



amenaaUMCA / Taller\_Arquitectura\_Python\_Modular\_2025C3-G02

[Code](#) [Issues](#) [Pull requests](#) [Actions](#) [Projects](#) [Security](#) [Insights](#)

Proyecto clase 11

GPL-3.0 license

0 stars 6 forks 0 watching Branches Activity Tags

Public repository



amenaaUMCA Carga Inicial de proyecto Clase 11 72273a9 · 2 hours ago

.gitattributes	Initial commit	2 hours ago
1_Bibliotecas_Python.md	Carga Inicial de proyecto Clase 11	2 hours ago
2_Trabajar_Archivos.ipynb	Carga Inicial de proyecto Clase 11	2 hours ago
2_Trabajar_Archivos.md	<a href="#">Carga Inicial de proyecto Clase 11</a>	2 hours ago
3_Modulos_Paquetes_Reto.i...	Carga Inicial de proyecto Clase 11	2 hours ago
LICENSE	Initial commit	2 hours ago
README.md	Carga Inicial de proyecto Clase 11	2 hours ago

README GPL-3.0 license



## Taller de Arquitectura Python: De Script a Paquete Profesional



Profesor: Andrés Mena Abarca

Este repositorio contiene el material didáctico para el taller sobre **Modularización, Paquetes y Bibliotecas en Python**. El objetivo es que usted, como estudiante, **desarrolle activamente** una estructura de software robusta, aplicando buenas prácticas de código.

# Objetivo Pedagógico del Taller

El principal objetivo es guiarle progresivamente a través de los conceptos de la arquitectura de software en Python:

1. **Módulo:** Aprender a crear funciones puras y reutilizables, aplicando **Docstrings** y **Type Hinting**.
2. **Paquete:** Entender la estructura de carpetas y el papel fundamental del archivo `__init__.py` como **fachada** (API pública).
3. **Robustez:** Implementar **Manejo de Errores** (`ValueError`) y **Context Managers** (`with open...`) para código defensivo.

 **Advertencia:** Este taller NO le dará el código resuelto. Solo le proporcionará las plantillas y las **Banderas de Prueba** (`asserts`) que su código debe pasar para ser considerado correcto.

## Guía de Inicio

### 1. Requisitos

- Python 3.8+
- Jupyter Notebook / JupyterLab

### 2. Estructura Inicial del Proyecto

Al inicio, su carpeta debe lucir así. Los archivos ya tienen la firma de las funciones, pero **la lógica interna** está vacía o incompleta. Taller\_Arquitectura\_Python\_Modular/ └── análisis\_datos/ └── init.py

```
| └── carga_datos.py  
| └── estadísticas.py  
└── 3_Modulos_Paquetes_Reto.ipynb <- INICIE AQUÍ
```

### 3. Desarrollo del Taller (Instrucciones Clave)

1. **Inicie el Notebook:** `3_Modulos_Paquetes_Reto.ipynb`.
2. Las celdas de código le pedirán **completar la lógica** en los archivos `.py` utilizando el comando mágico `%writefile`.
3. Después de escribir su código, ejecute la siguiente celda etiquetada como " **Validación (Test Unitario)**".
4. Si la validación falla (lanzando un `AssertionError`), significa que su código es incorrecto. Debe **volver al archivo de código, corregir la lógica y re-ejecutar la celda de validación**.
5. Solo si todas las pruebas pasan, usted podrá avanzar a la siguiente sección.

El reto es completar los 6 Retos ( RETO 1 a RETO 6 ) guiado por las pruebas unitarias del Notebook.

---

## Releases

No releases published

---

## Packages

No packages published

---

## Languages

- Jupyter Notebook 100.0%