

1) Frizar versión de Python en estado específico

Cuando ejecuto:

```
pip freeze > requirements.txt
```

Se crea un archivo que contiene todos los paquetes instalados en la versión de Python llamado requirements.txt, su contenido es el siguiente en el momento que lo ejecute:

```
Django==1.8.5  
Pillow==2.7.0  
ipython==2.3.1  
mysql-connector-python==2.0.3  
mysqlclient==1.3.6  
pl==1.0.0  
prueba1==1.0.0  
pyreadline==2.0
```

Para configurar el entorno de Python a un estado específico como el anterior se debe ejecutar:

```
pip install -r requirements.txt
```

2) Testing

2.1. Definiciones instalación

Test Unitario (Unit Test): Es un test que chequea una pequeña porción del código, como puede ser una función.

Integración de Test (Integration test): Es un test que chequea una gran porción de código, como puede ser un subsistema.

Test de Sistema (System test “end to end”): Es un test que chequea todo el sistema bajo análisis.

Test Funcional (Functional test): Es un test que chequea algunas funcionalidades de un sistema.

Test Subcutaneo (Subcutaneous test): -----

Instalar pytest

```
pip install pytest
```

2.2. Primer ejemplo

Ejemplo 1

operaciones.py

```
def sumar(a, b):  
    c = a + b  
    return c
```

Nota: Los archivos de test deben iniciar por test_

test_sumar.py

```
from operaciones import sumar  
  
def test_sumar():  
    assert sumar(1,2) == 7
```

Ejecutamos con:

```
pytest -rP
```

2.3. MARCAS (SON DECORADORES)

<https://docs.pytest.org/en/stable/how-to/mark.html>

Existen marcas específicas que permiten realizar acciones preestablecidas, por ejemplo skip permite omitir el test.

```
test_sumar_marca_1.py
```

```
from operaciones import sumar
import pytest

@pytest.mark.skip(reason="no way of currently
testing this")
def test_sumar():
    assert sumar(1,2) == 7
```

Ejecutamos con:

```
pytest -rP
```

- [usefixtures](#) - use fixtures on a test function or class
- [filterwarnings](#) - filter certain warnings of a test function
- [skip](#) - always skip a test function
- [skipif](#) - skip a test function if a certain condition is met
- [xfail](#) - produce an “expected failure” outcome if a certain condition is met
- [parametrize](#) - perform multiple calls to the same test function.

2.4. MARCAS personalizadas y archivo pytest.ini

pytest.ini = archivo de configuración

```
pytest.ini
# pytest.ini
[pytest]

markers =
    marca1: Primer marca
    marca2: Segunda marca
```

En el archivo de configuración puedo establecer mis marcas y luego indicarla como un decoradora sobre la rutina así:

```
from operaciones import sumar
import pytest

@pytest.mark.marca1
def test_sumar():
    assert sumar(1,2) == 7
```

Ejecutamos con:

```
pytest -m "marca2"
```

2.5. Multiples test

Es el archivo de configuración

Test_multiplicar_multiples.py

```
from operaciones import sumar
from operaciones import multiplicar
import pytest

"""
@pytest.mark.marca2
def test_sumar():
    assert multiplicar(1,2) == 7
"""

@pytest.mark.parametrize(
    "a, b, resultado",
    [
        (1, 2, 2),
        (sumar(1, 2), 2, 6),
        (5, 2, 10),
        (multiplicar(2, 2), 2, 8)
    ]
)

def test_multiplicar(a, b, resultado):
    assert multiplicar(a, b) == resultado
```

Lo ejecuto con

pytest -v

2.6. Fixture

Fixture permite preparar los datos a ser pasados a la prueba.

```
test_fixture1.py

from operaciones import sumar
import pytest

@pytest.fixture
def fixture_1():
    return 5

def test_sumar(fixture_1):
    variable = fixture_1
    assert sumar(variable,2) == 7
```

3) Crear ejecutable

`pip3 install pyinstaller`

En Windows

`pip install pywin32 --upgrade`

PASO CREO EJECUTABLE

Ingresa a donde está el archivo principal y ejecuto

```
pyinstaller --onefile --windowed --icon=icono.ico .\main.py
```

Obvio debe de estar el icono y el archivo principal llamarse main.py

OJO: SI TENGO REFERENCIADOS ARCHIVOS, BASES DE DATOS O IMÁGENES DEBO ESPECIFICARLO PARA QUE NO FALLE.

Debo copiar dentro de dist los archivos referenciados

Si sale el mensaje:

```
Microsoft Windows [Versión 10.0.19042.1348]
(c) Microsoft Corporation. Todos los derechos reservados.

C:\Users\usuario\Desktop\M2-u3-Ejercicio>pyinstaller --onefile --windowed --icon=icono.ico
.\main.py
The 'typing' package is an obsolete backport of a standard library package and is
incompatible with PyInstaller. Please `pip uninstall typing` then try again.
```

Hago lo que dice

```
C:\Users\usuario\Desktop\M2-u3-Ejercicio>pip uninstall typing
```