

Laboratorio: Creando un Paquete de Biblioteca en Python

Objetivo:

- Comprender la estructura y creación de paquetes y subpaquetes en Python.
- Aprender a organizar módulos y funciones dentro de un paquete.
- Practicar la definición y el uso de variables y funciones (públicas y "privadas").
- Familiarizarse con la importancia del archivo `__init__.py`.

Descripción:

En este laboratorio, los estudiantes crearán un paquete de Python llamado biblioteca que simulará la gestión de una biblioteca. El paquete incluirá subpaquetes y módulos para manejar diferentes aspectos de la biblioteca, como libros, usuarios y transacciones (alquiler, lectura en sala).

Instrucciones:

1. Estructura del Paquete:

- Crea una carpeta llamada biblioteca.
- Dentro de biblioteca, crea un archivo vacío llamado `__init__.py`. Esto convierte la carpeta en un paquete de Python.
- Crea dos subcarpetas dentro de biblioteca: libros y usuarios.
- Dentro de cada subcarpeta (libros y usuarios), crea un archivo vacío llamado `__init__.py`.
- Crea un subpaquete llamado transacciones, dentro del paquete principal biblioteca. Este sub paquete deberá contener dos archivos `.py` llamados `alquiler.py`, `lectura.py` y el obligatorio `__init__.py`

2. Módulo libros/libros.py:

- Crea un archivo llamado `libros.py` dentro de la carpeta libros.
- Define una variable (pública) llamada `lista_libros` que sea una lista de diccionarios, donde cada diccionario represente un libro (título, autor, ISBN).
- Crea una función (pública) llamada `buscar_libro(isbn)` que busque un libro por su ISBN en `lista_libros` y devuelva el libro (diccionario) o `None` si no se encuentra.
- Crea una función "privada", que comience su nombre con un guión bajo `"_"`. Esta función, `"_contar_libros"`, debe retornar el total de libros existentes dentro de la `"lista_libros"`

3. **Módulo usuarios/usuarios.py:**

- Crea un archivo llamado usuarios.py dentro de la carpeta usuarios.
- Define una variable (pública) llamada lista_usuarios que sea una lista de diccionarios, donde cada diccionario represente un usuario (nombre, ID, correo electrónico).
- Crea una función (pública) llamada buscar_usuario(id_usuario) que busque un usuario por su ID en lista_usuarios y devuelva el usuario (diccionario) o None si no se encuentra.

4. **Módulo transacciones/alquiler.py:**

- Crea un archivo llamado alquiler.py dentro de la carpeta transacciones.
- Crea una función llamada alquilar_libro(id_usuario, isbn_libro) esta función registrará en un diccionario el alquiler del libro, deberá regresar un mensaje indicando si el libro fue alquilado por el usuario con el ID especificado.

5. **Módulo transacciones/lectura.py:**

- Crea un archivo llamado lectura.py dentro de la carpeta transacciones.
- Crea una función llamada libro_leido(id_usuario, isbn_libro) esta función registrará en un diccionario que el usuario leyo tal libro en las instalaciones de la biblioteca. Deberá retornar un mensaje especificando que el usuario leyo el libro.

6. **Importación y Prueba:**

- Crea un archivo Python fuera del paquete biblioteca para probar los módulos.
- Importa las funciones y variables necesarias desde los módulos del paquete biblioteca.
- Ejecuta algunas pruebas para verificar que las funciones y variables funcionen correctamente.
- Imprime el valor retornado de la función privada "_contar_libros", para determinar que el llamado no es posible.

Este laboratorio proporcionará a tus estudiantes una experiencia práctica en la creación y organización de paquetes en Python.