

JAVA CLASS.
this
CLASS.
this
object
this
P O O

| Introduccion a la programación
orientada a objetos (POO)



Programando en c

Entidades

```
struct
{
    char nombre[64];
    char apellido[64];
    char email[256];
    char genero;
    char profesion[256];
    char nacionalidad[256];
    char usuario[64];
    int id;
}typedef Cliente;
```

Funciones

```
int cliente_setNombre(Cliente* this, char* nombre)
{
    int retorno = -1;
    if(this != NULL && nombre != NULL && isValidName(nombre))
    {
        retorno = 0;
        strcpy(this->nombre, nombre);
    }
    return retorno;
}
```

Uso Funciones

```
Cliente* auxCliente = cliente_new();
```

```
cliente_setNombre(auxCliente, "Pedro");
```

Clases

- **Modelo (o plantilla)** para crear objetos de ese tipo.
- Describe el estado y el comportamiento que todos los objetos de la clase comparten.
- Formada por **atributos** y **métodos**

Objeto

- Un objeto representa un item individual e identificable, o una entidad real o abstracta
- Los objetos creados a partir de una clase los llamamos **instancias** de la misma
- En tiempo de ejecución realiza las tareas de un programa.
- Es capaz de recibir mensajes, procesar datos y enviar mensajes a otros objetos

Atributos

Lenguaje C

```
struct typedef
{
    char nombre[20];
    int edad;
}Persona
```

```
Persona p;
p.edad = 18;
```

Lenguaje C#

```
public class Persona
{
    int edad;
    String nombre;
}
```

```
Persona p = new Persona();
p.edad = 18;
```


Funciones en C

```
void per_setNombre(Persona* p,char* nombre)
{
    strncpy(p->nombre,nombre,20);
}
```

```
void per_setEdad(Persona* p,int edad)
{
    if(edad<=99)
        p->edad=edad;
}
```

```
Persona p;
per_setEdad(&p,56);
per_setNombre(&p,"juan");
```

Funciones en C#

```
public void setNombre(String nombre) {  
    this.nombre = nombre;  
}
```

```
public void setEdad(int edad) {  
    this.edad = edad;  
}
```

```
public void printPersona() {  
    Console.WriteLine(this.nombre+" edad:"+this.edad);  
}
```

```
p.setEdad(56);  
p.setNombre("juan");  
p.printPersona();
```

Constructores

- Es un método más en la clase
- No devuelve ningun valor
- Se ejecuta luego de construir el objeto en memoria
- No es necesario definirlo
- Puede recibir argumentos:
Inicializacion de atributos al generar el objeto

Persona p = new Persona();

Atributos estáticos

- Son atributos de la clase, no de cada objeto
- No se necesita un objeto para usarlos

```
class Persona {  
    String nombre;  
    int edad;  
    static int a;  
  
    public Persona() {  
        this.nombre = "";  
        this.edad = 0;  
    }  
}  
  
Persona.a = 5;
```

Métodos estáticos

- No se ejecutan en el contexto de un objeto
- No se necesita un objeto para usarlos
- Por eso no pueden usar this

```
public static void saludo()  
{  
    Console.WriteLine("Hola");  
}
```