

## TP - POO en Python

### 1. Codifique la clase celda con los atributos:

```
fila; //entero
columna; //entero
valor; //cadena
```

- Crea una clase Matriz que contenga una variable celdas

```
celdasMatriz = [];
```

- Codifique un programa que solicite al usuario un valor para la celda y que solicite la posición donde se desea almacenar el valor, cree una instancia de la clase Celda, asigne los valores cargados por el usuario (fila, columna y valor) y agregue la instancia a la lista celdasMatriz; repita este proceso hasta que el usuario ingrese como valor la cadena "FIN". Valide que la celda creada ya no exista anteriormente es decir si la fila y columna indicados ya fueron cargados en celdasMatriz.
- Muestre por pantalla los valores cargados en la lista celdas.
- Codifique un método que reciba como parámetro los valores fila y columna y retorne el valor almacenado en la Celda correspondiente, en caso de que la fila y la columna no exista retorne el mensaje "La fila y columna indicada no ha sido asignada en ninguna celda"

### 2. Codifique las siguientes clases Python en archivos separados

**Clase:** Nota

**Atributos:** catedra (cadena), notaExamen(decimal)

**Clase:** Alumno

**Atributos:** nombreCompleto(cadena), legajo (entero), lista de objetos notas[]

**Clase:** CargaNotas (clase que tendrá declarado el método main para ejecutar el código.)

Lógica a Implementar:

En la clase CargaNotas debera declarar una variable que permita contener un conjunto de Alumnos:

Ejemplo:

```
alumnos = []
```

El algoritmo debe permitir cargar N cantidad de alumnos y para cada alumno N cantidad de Notas. Al finalizar la carga de los alumnos y sus notas mostrar la información cargada y para cada alumno mostrar el promedio de las notas que posee. Valide que se ingrese al menos 1 nota. Agregue en la clase Alumno un método que calcule el promedio de las notas que posee.

### Posible Pseudocódigo básico:

Definir alumnos como Lista

#### Repetir

- Crear instancia alumno

- Escribir "INGRESE DATOS DEL ALUMNO"

- Escribir "INGRESE NOMBRE COMPLETO"

- Leer nombreCompleto

- Escribir "INGRESE LEGAJO"

- Leer legajo

#### Repetir

- Definir nota como Nota

- Escribir "INGRESE NOMBRE CATEDRA"

- Leer nombreCatedra

- Escribir "Nota"

- Leer notaExamen

- Escribir "DESEA SALIR DE LA CARGA DE NOTAS"

- Cargar nota en Lista de Notas

- Leer salirNotas

#### Hasta Que salirNotas

- Cargar alumno en Lista alumnos

- Escribir "DESEA SALIR DE CARGA DE ALUMNOS"

- Leer salirAlumno

#### Hasta Que salirAlumno

#### Para alumno en alumnos Hacer

- Escribir "Datos Alumno", .....

#### Para nota en notas Hacer

- Escribir "Nota" .....

#### Fin Para

- Escribir "El promedio del alumno es: " promedio

#### Fin Para

**3.** Codifique las siguientes clases Python en archivos diferentes

**Clase:** Ingrediente

**Atributos:** nombre (cadena), cantidad(decimal), unidad de medida (cadena)

**Clase:** Plato

**Atributos:** nombreCompleto(cadena), precio (decimal), esBebida (boolean), lista de ingredientes[]

**Clase:** MenuRestaurant (clase que tendrá declarado el método main para ejecutar el código.)

Lógica a Implementar:

En la clase MenuRestaurant debera declarar una variable “platosMenu[]” que permita contener un conjunto de Platos que componen el menú de un restaurant.

**Ejemplo:**

platos = [];

El algoritmo debe permitir cargar N platos y para cada plato indicar los N ingredientes que lo componen. Si el plato es de tipo Bebida entonces no se deben solicitar los ingredientes, en cambio si no lo es, será obligatorio que se asigne al menos 1 ingrediente.

Al finalizar la carga de los platos y sus ingredientes mostrar la información cargada, la cual será equivalente al menú del restaurant.

**Descripción del algoritmo:**

Solicito los datos del plato (Ejemplo Pizza Especial, 450 pesos) y sus ingredientes (harina 100 gramos, huevos 2 unidades, queso 300 gramos, jamón 100 gramos, etc) o si es una bebida solo pido el nombre y el precio.

Almaceno el plato anterior en la lista de platos “platosMenu[]”, repito este proceso para tantos platos como desee. Al finalizar muestro la información respetando el siguiente formato ejemplo

-----MENÚ-----

Pizza Especial

Precio: \$ 450

Ingredientes:

Nombre	Cantidad	Unidad de Medida
Queso Muzza	300	gramos
Jamon Cocido	100	gramos

-----  
Vino Tinto Elementos

Precio: \$ 300  
-----

Empanadas Criollas

Precio: \$ 500

Ingredientes:

Nombre	Cantidad	Unidad de Medida
Picadillo	150	gramos
Tapa Empanada	1	unidad

-----

.....repetir.....para el resto de los platos

4. Codifique las siguientes clases Python en archivos diferentes

**Clase: ComponenteCPU**

**Atributos:** componente (cadena), marca (cadena), cantidad (entero), precio (decimal)

**Clase: Computadora**

**Atributos:** marca(cadena), modelo (cadena), lista de objetos componenteCPU []

**Clase: CostoComputadora** (clase que tendrá declarado el método main para ejecutar el código.)

Lógica a Implementar:

El algoritmo debe permitir cargar la marca y el modelo de una computadora y para cada computadora indicar los N componentes de CPU que posee indicando la cantidad y el precio por ejemplo componente "Memoria RAM 2 Gb", marca "Kingston", cantidad "2", precio "8000"

Al finalizar la carga de los componentes mostrar la información cargada, y determinar el costo de la computadora el cual será el equivalente de sumar el precio por la cantidad usada de cada componente y el precio de venta sugerido para la computadora el cual será equivalente a el precio de costo más el 40% del precio de costo si el precio de costo es menor a 50000, o equivalente a el precio de costo más el 30% del precio de costo si el precio de costo es mayor a 50000.

Al terminar de mostrar los datos y precio de la computadora deberá preguntar si desea cotizar una nueva computadora, si la respuesta es "SI" deberá iniciar nuevamente el programa, si la respuesta es "NO", terminar la ejecución del programa.

-----Computadora-----

**Marca:** Lenovo

**Modelo:** X64K

**Componentes:**

Componente	Marca	Cantidad	Precio X Unidad	SubTotal
Placa Madre	ASUS	1	16000	16000
Procesador	INTEL I7	1	20000	20000
Memoria RAM 2Gb	Kingston	2	8000	16000
Disco Rigido	WD SSD	1	7500	7500
<b>Costo Total</b>				59500

El precio sugerido de venta es  $59500 + 17850 = 77350$

5. Codifique las siguientes clases Python en archivos diferentes

**Clase: Barrio**

**Atributos:** nombre (cadena), empresaConstructora (cadena), lista de objetos Vivienda[]

**Clase: Vivienda**

**Atributos:** calle (cadena), numero (entero), manzana (cadena), nroCasa (entero), superficieTerreno (decimal), lista de objetos Habitacion []

**Clase: Habitacion**

**Atributos:** nombre (cadena), metrosCuadrados (decimal)

Codifique los siguientes métodos en las Clases indicadas

a) Codifique en la clase **Barrio** un método llamado

**getSuperficieTotalTerreno()**

que retorne el total de metros de terreno del barrio teniendo en cuenta la totalidad de viviendas asociadas al mismo. Es decir deberá sumar el atributo superficieTerreno de las viviendas asociadas.

b) Codifique en la clase **Barrio** un método llamado

**getSuperficieTotalTerrenoXManzana(manzana)**

que retorne el total de metros de terreno de una manzana especifica del barrio teniendo en cuenta la totalidad de viviendas de esa manzana. Es decir deberá sumar el atributo superficieTerreno de las viviendas asociadas filtrando por su manzana.

c) Codifique en la clase **Vivienda** un método denominado

**getMetrosCuadradosCubiertos()**

que retorne el total de metros cuadrados de la vivienda teniendo en cuenta la cantidad de habitaciones asociadas y sus respectivos metros cuadrados. Al finalizar el cálculo valide que el valor obtenido no sea mayor que la superficie del terreno, si ocurre esa situación emita una excepción con el mensaje "La superficie cubierta no puede ser mayor a la superficie del terreno"

- d) Codifique en la clase **Barrio** un método llamado **getSuperficieTotalCubierta()** que retorne los metros cuadrados cubiertos del barrio sumando la totalidad de metros cuadrados cubiertos de las viviendas que componen al barrio. Reutilice el método `getMetrosCuadradosCubiertos()` del punto anterior.
- e) Codifique una clase donde cree los objetos Barrio, Vivienda y Habitaciones, ejecute las asociaciones correspondientes y muestre por pantalla el resultante de la ejecución de los métodos codificados en los métodos anteriores.