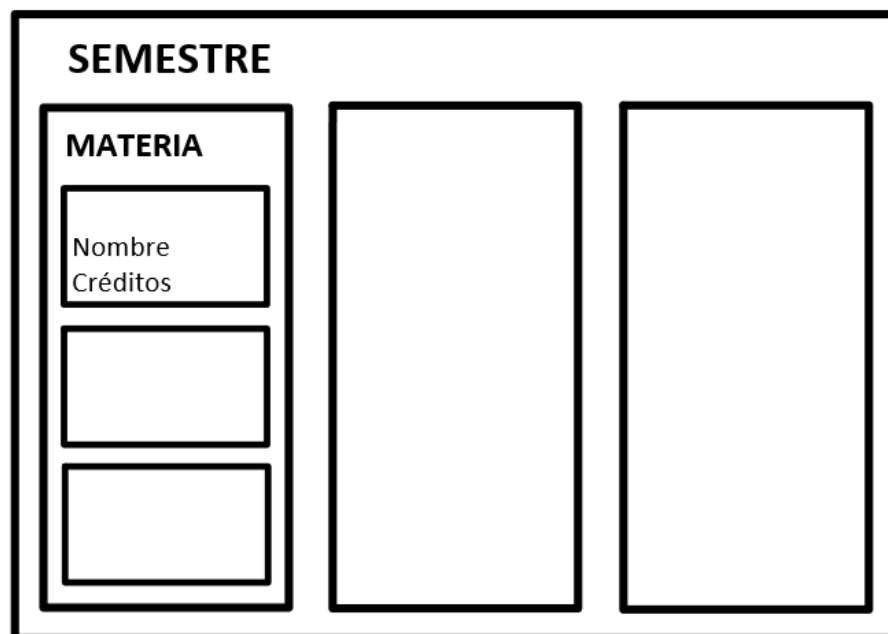




## RETO 4 – Fundamentos de Programación

La Universidad de software AllStacks necesita un sistema de información para gestionar los pensums o currículos de los programas que ofrece, ya que este será la base sobre la que se construirá un el sistema de matrículas. Usted hace parte del equipo de desarrollo y su Jefe le encargó implementar la funcionalidad de modificar y eliminar una materia de un pensum, además, le dió la especificación de la definición de las estructuras de datos.

### PENSUM



Dado que un pensum tiene  $n$  semestres y estos son ordinales, es decir, primero está el semestre 1, luego está el semestre 2 y así, este es representado por una lista donde cada posición será un semestre, este no puede ser una tupla dado que necesitamos modificarlo.

Cada semestre es, a su vez, una colección, esta vez de materias, estas tienen un código que las permite identificar y no son ordinales, por lo tanto los semestres serán representados como diccionarios donde las claves serán los códigos de las materias y en los valores estarán las materias.

Por último tenemos las materias, dado que necesitamos almacenar el nombre y cantidad de créditos de las mismas, también usaremos diccionarios.





En la gráfica anterior se puede apreciar la estructura anidada

A continuación se le muestra un ejemplo de un pensum de un programa de 3 semestres que solo se le añadieron materias en el primer semestre:

```
pensum = [  
    {'0123': {'nombre': 'intro a la ing', 'créditos': 2},  
    '4567': {'nombre': 'inglés', 'créditos': 1}},  
    {}, {}  
]
```

Como se mencionó, solo 2 materias fueron añadidas al primer semestre, la primera tiene código: 0123, nombre: intro a la ing y créditos 2, la segunda tiene código: 4567, nombre: inglés y créditos 1.

Dado un pensum como el ya descrito, para implementar las funcionalidades de *modificar\_materia(pensum, semestre, materia, nombre, credits)* y *eliminar\_materia(pensum, semestre, materia)*, debe:

- Validar que el número del semestre (int) donde está la materia que el usuario desea modificar/eliminar si corresponde a un semestre del pensum dado.
- Validar que el semestre que se desea modificar/eliminar no esté vacío.
- Validar que el código de la materia a modificar/eliminar esté en el semestre seleccionado por el usuario.

**NOTA:** Recuerde que el usuario se referirá al primer semestre o semestre 1 y este en realidad se encuentra en la posición con índice cero en el pensum, usted debe tener esto en cuenta para convertir la entrada del usuario apropiadamente.

Luego de las validaciones, en el caso de *modificar\_materia*, puede proceder a modificar el nombre y créditos de la materia seleccionada por los valores proporcionados por el usuario, en el caso de *eliminar\_materia*, puede proceder a eliminar la materia que coincide con el código proporcionado por el usuario .

Como puede ver ambas funcionalidades necesitan las mismas validaciones, para esto se pueden crear funciones para cada validación lo que le permitirá hacer uso de mejores prácticas. Las funciones serán las siguientes:





`es_semestre_valido(pensum, semestre)` retornará True si el semestre es válido según la condición ya expuesta, False de lo contrario.

`es_semestre_vacio(pensum, semestre)` retornará True si el semestre es vacío, False de lo contrario.

`es_materia_valida(pensum, semestre, materia)` retornará True si es la materia es válida según la condición ya expuesta, False de lo contrario.

En resumen usted debe implementar las siguientes funciones teniendo en cuenta sus entradas y salidas (retornos):

- `modificar_materia(pensum, semestre, materia, nombre, creditos)` **NO retorna.**
- `eliminar_materia(pensum, semestre, materia)` **NO retorna.**
- `es_semestre_valido(pensum, semestre)` **retorno booleano.**
- `es_semestre_vacio(pensum, semestre)` **retorno booleano.**
- `es_materia_valida(pensum, semestre, materia)` **retorno booleano.**

## EJEMPLO

Suponiendo que se tiene el pensum de tres semestres presentado anteriormente:

`pensum`

```
[{'0123': {'créditos': 2, 'nombre': 'intro a la ing'},  
  '4567': {'créditos': 1, 'nombre': 'inglés'}},  
 {},  
 {}]
```

```
modificar_materia(pensum, 4, '2345', 'lectoescritura', 3)  
pensum
```

```
[{'0123': {'créditos': 2, 'nombre': 'intro a la ing'},  
  '4567': {'créditos': 1, 'nombre': 'inglés'}},  
 {},  
 {}]
```





```
modificar_materia(pensum, 0, '2345', 'lectoescritura', 3)  
pensum
```

```
[{'0123': {'créditos': 2, 'nombre': 'intro a la ing'},  
  '4567': {'créditos': 1, 'nombre': 'inglés'}},  
{},  
{}]
```

Note que los semestres 4 y 0 no existen, por lo tanto, la operación no es efectuada, el pensum solo tiene 3 semestres, para ingresar un semestre válido, el usuario tendría que ingresar un número entero entre 1 y 3, ambos incluidos.

```
modificar_materia(pensum, 2, '2345', 'lectoescritura', 3)  
pensum
```

```
[{'0123': {'créditos': 2, 'nombre': 'intro a la ing'},  
  '4567': {'créditos': 1, 'nombre': 'inglés'}},  
{},  
{}]
```

Aunque 2 es un semestre válido, este no tiene materias, por lo tanto no se puede realizar la modificación satisfactoriamente.

```
modificar_materia(pensum, 1, '0123', 'lectoescritura', 3)  
pensum
```

```
[{'0123': {'créditos': 3, 'nombre': 'lectoescritura'},  
  '4567': {'créditos': 1, 'nombre': 'inglés'}},  
{},  
{}]
```

El semestre 1 es válido, no vacío, y existe una materia con el código 0123, por lo tanto se realiza la modificación satisfactoriamente.





```
eliminar_materia(pensum, 1, '2345')  
pensum
```

```
[{'0123': {'créditos': 3, 'nombre': 'lectoescritura'},  
  '4567': {'créditos': 1, 'nombre': 'inglés'}},  
 {},  
 {}]
```

Aunque 1 es un semestre válido y con materias, en este semestre no hay una materia con el código 2345, por lo tanto no se puede realizar la eliminación satisfactoriamente.

```
eliminar_materia(pensum, 1, '0123')  
pensum
```

```
[{'4567': {'créditos': 1, 'nombre': 'inglés'}}, {}, {}]
```

Nuevamente el semestre 1 es válido, no vacío, y existe una materia con el código 0123, por lo tanto se realiza la eliminación satisfactoriamente.





## NOTA ACLARATORIA

Se recomienda desarrollar la prueba en un IDE como G Colab, VSCode, PyCharm, Spyder, etc. Para esto se puede copiar y pegar el esquema de solución proporcionado en el VPL a su IDE preferido, recuerde que al final debe copiar y pegar el código del IDE a la herramienta VPL, pero **NO** deberá subir archivos, es decir:

### Modo incorrecto:

**Examen caracterización-estudiantes**

**NO SUBIR NINGÚN ARCHIVO**

Descripción Entrega **Editar** Ver entrega

Entrega

Comentarios

Seleccione un archivo... Tamaño máximo para archivos nuevos: 5MB

solucion.py

Puede arrastrar y soltar archivos aquí para añadirlos

Enviar Cancelar

### Modo correcto:

**Examen caracterización-estudiantes**

Descripción Entrega **Editar** Ver entrega

**LUGAR CORRECTO**

```
1 #NO ELIMINAR LAS SIGUIENTES IMPORTACIONES, sirven para probar tu código en consola, y el funcionamiento de la librería csv respectivamente
2 from test import tester
3 import csv
4
5 """NOTAS:
6 - PARA ESTE RETO PUEDES PROBAR TU PROGRAMA, DANDO CLICK EN LA NAVE ESPACIAL
7 - LA CONSOLA TE DIRÁ SI TU SOLUCIÓN ES CORRECTA O NO
8 - NO olvidar evaluar tu solución
9 """
10
11
12 """Inicio espacio para programar funciones propias"""
13 #En este espacio podrás programar las funciones que deseas usar en la función solución (ES OPCIONAL)
14
15
16
```

TRIPULANTE, ¡MUCHOS ÉXITOS EN EL DESARROLLO DEL RETO 4!

