

TAREA FINAL HITO 2

PROGRAMACIÓN DE SISTEMAS EMBEBIDOS

→ Univ. Gerardo Misael Tito Massi

INGENIERÍA DE SISTEMAS



01 Sistemas Embebidos

Son equipos que procesan datos digitalmente, es una combinación de software y hardware.

En si son sistemas operativos creados con el fin de ser controlados por microprocesadores o microcontroladores, de igual manera a los sistemas normales, pero llevados a un fin completamente sistematizado

02



- Un sistema de control de acceso.
- Taxímetro.
- Un sistema de una fotocopidora.
- Control de maquina expendedora.
- La unidad central de control de un automóvil

03

Diferencias Similitudes

En si un sistema embebido es un poco más difícil de definir. Se dedica a un solo propósito, o un pequeño conjunto de propósitos. Los sistemas embebidos se pueden encontrar en casi cada pieza de la electrónica moderna de hecho, son la electrónica.

Mientras que un sistema operativo es un programa, un conjunto de programas que, después de ser cargado inicialmente en la computadora por un programa de arranque, administra todos los demás programas de aplicación en una computadora.



04



MPU

Los microcontroladores MPU no tienen restricciones de memoria de la misma forma que los MCU, utilizan memoria externa para proporcionar almacenamiento de datos y programa

MCU

Es un circuito integrado programable, capaz de ejecutar las órdenes grabadas en su memoria. Está compuesto de varios bloques funcionales que cumplen una tarea específica

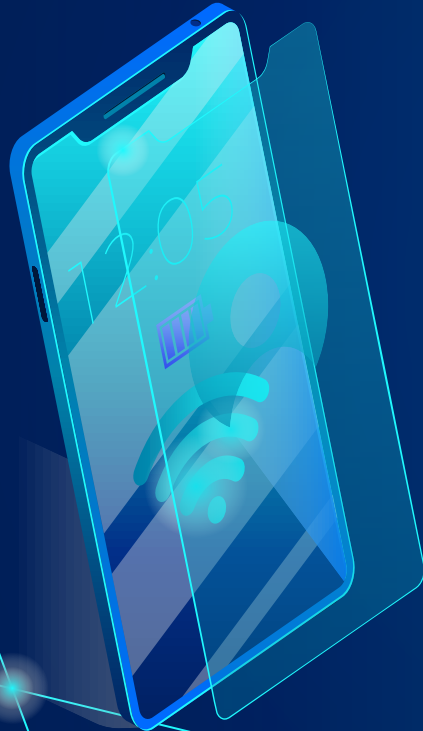


- **Abstracción:** Es cuando separamos los datos de un objeto para luego generar un molde (una clase).
- **Encapsulamiento:** Lo puedes utilizar cuando desees que ciertos métodos o propiedades sean inviolables o inalterables.
- **Herencia:** Nos permite crear nuevas clases a partir de otras.
- **Polimorfismo:** Proviene de Poli = muchas, morfismo = formas. Se utiliza para crear métodos con el mismo nombre pero con diferente comportamiento



06

Componentes de POO



Objeto: Se trata de un ente abstracto usado en programación que permite separar los diferentes componentes de un programa, simplificando así su elaboración, depuración y posteriores mejoras.

Métodos: Son aquellas funciones que permite efectuar el objeto y que nos rinden algún tipo de servicio durante el transcurso del programa.

Eventos: Son aquellas acciones mediante las cuales el objeto reconoce que se está interactuando con él.

Atributos: Características que aplican al objeto solo en el caso en que el sea visible en pantalla por el usuario;

07

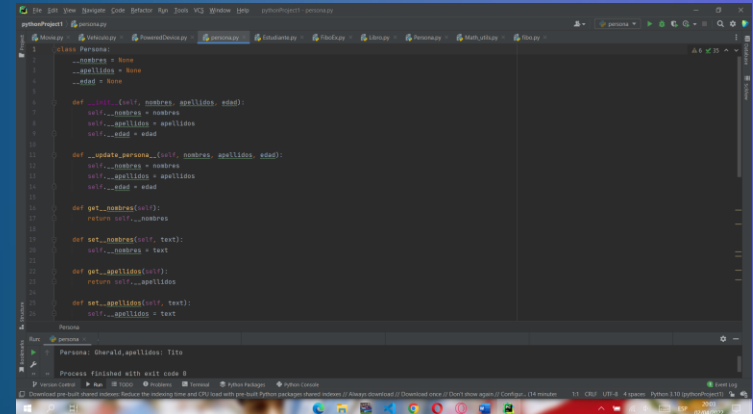
Conceptos

- Multiplataforma.
- Multiparadigma.
- Multipropósito.
- Lenguaje interpretado.

08

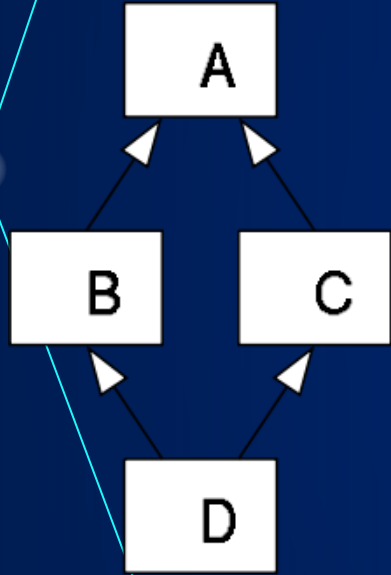
Encapsulamiento:

Es el proceso de almacenar en una misma sección los elementos de una abstracción que constituyen su estructura y su comportamiento; sirve para separar el interfaz contractual de una abstracción y su implantación

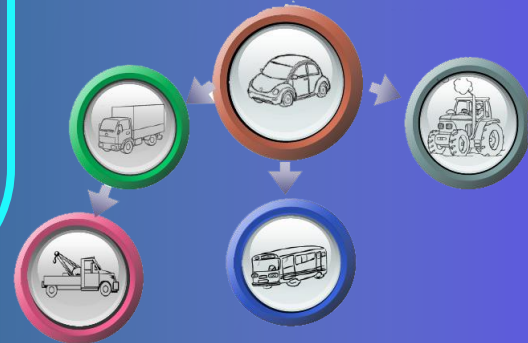


09

Herencia



La herencia permite que se puedan definir nuevas clases basadas de unas ya existentes a fin de reutilizar el código, generando así una jerarquía de clases dentro de una aplicación. Si una clase deriva de otra, esta hereda sus atributos y métodos y puede añadir nuevos atributos, métodos o redefinir los heredados.



10

Conceptos

- Clase: Es una herramienta que tenemos para modelar objetos de programación, lograr que se comporten como queremos y hacer tantas copias de estos objetos como necesitemos.
- Objeto: Se trata de un ente abstracto usado en programación que permite separar los diferentes componentes de un programa, simplificando así su elaboración, depuración y posteriores mejoras.
- Instancia: Se llama instancia a todo objeto que derive de algún otro



01

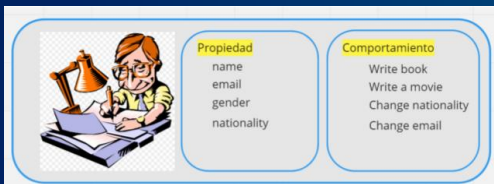
Java a python

```
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);  
    }  
}
```

```
class uno:  
  
    print("Enter two numbers")  
    first = 10  
    second = 20  
    print(str(first) + " " + str(second))  
    sum = first + second  
    print("The sum: " + str(sum))  
  
uno  
uno x  
C:\Users\MASSI\PycharmProjects\pythonProject  
Enter two numbers  
10 20  
The sum: 30  
  
Process finished with exit code 0
```

02

Clase Libro en python



```
C:\Users\MASSI\PycharmProjects\pythonProject1\env\Scripts\python.exe C:/Users/MASSI/Pycha
Comportamiento 1: Massi la llave del gol
Comportamiento 2: Massi mundialero
Comportamiento 3: Argentina
Comportamiento 4: misael10@gmail.com
Nombre: Misael Massi,Email: massi10@gmail.com,Genero: Masculino,Nacionalidad: Argentina

Process finished with exit code 0
```

```
pythonProject1 Testlugador Libro.py
Project Vehiculo.py PoweredDevice.py herenciamultiple.py persona.py FiboEx.py Libro.py
1 class Libro:
2     name = None
3     email = None
4     gender = None
5     nationality = None
6
7
8     def __init__(self, nb, e, g, nc):
9         self.name = nb
10        self.email = e
11        self.gender = g
12        self.nationality = nc
13
14    def writebook(self):
15        print("Comportamiento 1: Massi la llave del gol")
16
17    def writemovie(self):
18        print("Comportamiento 2: Massi mundialero")
19
20    def changenationality(self):
21        print("Comportamiento 3: Argentina")
22
23    def changeemail(self):
24        print("Comportamiento 4: misael10@gmail.com")
25
26    def __str__(self):
27        return "Nombre: {},Email: {},Genero: {},Nacionalidad: {}".format(self.name,
28
29
30    lec1 = Libro("Misael Massi", "massi10@gmail.com", "Masculino", "Argentina")
31    print(lec1)
32    Libro.writebook
33    Libro.writemovie
34    Libro
35
36 Run: uno
37 Process finished with exit code 0
38
39 Version Control Run TODO Problems Terminal Python Packages Python Console
40 Download pre-built shared indexes: Reduce the indexing time and CPU load with pre-built Python packages shared indexes // Always dc
```

03

Fibbonaci

- El programa tiene que leer un valor por consola.
 - Ejem: **N = 8**
- Para el valor leído anteriormente, la salida debería ser:
 - **0, 1, 1, 2, 3, 5, 8, 13,**
- Adjuntar el código generado.
- Adjuntar la imagen(captura) del correcto funcionamiento



```
class FiboEx:
    def fibonacci(num):
        arr = [0, 1]
        if num == 1:
            print('0')
        elif num == 2:
            print('[0, ' + '1]')
        else:
            while (len(arr) < num):
                arr.append(0)
            if (num == 0 or num == 1):
                return 1
            else:
                arr[0] = 0
                arr[1] = 1
                for i in range(2, num):
                    arr[i] = arr[i - 1] + arr[i - 2]
                print(arr)
        fibonacci(num=int(input("Ingrese límite de fibbonaci: ")))
```

FiboEx

FiboEx ×

C:\Users\MASSI\PycharmProjects\pythonProject1\venv\Scripts\python.exe

Ingrese límite de fibbonaci: 10

[0, 1, 1, 2, 3, 5, 8, 13, 21, 34]

Process finished with exit code 0

04

Clases vehiculos

```

class Vehiculo():
    color = None
    wheels = None

    def __init__(self, color, wheels):
        self.color = color
        self.wheels = wheels

    def __str__(self):
        return "Vehiculo Color: {}, Cantidad ruedas {}".format(self.color, self.wheels)

class Car(Vehiculo):

    def __init__(self, color, wheels, seats, engine):
        Vehiculo.__init__(self, color, wheels)
        self.seats = seats
        self.engine = engine

    def __start__(self):
        print("Encendiendo vehiculo")

    def __accelerate__(self):
        print("Acelerando vehiculo")

    def __str__(self):
        return Vehiculo.__str__(self) + ", {} km/h, {} cc".format(self.seats, self.engine, self

c = Car("azul", 4, 49, 1000)
print(c)
Car.__start__

```

```

Car.__start__
Car.__accelerate__

class Bicycle(Vehiculo):

    def __init__(self, color, wheels, saddles, chain):
        Vehiculo.__init__(self, color, wheels)
        self.saddles = saddles
        self.chain = chain

    def __startb__(self):
        print("Iniciando Bici")

    def __accelerateb__(self):
        print("Acelerando bici")

    def __str__(self):
        return Vehiculo.__str__(self) + ", {} sillass, {} cond.".format(self.seats, self.engine, self

b = Bicycle("Rojo", 2, 4, 10)
print(b)
Bicycle.__startb__
Bicycle.__accelerateb__

```

```

C:\Users\MASSI\PycharmProjects\pythonProject1\venv\Scripts\python.exe
Encendiendo vehiculo
Acelerando vehiculo
Vehiculo Color: azul, Cantidad ruedas 4 , 49 km/h, 1000 cc
Iniciando Bici
Acelerando bici
Vehiculo Color: Rojo, Cantidad ruedas 2 , 4 sillass, 10 cond.

Process finished with exit code 0

```

05

Powered Device

```

class PoweredDevice:
    _Modelo = None
    _Serie = ''

    def __init__(self, M, S):
        self._Modelo = M
        self._Serie = S

    def __str__(self):
        return f'Modelo: {self._Modelo}\nSerie: {self._Serie}'

    def Proceso(self):
        print("Iniciando Proceso")

class Scanner(PoweredDevice):
    _Tamaño = None

    def __init__(self, _Modelo, _Serie, _T):
        PoweredDevice.__init__(self, _Modelo, _Serie)
        self._Tamaño = _T

class Printer(PoweredDevice):
    _capacidad = None

    def __init__(self, _Modelo, _Serie, _Capacidad):
        PoweredDevice.__init__(self, _Modelo, _Serie)
        self._capacidad = _Capacidad

class Copier(Printer):

```

```

    _Fax = None

    def __init__(self, _Modelo, _Serie, _Tamaño, _capacidad, _F):
        Scanner.__init__(self, _Modelo, _Serie, _Tamaño)
        Printer.__init__(self, _Modelo, _Serie, _capacidad)
        self._Fax = _F

    def __str__(self):
        return f'Modelo: {self._Modelo}\nSerie: {self._Serie}\nTamaño: {self._Tamaño}\nCapacidad: {self._capacidad}\nFax: {self._Fax}{self.Proceso()}'

Prt = Copier("L735", "1000", "Industrial jj", "Ea", "Si")
print(Prt)

```

```

C:\Users\MASSI\PycharmProjects\pythonProject1\venv\Scripts\python.exe
Iniciando Proceso
Modelo: L735
Serie: 1000
Tamaño: Industrial jj
Capacidad: Ea
Fax: SiNone

Process finished with exit code 0

```

06

Caso: Selección Boliviana

```

class SeleccionBolivia:
    nombres: None
    apellidos: None
    edad: None
    __sueldo: None

    def __init__(self, nombres, apellidos, edad, sueldo):
        self.nombres = nombres
        self.apellidos = apellidos
        self.edad = edad
        self.__sueldo = sueldo

    def get__sueldo(self):
        return self.__sueldo

    def set__sueldo(self, num):
        self.__sueldo = num

class JugadorTitular(SeleccionBolivia):
    dorsal = None
    equipo = None

    def __init__(self, nombres, apellidos, edad, sueldo, dorsal, equipo):
        SeleccionBolivia.__init__(self, nombres, apellidos, edad, sueldo)
        self.dorsal = dorsal
        self.equipo = equipo

```

```

class JugadorSuplente(SeleccionBolivia):
    categoria = None

    def __init__(self, nombres, apellidos, edad, sueldo, categoria):
        SeleccionBolivia.__init__(self, nombres, apellidos, edad, sueldo)
        self.categoria = categoria

class Inscrpion(JugadorTitular, JugadorSuplente):
    contrato = None

    def __init__(self, nombres, apellidos, edad, sueldo, dorsal, equipo, categoria, contrato):
        JugadorTitular.__init__(self, nombres, apellidos, edad, sueldo, dorsal, equipo)
        JugadorSuplente.__init__(self, nombres, apellidos, edad, sueldo, categoria)
        self.contrato = contrato

    def __str__(self):
        return f'El jugador: {self.nombres} {self.apellidos}\nEdad: {self.edad}\nSueldo: {self.get__sueldo()}\nDorsal: {self.dorsal}\nEquipo: {self.equipo}\nCategoria: {self.categoria}\nContrato: {self.contrato}'

insc = Inscrpion('Misael', 'Massi', '25', 15000, '10', 'Bolivar', 'A', '1año')
print(insc)

```

```

C:\Users\MASSI\PycharmProjects\python
El jugador: Misael Massi
Edad: 25
Sueldo: 15000
Dorsal: 10
Equipo: Bolivar
Categoria: A
Tiene un contrato de: 1año

Process finished with exit code 0

```



GRACIAS!

Sigue adelante nunca te rindas.
Sí cañas me invitas.



CREDITS: This presentation template was created by **Slidesgo**, including icons by **Flat** and infographics & illustrations by **Freepik**

