

DEFENSA HITO 2 - TAREA FINAL

Estudiante: Univ. Brayan Erwin Honorio Rojas
Asignatura: Programación de Sistemas Embebidos
Carrera: INGENIERÍA DE SISTEMAS
Paralelo: PSE (1)
Docente: Lic. William Barra Paredes
fecha: 05/04/2022
Link video:
<https://drive.google.com/file/d/1DHIE5CbwjuWAJzKH0FvvjwDefURz40mj/view?usp=sharing>



Manejo de conceptos.

¿Qué es un sistema embebido?

son sistemas operativos creados con el fin de ser controlados por microprocesadores o micro controladores, de igual manera a los sistemas normales, pero llevados a un fin completamente sistematizado y sin llevar a tantas tareas, son mejor dicho, sistemas que cumplen con una tarea en específico.

¿Mencione 5 ejemplos de sistemas embebidos?

Tal como ya mencionamos anteriormente, el sistema embebido es un procesador que se especializa en una función específica, es por ello que uno de sus ejemplos:

- Dispositivos electrónicos.
- Elementos compactos.
- Dispositivos autónomos.
- Computadoras y ordenadores.
- Procesadores de datos.

¿Menciona las diferencias o similitudes entre un sistema operativo, un sistema móvil y un sistema embebido?

La diferencia entre un sistema operativo móvil y un sistema operativo de computadora tiene que ver con cómo las compañías tecnológicas individuales han implementado varias versiones de los sistemas operativos que proporcionan los entornos fundamentales para las aplicaciones de software tradicionales, así como las nuevas aplicaciones móviles.

Los sistemas operativos móviles y de computadora se han desarrollado de diferentes maneras y para diferentes usos. Los productos del sistema operativo son más antiguos y más familiares para grupos más grandes de usuarios.

En cambio los sistemas embebidos son sistemas electrónicos que emplean hardware y software de un microcomputador para desarrollar.

¿A que se referirán los términos MCU y MPU?
Explique cada una de ellas.

MCU.-Un micro controlador (MCU) es un IC que incluye una CPU, memoria y circuitos. Entre los subsistemas que incluyen los micro controladores se encuentran los temporizadores, los convertidores analógico a digital (ADC) y digital a analógico (DAC) y los canales de comunicaciones serie.

Es un microordenador comprimido que se utiliza para controlar las funciones del sistema integrado en robots, máquinas de oficina, vehículos de motor, electrodomésticos y otros dispositivos electrónicos.

Ejemplos de componentes:

- procesador
- periféricos
- memoria

MPU.- Según se fue desarrollando la tecnología los diferentes componentes se fueron fusionando dentro del mismo circuito. Así se pasó de tener un procesador formado por muchos circuitos integrados interconectados, a tener lo que denominamos microprocesador, que incorporaba todos estos elementos en un único circuito integrado. los microprocesadores no se utilizan de manera aislada. No tienen uso por sí mismos, sino que se integran dentro de otros sistemas que les dan un uso concreto, como por ejemplo los micro controladores.

Ejemplos de componentes:

- Memoria caché. ...
- Coprocesador matemático



¿Cuáles son los pilares de POO?

Abstracción

Es cuando separamos los datos de un objeto para luego generar un molde (una clase).

Herencia

Nos permite crear nuevas clases a partir de otras. Esto nos da una jerarquía de padre e hijo.

Encapsulamiento

Lo puedes utilizar cuando deseas que ciertos métodos o propiedades sean inviolables o inalterables.

Polimorfismo

Proviene de Poli = muchas, morfismo = formas. Se utiliza para crear métodos con el mismo nombre pero con diferente comportamiento.

¿Mencione los componentes en lo que se basa POO?. Y explicar cada una de ellas.

1. Clases: Las clases pueden ser definidas como un molde que contendrá todas las características y acciones con las cuales podemos construir N cantidad de objetos.
2. Propiedades: Las propiedades son las características de una clase, tomando como ejemplo la clase humanos, las propiedades podrían ser: nombre, el género, la altura, color de cabello, color de piel, etc.
3. Métodos: Los métodos son las acciones que una clase puede realizar, siguiendo el mismo ejemplo anterior, estas podrían ser: caminar, comer, dormir, soñar, respirar, nadar, etc.
4. Objetos: Son aquellos que tienen propiedades y comportamientos, estos pueden ser físicos o conceptuales.



Defina los siguientes:

Multiplataforma

Se puede dividir en dos tipos; uno requiere compilación o compilación individual para cada plataforma que admite, y el otro se puede ejecutar directamente en cualquier plataforma sin preparación especial, por ejemplo, software escrito en un lenguaje interpretado o bytecode portátil precompilado para el cual los intérpretes o el tiempo de ejecución los paquetes son componentes comunes o estándar de todas las plataformas.

Multiparadigma

Es el lenguaje que permite a los programadores utilizar el mejor paradigma para cada trabajo, admitiendo que ninguno resuelve todos los problemas de la forma más fácil y eficiente posible. Por ejemplo, lenguajes de programación como C++, Genie, Delphi, Visual Basic o PHP, combinan el Paradigma imperativo con la orientación a objetos.



Defina los siguientes:

Multipropósito

Lo que significa que es aplicable en numerosos campos de la industria del software tales como Inteligencia Artificial, Big Data, Hacking ético, desarrollo web, creación de aplicaciones y scripting, y su flexibilidad abarca la posibilidad de que escribas tu código de múltiples maneras y estilos sin complicaciones.

Lenguaje interpretado

Es aquel que el código fuente se ejecuta directamente, instrucción a instrucción.
Es decir, el código no pasa por un proceso de compilación, sino que tenemos un programa llamado intérprete que lee la instrucción en tiempo real, y la ejecuta.



Defina a que se refiere cuando se habla de encapsulación y muestre un ejemplo(Código en Python).

```
1 class Persona:
2     __nombres = None
3     __apellidos = None
4     __edad = None
5
6     def _init_(self, nombres, apellidos, edad):
7         self.nombres = nombres
8         self.apellidos = apellidos
9         self.edad = edad
10
11     def __metodo_private(self):
12         print('metodo privado')
13
14     def metodo_publico(self):
15         print('metodo publico')
16
17     def _str_(self) -> str:
18         return f'Persona: {self.nombres}, apellidos: {self.apellidos}'
19
```



Defina a que se refiere cuando se habla de herencia y muestre un ejemplo(Código en Python).

```
1 class Persona:
2     __nombres = None
3     apellidos = None
4     documento = None
5     tipo = None
6
7     #constructor
8
9     def __init__(self, nombres, apellidos, documento, tipo):
10         self.nombres = nombres
11         self.apellidos = apellidos
12         self.documento = documento
13         self.tipo = tipo
14
15     def __str__(self):
16         return f'Nombres: {self.nombres} \nApellidos: {self.apellidos} \nCI: {self.documento}'
17
18     def modifica_tipo(self, tipo):
19         self.tipo = tipo
20
21     #Herencia
22     class Cliente(Persona):
23         categoria = None
24         codigo = None
25
```




```
def __init__(self, categoria, codigo, nombres, apellidos, documento, tipo):
    Persona.__init__(self, nombres, apellidos, documento, tipo)
    self.categoria = categoria
    self.codigo = codigo

def __str__(self):
    return Persona.__str__(self) + f'\nCategoria: {self.categoria} \nCodigo: {self.codigo} \nTipo: {self.tipo}'

def generar_codigo(self):
    return self.codigo

def modifica_tipo_cliente(self, tipo):
    Persona.modifica_tipo(self, tipo)

client1 = Cliente('VIP', 'cod-123', 'William', 'Barra', '6789003', 'LP')
print(client1)
```

- **Que es una Clase**

pueden ser definidas como un molde que contendrá todas las características y acciones con las cuales podemos construir N cantidad de objetos.

- **Que es un Objeto**

Son aquellos que tienen propiedades y comportamientos, estos pueden ser físicos o conceptuales.

- **Que es una instancia.**

Se llama instancia a todo objeto que derive de algún otro. De esta forma, todos los objetos son instancias de algún otro, menos la clase Object que es la madre de todas.





Llevar el siguiente código JAVA a Python.

```
class Main {  
  
    public static void main(String[] args) {  
  
        System.out.println("Enter two numbers");  
        int first = 10;  
        int second = 20;  
  
        System.out.println(first + " " + second);  
  
        // add two numbers  
        int sum = first + second;  
        System.out.println("The sum is: " + sum);  
    }  
}
```

```
1  n1 = int(input('Ingrese el primer numero: '))  
2  n2 = int(input('Ingrese el segundo numero: '))  
3  sum = n1 + n2  
4  print('La suma es:', sum)  
5
```

Run: main (1) ×

"C:\Users\BRAYAN HONORIO\PycharmProjects\SEMESTRE0
Ingrese el primer numero: 20
Ingrese el segundo numero: 10
La suma es: 30

Process finished with exit code 0



Crear el código JAVA y Python para el siguiente análisis.

```
1 class Persona:
2     name = None
3     email = None
4     gender = None
5     nationality = None
6
7     def __init__(self, name, email, gender, nationality):
8         self.name = name
9         self.email = email
10        self.gender = gender
11        self.nationality = nationality
12
13    def __str__(self):
14        return f'Nombre: {self.name} \nEmail: {self.email} \nGenero: {self.gender} \nNacionalidad: {self.nationality}'
15
16    def escribir_libro(self):
17        print("Escribe un libro")
18
19    def escribir_pelicula(self):
20        print("Escribe una pelicula")
21
22    def modifica_nacionalidad(self, nationality):
23        self.nationality = nationality
24
25    def modifica_email(self, email):
26        self.email = email
```

Activar Windows

```
Run: propiedad x
"C:\Users\BRAYAN HONORIO\PycharmProject
Nombre: Juan
Email: juan@gamil.com
Genero: hombre
Nacionalidad: boliviano
Escribe un libro
Nombre: Juan
Email: juan@gamil.com
Genero: hombre
Nacionalidad: boliviano
Escribe una pelicula
Nombre: Juan
Email: juan@gamil.com
Genero: hombre
Nacionalidad: boliviano
Nombre: Juan
Email: juan@gamil.com
Genero: hombre
Nacionalidad: peruano
```

```
Nombre: Juan
Email: juan@gamil.com
Genero: hombre
Nacionalidad: boliviano
Nombre: Juan
Email: Juanjose@gamil.com
Genero: hombre
Nacionalidad: boliviano

Process finished with exit code 0
```




Crear un programa Python que genere los primeros N números de la serie Fibonacci.

```
1  num = int(input('Digite la cantidad de N:'))
2  pen = 1
3  ult = 1
4  if num == 1:
5      print(0)
6  elif num == 2:
7      print(0, 1)
8  elif num == 3:
9      print(0, 1, 1)
10 else:
11     print(0, 1, 1, end='')
12     for i in range(num - 3):
13         numero = ult + pen
14         print(numero, end='')
15         pen = ult
16         ult = numero
17
```

elif num == 2

Run: fibonacci x

Digitate la cantidad de N: 5

0 1 1 2 3 5 8 13

Process finished with exit code 0

GRACIAS!!!!

