UNIVERSIDAD ESTATAL DE MILAGRO
UNEMI
Evolución Académica

SÍGUENOS

Formato para Recurso de Aprendizaje

TAREA

(04) 2 715081 - 2 715079
Cdla. Universitaria Km 1.5 vía al km 26
www.unemi.edu.ec

Nombre de la Asignatura	*ESTRUCTURA DE DATOS*	
Unidad N°	**1-2**	*ESTRUCTURA DE DATOS*
Tema N°	**T**	*ESTRUCTURA DE DATOS*

Tipo de Tarea

APLICACIÓN PRACTICA "Desarrollo de un menú de clases".

Objetivo de la Tarea

Desarrollar un menú de cuatro clases de ejercicios de programación

INSTRUCCIONES

Bienvenido(a), a la actividad de aprendizaje practico **No 4:** de gestión en el aula.

Desarrollo de un menú de cuatro clases de ejercicios de programación:

1.2. Instrucciones:

a) El profesor, coordinará la organización de los equipos de trabajo colaborativo dentro del aula de clases, máximo 3 estudiantes por equipo.

b) Subir como evidencia en github.

c) La entrega de trabajo grupal es individual, y su calificación será grupal, deberá describir los integrantes de grupo en el trabajo compartido.

Los datos a considerar están considerados sobre la siguiente realidad del problema:
Menu. Es una clase de 5 opciones:

Menu Principal

1). Calculadora

2). Operación Numeros

3). Tratamiento de Listas

4). Operaciones de Cadenas

5). Salir

1) **Calculadora**. Esta opción será un submenú de 10 opciones:

   Menu Calculadora

   1) Suma
   2) Resta
   3) Multiplicacion
   4) Division
   5) Exponente
   6) Valor Absoluto
   7) Circunferencia
   8) Area Circulo
   9) Area Cuadrado
   10) Salir

A continuación se muestra la estructura de las clases del submenú Calculadora

```
class Calculadora

    def __init__(self, numero1, numero2)

    def suma()

    def resta()

    def mutiplicacion()

    def división()

class calEstandar(Calculadora):

    def __init__(self, numero1, numero2)

        super().__init__()

    def mutiplicacion() # aplicar polimorfismo

    def exponente()

    def valorAbsoluto(numero)
```

class calCientifica(Calculadora):

PI = 3.1416

def __init__(self, numero1, numero2)

super().__init__(numero1,numero2)

def circunferencia(radio)

def areaCirculo(radio)

def areaCuadrado(lado)

2) Operación Numeros. Esta opción será un submenú de 11 opciones:

Menu Operación Numeros

1) Presentar los números de 1 a n
2) Sumar los números de 1 a n
3) Múltiplo de cualquier numero
4) Presentar los divisores de un numero
5) Numero Primo
6) Factorial de cualquier numero
7) Fibonacci de una serie de n números
8) Perfecto
9) Primos gemelos
10) Números amigos
11) Salir

A continuación se muestra la estructura de las clases del submenú Operación Números

class Basico:

def numerosN(n)

def multiplo(numero)

def DivisoresNumero(numero)

def primo(numero)

def perfecto(numero)

```
class Intermedio(Basico):

    def numerosN(n) # aplicar polimorfismo

    def factorial(numero) #

    def fibonacci(n)

    def primosGemelos(num1,num2)

    def  amigos(num1,num2)
```

3) Tratamiento Listas. Esta opción será un submenú de 11 opciones:

Menu Tratamiento Listas

1) Recorrer y presentar los datos de una lista
2) Buscar un valor en una lista
3) Retornar una lista con los factoriales
4) Retornar una lista de números primos
5) Recorrer una lista de diccionario con notas de alumnos
6) Insertar un dato en una Lista dada lo Posición
7) Eliminar todas las ocurrencias en una Lista
8) Retornar cualquier valor de una lista eliminándolo
9) Copiar cada elemento de una tupla en una lista
10) Dar el vuelto a varios clientes
11) Salir

A continuación, se muestra la estructura de la clase del submenú Tratamiento Lista

```
class Lista(Intermedio):

    def __init__(self, lista)

    def  presentarLista()

    def  buscarLista(valor)

    def  listaFactorial()

    def listaPrimo()

    def listaNotas(listaNotasDicccionario)

    def insertarLista(valor,posicion)
```

def eliminarLista(valor)

def retornaValorLista(posicion)

def copiarTuplaLista(tupla)

def vueltoLista(listaClientesDiccionario)

4) **Operaciones de Cadenas**. Esta opción será un submenú de 11 opciones:

Menu Operaciones de Cadenas

1) Recorrer y presentar los datos de una cadena
2) Buscar un carácter en una cadena
3) Retornar una lista con la posiciones dado un carácter de la cadena
4) Retornar una lista con todas las palabras de una cadena
5) Retornar una cadena a partir de una lista
6) Insertar un dato en una cadena dada lo Posición
7) Eliminar todas las ocurrencias en una cadena
8) Retornar cualquier valor de una cadena eliminándolo
9) Concatenar cadenas
10) Salir

A continuación, se muestra la estructura de la clase del submenú Operaciones de Cadenas

class Cadena:

def __init__(self, cadena)

def recorrerCadena()

def buscarCaracter(buscado)

def listaPosiciones(caracter)

def listaPalabras()

def cadenaLista()

def insertarDato(buscado,posicion)

def eliminarOcurrencias(buscado)

def retornaValor(posicion)

def concatenarCadena(dato)

Nota: cualquier duda sobre tarea indicarlo en la clase de zoom

RECOMENDACIONES

- Recuerda que el trabajo será validado mediante preguntas de como ha realizado el desarrollo.
- Revise el material de diapositivas y ejercicios realizados en el curso.
- Antes de enviar el trabajo, tome en consideración estos aspectos:
  - ✓ Revise la gramática y ortografía.
  - ✓ Organice las ideas que vaya a utilizar.
  - ✓ Tome en cuenta los parámetros y tiempos establecidos.
  - ✓ Revise la rúbrica de evaluación.

Daniel Vera Paredes, Msc.
DOCENTE