODIFIED RULE
# Minimum number of public methods for a class (see R0903).
# Reduced from 2 to 1 to allow single-responsibility classes that expose
# only one public operation (e.g. a validator class with only validate()).
# min-public-methods=2
min-public-methods=1
```

Maximum number of characters on a single line

We increase the number of characters on a single line so that we have more margin to use.

```
# MODIFIED RULE
# Maximum number of characters on a single line.
# Increased from 100 to 120 to give developers more space for readable
# expressions and long string literals without forced wrapping.
# max-line-length=100
max-line-length=120
```

Maximum number of lines in a module.

1200 lines is an appropriate number for larger modules.

```
# MODIFIED RULE
# Maximum number of lines in a module.
# Increased from 1000 to 1200 to accommodate well-structured but larger modules.
# max-module-lines=1000
max-module-lines=1200
```

Disable import errors

We disable import errors which show when pylint doesn't properly detect root's directory.

```
# MODIFIED RULE
# Disable the message, report, category or checker with the given id(s).
# Added 'import-error' to suppress false positives when Pylint cannot resolve
# project-specific module imports in the virtual environment context.
# disable=raw-checker-failed,bad-inline-option,locally-disabled,file-ignored,suppressed-message,useless-suppression,deprecated-pragma,use-symbolic-message-instead,use-
disable=raw-checker-failed,
    bad-inline-option,
    locally-disabled,
    file-ignored,
    suppressed-message,
    useless-suppression,
    deprecated-pragma,
    use-symbolic-message-instead,
    use-implicit-booleness-not-comparison-to-string,
    use-implicit-booleness-not-comparison-to-zero,
    import-error
```

Pylint scans

```
✓ Pylint found 1 error, 11 conventions in 1 file
  ✓ main.py : 1 error, 11 conventions
      Missing module docstring (1:0) [missing-module-docstring]
      Unable to import 'UC3MConsulting' (1:0) [import-error]
      Constant name "letters" doesn't conform to UPPER_CASE naming style (5:0) [invalid-name]
      Constant name "shift" doesn't conform to UPPER_CASE naming style (6:0) [invalid-name]
      Function name "Encode" doesn't conform to snake_case naming style (9:0) [invalid-name]
      Function name "Decode" doesn't conform to snake_case naming style (19:0) [invalid-name]
      Missing function or method docstring (30:0) [missing-function-docstring]
      Variable name "strRes" doesn't conform to snake_case naming style (34:4) [invalid-name]
      Unnecessarily calls dunder method __str__. Use str built-in function. (34:13) [unnecessary-dunder-call]
      Variable name "EncodeRes" doesn't conform to snake_case naming style (36:4) [invalid-name]
      Variable name "DecodeRes" doesn't conform to snake_case naming style (38:4) [invalid-name]
      standard import "string" should be placed before first party import "UC3MConsulting.EnterpriseManager" (2:0) [wrong-import-order]
```

main.py before fixing

Pylint found no problems

2.GE1 >  main.py

main.py after fixing

✓ Pylint found 3 errors, 2 conventions in 1 file

 ✓  __init__.py : 3 errors, 2 conventions

 Missing module docstring (1:0) [missing-module-docstring]
 Module name "GE1" doesn't conform to snake_case naming style (1:0) [invalid-name]
 Unable to import 'UC3MConsulting.EnterpriseRequest' (1:0) [import-error]
 Unable to import 'UC3MConsulting.EnterpriseManager' (2:0) [import-error]
 Unable to import 'UC3MConsulting.EnterpriseManagementException' (3:0) [import-error]

__init__.py before fixing

✓ Pylint found 1 convention in 1 file

 ✓  __init__.py : 1 convention

 Module name "GE1" doesn't conform to snake_case naming style (1:0) [invalid-name]

__init__.py after fixing (The error is omitted as this error comes from the naming of the project)

✓ Pylint found 3 errors, 2 conventions in 1 file

 ✓  __init__.py : 3 errors, 2 conventions

 Missing module docstring (1:0) [missing-module-docstring]
 Module name "UC3MConsulting" doesn't conform to snake_case naming style (1:0) [invalid-name]
 Unable to import 'UC3MConsulting.EnterpriseRequest' (1:0) [import-error]
 Unable to import 'UC3MConsulting.EnterpriseManager' (2:0) [import-error]
 Unable to import 'UC3MConsulting.EnterpriseManagementException' (3:0) [import-error]

UC3MConsulting/__init__.py before fixing

Pylint found no problems

UC3MConsulting/__init__.py after fixing

```
✓ Pylint found 4 conventions in 1 file
  ✓ 📄 EnterpriseManagementException.py : 4 conventions
    Trailing newlines (13:0) [trailing-newlines]
    Missing module docstring (1:0) [missing-module-docstring]
    Module name "EnterpriseManagementException" doesn't conform to snake_case naming style (1:0) [invalid-name]
    Missing class docstring (1:0) [missing-class-docstring]
```

UC3MConsulting/EnterpriseManagementException.py before fixing

Pylint found no problems

UC3MConsulting/EnterpriseManagementException.py after fixing

```
✓ Pylint found 2 errors, 2 warnings, 12 conventions in 1 file
  ✓ 📄 EnterpriseManager.py : 2 errors, 2 warnings, 12 conventions
    Final newline missing (34:0) [missing-final-newline]
    Missing module docstring (1:0) [missing-module-docstring]
    Module name "EnterpriseManager" doesn't conform to snake_case naming style (1:0) [invalid-name]
    Unable to import 'G89.2026.T12.GE1.UC3MConsulting.EnterpriseManagementException' (2:0) [import-error]
    Unable to import 'G89.2026.T12.GE1.UC3MConsulting.EnterpriseRequest' (3:0) [import-error]
    Missing class docstring (5:0) [missing-class-docstring]
    Method name "ValidateCIF" doesn't conform to snake_case naming style (9:4) [invalid-name]
    Argument name "CiF" doesn't conform to snake_case naming style (9:27) [invalid-name]
    Unused argument 'CiF' (9:27) [unused-argument]
    Missing function or method docstring (14:4) [missing-function-docstring]
    Method name "ReadproductcodefromJSON" doesn't conform to snake_case naming style (14:4) [invalid-name]
    Using open without explicitly specifying an encoding (17:17) [unspecified-encoding]
    Variable name "DATA" doesn't conform to snake_case naming style (18:16) [invalid-name]
    Variable name "T_CIF" doesn't conform to snake_case naming style (26:12) [invalid-name]
    Variable name "T_PHONE" doesn't conform to snake_case naming style (27:12) [invalid-name]
    Variable name "E_NAME" doesn't conform to snake_case naming style (28:12) [invalid-name]
```

UC3MConsulting/EnterpriseManagement.py before fixing

Pylint found no problems

UC3MConsulting/EnterpriseManagement.py after fixing

```
✓ Pylint found 1 warning, 17 conventions in 1 file
  ✓ 📄 EnterpriseRequest.py : 1 warning, 17 conventions
    Final newline missing (35:0) [missing-final-newline]
    Missing module docstring (1:0) [missing-module-docstring]
    Module name "EnterpriseRequest" doesn't conform to snake_case naming style (1:0) [invalid-name]
    Missing class docstring (5:0) [missing-class-docstring]
    Attribute name "__EnterpriseNAME" doesn't conform to snake_case naming style (7:8) [invalid-name]
    Attribute name "__cIF" doesn't conform to snake_case naming style (8:8) [invalid-name]
    Attribute name "__pHOnE" doesn't conform to snake_case naming style (9:8) [invalid-name]
    Attribute name "__timeStamp" doesn't conform to snake_case naming style (11:8) [invalid-name]
    Argument name "Cif" doesn't conform to snake_case naming style (6:23) [invalid-name]
    Argument name "phOnE" doesn't conform to snake_case naming style (6:27) [invalid-name]
    Argument name "ENAME" doesn't conform to snake_case naming style (6:34) [invalid-name]
    Attribute name "ENTERPRISE_CIF" doesn't conform to snake_case naming style (17:4) [invalid-name]
    Attribute name "ENTERPRISE_CIF" doesn't conform to snake_case naming style (20:4) [invalid-name]
    Attribute name "PHONE_NUMBER" doesn't conform to snake_case naming style (24:4) [invalid-name]
    Attribute name "PHONE_NUMBER" doesn't conform to snake_case naming style (27:4) [invalid-name]
    Attribute name "ENTerprise_Name" doesn't conform to snake_case naming style (31:4) [invalid-name]
    Attribute name "ENTerprise_Name" doesn't conform to snake_case naming style (34:4) [invalid-name]
    Unused private member `EnterpriseRequest.__timeStamp` (11:8) [unused-private-member]
```

UC3MConsulting/EnterpriseRequest.py before fixing

```
Pylint found no problems
```

UC3MConsulting/EnterpriseRequest.py after fixing