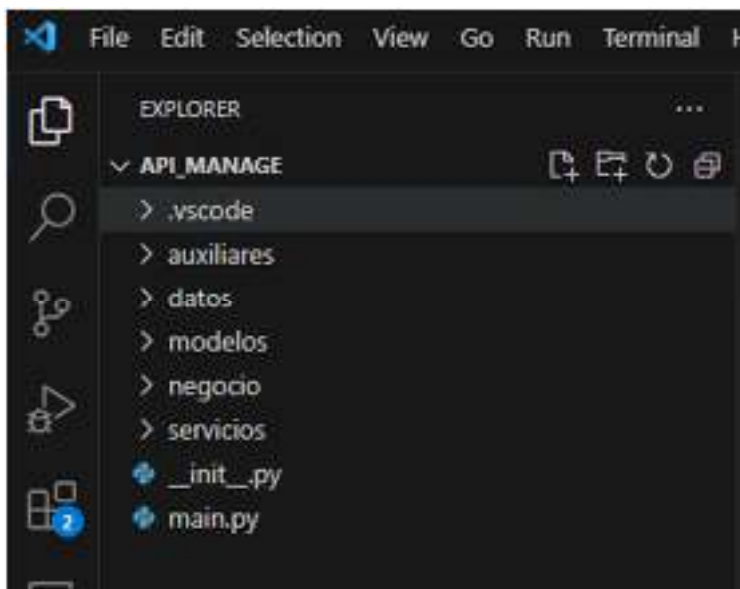


En base a la actividad programada para evaluar la unidad 3 de la asignatura POOS, debo entregar algunas aclaraciones para su correcto y completo cumplimiento de la actividad.

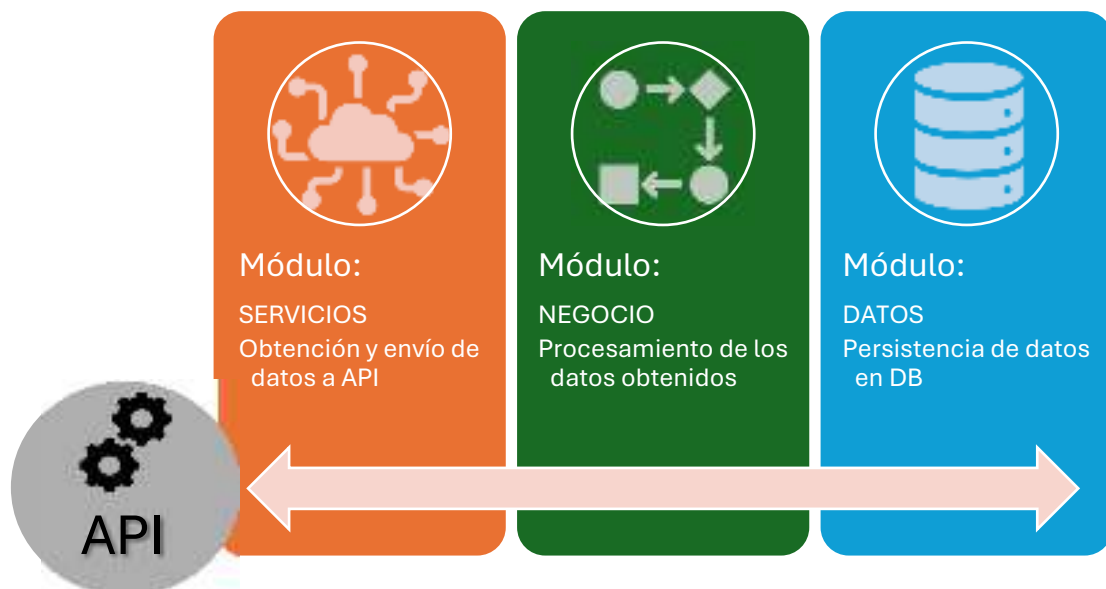
1. Como la actividad está programada para trabajar en grupos de 2 o 3 personas. Uds. Deberán conformar los grupos y enviar esa información mediante correo electrónico a erick.bailey@inacapmail.cl.
2. Teniendo los grupos definidos, la idea es que todos trabajen sobre un mismo repositorio, por lo que un integrante del grupo será el encargado de crear un repositorio e incorporar a sus compañeros para compartir el trabajo y que pueda cada uno escribir su código y luego publicarlo en la rama correspondiente del repositorio. Para hacer esto, necesitan tener el nombre de usuario del colaborador en github y escribirlo en el cuadro de búsqueda de la barra de navegación superior. Al encontrarse el usuario, podrá seleccionarlo y github le enviará una solicitud de colaboración al correo personal del invitado. Una vez invitado, podrá manejar incluso los privilegios y dar o restringir acceso a lectura y escritura.
3. Una vez completados estos pasos se deberá crear la estructura del proyecto, respetando los conceptos de modularización. No debe olvidar incluir en cada directorio/módulo un archivo `__init__.py` para indicarle a Python que ese directorio es un módulo y debe ser usado como tal. La estructura que se debe generar será como se indica a continuación:



- I. **AUXILIARES:** Como se ha venido haciendo, datos que no cambiarán a menudo, listas y constantes serán los que se

contendrán en este módulo, para ser llamados cuando sea necesario.

- II. **DATOS:** El módulo de acceso a datos es el que contiene su conexión a DB y los scripts de DB. En esta unidad Uds. Deberán trabajar con los datos entregados por la API <https://jsonplaceholder.typicode.com> y deberá respaldarlos en su DB. Hay 3 grupos de datos para usar, publicaciones con comentarios (posts y comments), álbumes con fotos (albums y photos), y finalmente, tareas por realizar y usuarios que deben realizar esas tareas (todos y users). Su DB deberá reflejar estas entidades (fíjense muy bien en los objetos JSON que devuelve el servicio) y deberá guardar los datos obtenidos desde la API mediante procesos CRUD.
- III. **MODELOS:** Habiendo definido con qué grupo de datos trabajarán desde jsonplaceholder, se debe generar las clases que permitirán trabajar con esos objetos. Esas clases estarán contenidas en este módulo.
- IV. **NEGOCIO:** Como se ha indicado anteriormente, todo lo que sea procesamiento de datos tiene que ver con reglas de negocio del cliente (dueño del sistema), por lo que queda reflejado en este módulo. El procesamiento de datos tiene que ver con los datos obtenidos desde la API. Estos datos se procesan para luego ser enviados al módulo de DB para persistirlos (guardarlos) en DB.

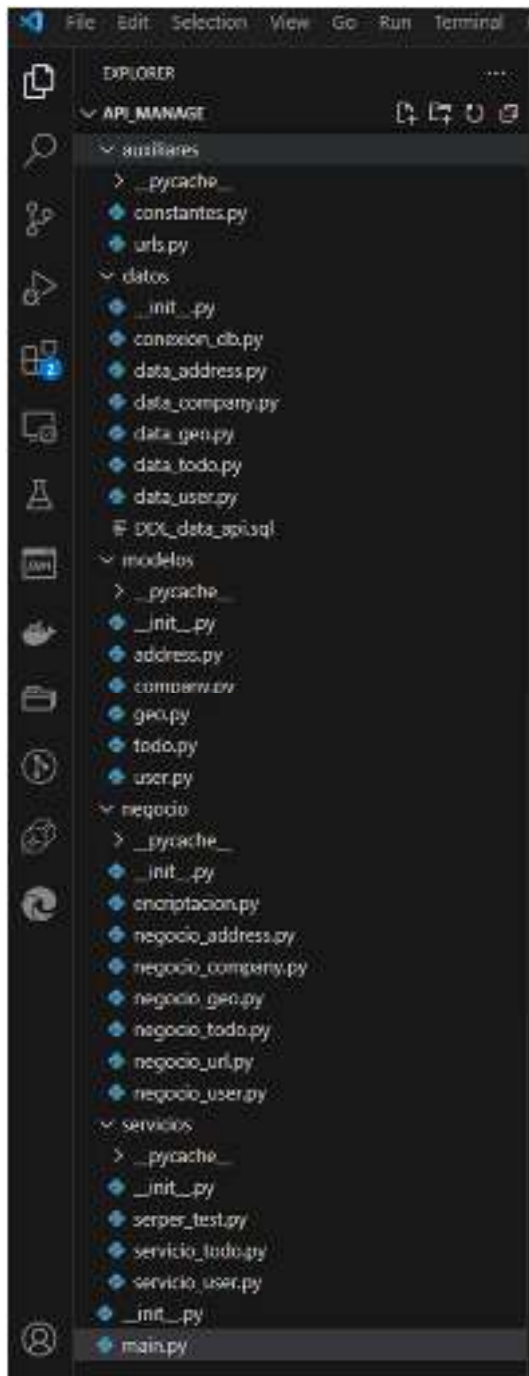


Además, como en la unidad 2 no se incluía la encriptación y siendo esta un tipo de procesamiento de datos (contraseñas) se incluirá en este módulo una clase que encripte un string y luego lo

desencripte para compararlo con el string original y entregar una respuesta/resultado en caso de que ambas sean iguales.

- V. **SERVICIOS:** En esta capa se encontrará toda la comunicación con APIs o servicios externos. Conexiones, solicitudes y respuestas, los que luego serán procesadas en los siguientes módulos, como ya se ha indicado.

Dependiendo de las clases que vaya a trabajar desde jsonplaceholder, su estructura de proyecto debería quedar aproximada a la siguiente:



Finalmente, todo debería ser ejecutado desde un menú principal en la clase main.py.

Los procesos que se deben incluir en este menú serán:

1. Encriptación de contraseña: El sistema debe solicitar al usuario el string de contraseña para encriptar. Se deberá mostrar el string ingresado, luego se deberá encriptar y mostrar el resultado de la contraseña encriptada, finalmente se deberá desencriptar y comparar con el string ingresado originalmente, entregando una respuesta acorde.
2. Solicitar la obtención de datos desde la API. Se deberá indicar que datos se capturarán desde la API de jsonplaceholder. Los datos obtenidos se almacenarán en DB local y luego se hará una consulta a DB para ver los datos insertados. Esto cubre la solicitud HTTP GET.
3. Para la solicitud HTTP POST se hará un envío de data de acuerdo a los objetos incluidos en la API. Se deberá solicitar al usuario que ingrese los datos requeridos para crear un objeto del tipo indicado y este debe ser enviado mediante POST a la API que debe retornar una respuesta 200, indicando que todo se hizo correctamente.
4. Para trabajar con una solicitud HTTP con autenticación se usará la API Serper para realizar una búsqueda con la API de Google, por lo que se deberá solicitar el string de búsqueda al usuario y generar la solicitud HTTP con esa data y el header con su autenticación correspondiente. Este proceso debe retornar los resultados de la búsqueda.

Con todo lo anterior, espero un excelente resultado para todos Uds.

Nos vemos en la evaluación N° 3.