

# **Unidad IV Modelado Orientado a Objetos con Estructuras de Control Secuenciales y Selectivas**

**Tema 02 Metodología para la construcción de modelos Orientados a Objetos.**

**Introducción a la Computación**

# Contenido

## 2. Metodología para la construcción de modelos Orientados a Objetos.

### a. Fase de Análisis:

- Subrayado de nombres y verbos.
- Asignación de responsabilidades.
- Definir la lógica de los métodos

### b. Fase de Diseño.

### c. Fase de Implementación

## 2. Metodología para la construcción de Modelos Orientados a Objetos.

La Metodología para la construcción de modelos Orientados a Objetos comprende tres fases:

**Fase de Análisis:** Separa las partes de un problema con la finalidad de estudiar su naturaleza. Busca determinar los objetos del mundo real, presentes en el dominio del problema. Comprende 3 pasos:

- 1) ***Subrayado de nombres y verbos:*** Determinar los candidatos a Clases, Atributos y métodos.
- 2) ***Asignación de responsabilidades*** determinando clases, atributos y métodos
- 3) ***Definir la lógica de los métodos:*** Describir la lógica de los métodos así como la estructura de control que se usará en cada uno de estos.

## 2. Metodología para la construcción de Modelos Orientados a Objetos.

Vemos como un ejemplo cada paso de la metodología para cada fase:

**Fase de Análisis:** Separa las partes de un problema con la finalidad de estudiar su naturaleza. Busca determinar los objetos del mundo real, presentes en el dominio del problema. Comprende 3 pasos:

1) ***Subrayado de nombres y verbos:*** Determinar los candidatos a Clases, Atributos y métodos.

### Ejemplo

De un vendedor se conoce el nombre, el monto de las ventas, sueldo base. Calcule la comisión que recibirá el vendedor que será del 17% del monto de las ventas. Calcule el sueldo final a percibir por el vendedor. Muestre el nombre la comisión y el sueldo final

## 2. Metodología para la construcción de Modelos Orientados a Objetos.

**Fase de Análisis:** Separa las partes de un problema con la finalidad de estudiar su naturaleza. Busca determinar los objetos del mundo real, presentes en el dominio del problema. Comprende 3 pasos:

**2) *Asignación de responsabilidades*** determinando clases, atributos y métodos

Lista	Responsabilidad
Vendedor	Clase
nombre montoVenta sueldoBase	Atributos/Vendedor
setNombre setMontoVenta setSueldoBase getNombre getMontoVenta getSueldoBase calcularComision calcularSueldoFinal	Métodos /Vendedor
InterfaceVendedor	Clase
leerNombre leerMontoVentas leerSueldoBase	Métodos /InterfaceVendedor

## 2. Metodología para la construcción de Modelos Orientados a Objetos.

**Fase de Análisis:** Separa las partes de un problema con la finalidad de estudiar su naturaleza. Busca determinar los objetos del mundo real, presentes en el dominio del problema. Comprende 3 pasos:

3) ***Definir la lógica de los métodos:*** Describir la lógica de los métodos así como la estructura de control que se usará en cada uno de estos.

calcularComision

Estructura usada: secuencial

montoVenta \* 0.17;

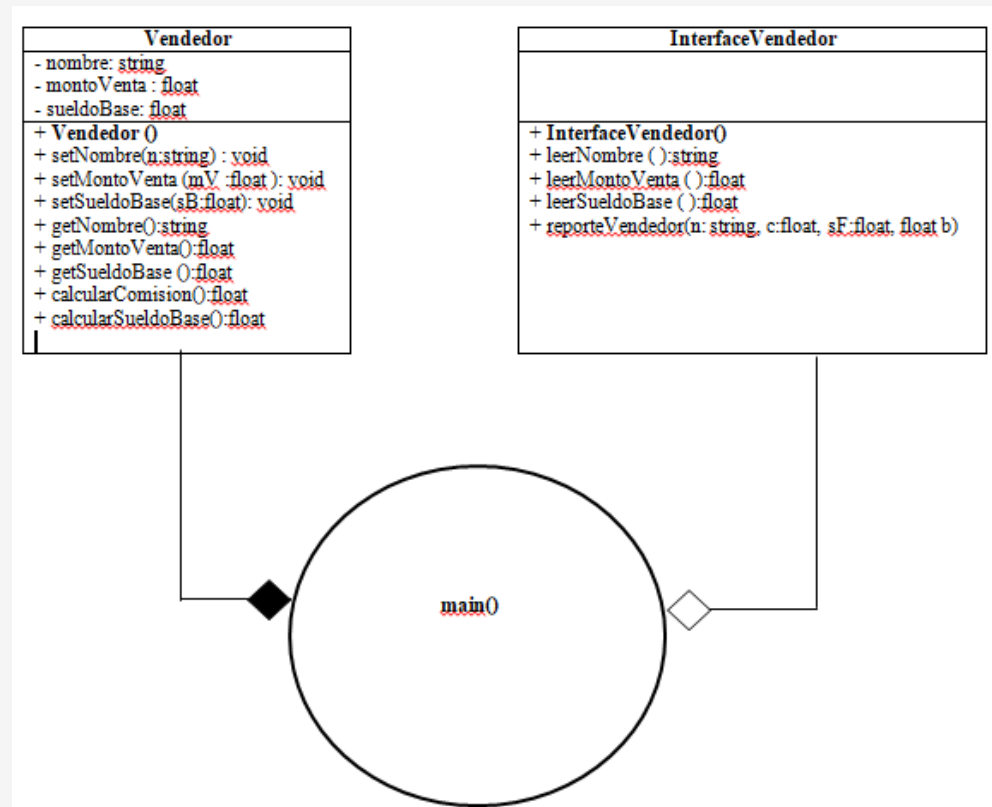
calcularSueldoFinal

Estructura usada: secuencial

sueldoBase + calcularComision()

## 2. Metodología para la construcción de Modelos Orientados a Objetos.

**Fase de Diseño:** Representación gráfica de las clases usando Diagramas de clases en UML. Se emplean las siguientes relaciones de asociación entre clases (Uso -Agregación)



## 2. Metodología para la construcción de Modelos Orientados a Objetos.

**Fase de Implementación:** Construcción de modelos computacionales empleando un lenguaje formal (Lenguaje de programación)

```
#ifndef VENDEDOR_H
#define VENDEDOR_H
#include <string>
using namespace std;

class Vendedor {
private:
    string nombre;
    float montoVenta, sueldoBase;
public:
    Vendedor();
    void setNombre ( string n);
    void setMontoVenta(float mV);
    void setSueldoBase(float sB);
    string getNombre ();
    float getMontoVenta ();
    float getSueldoBase ();
    float calcularComision();
    float calcularSueldoFinal();
};

#endif /* VENDEDOR_H_ */
```

```
#include "Vendedor.h"

Vendedor::Vendedor() { }

void Vendedor::setNombre ( string n)
{
    nombre = n;
}

void Vendedor::setMontoVenta(float mV)
{
    montoVenta = mV;
}

void Vendedor::setSueldoBase(float sB)
{
    sueldoBase =sB;
}

string Vendedor::getNombre ()
{
    return nombre;
}

float Vendedor::getMontoVenta ()
{
    return montoVenta;
}

float Vendedor::getSueldoBase ()
{
    return sueldoBase;
}

float Vendedor::calcularComision()
{
    return montoVenta * 0.17;
}

float Vendedor::calcularSueldoFinal()
{
    return sueldoBase + calcularComision()+ calcularBono();
}
```



## 2. Metodología para la construcción de Modelos Orientados a Objetos.

**Fase de Implementación:** Construcción de modelos computacionales empleando un lenguaje formal (Lenguaje de programación)

```
#ifndef INTERFACEVENDEDOR_H_
#define INTERFACEVENDEDOR_H_
#include <string>
using namespace std;

class InterfaceVendedor {
public:
    InterfaceVendedor();
    string leerNombre ();
    float leerMontoVentas();
    float leerSueldoBase ();

    void reporteVendedor(string n , float c ,
float sF);
};

#endif /* INTERFACEVENDEDOR_H_ */

#include "InterfaceVendedor.h"
#include <iostream>
using namespace std;

InterfaceVendedor::InterfaceVendedor() {
}

string InterfaceVendedor::leerNombre ()
{
    string n;
    cout << "Nombre";
    cin >> n;
    return n;
}

float InterfaceVendedor::leerMontoVentas()
{
    float mV;
    cout << "Monto de las Ventas";
    cin >> mV;
    return mV;
}

float InterfaceVendedor::leerSueldoBase ()
{
    float sB;
    cout << "Sueldo Base";
    cin >> sB;
    return sB;
}

void
InterfaceVendedor::reporteVendedor(string n
, float c , float sF)
{
    cout << "\n REPORTE DEL VENDEDOR\n";
    cout << "\nNombre: "<< n;
    cout << "\nComisión: "<< c;
    cout << "\nSueldo Final: "<< sF;
    |
}
```

## 2. Metodología para la construcción de Modelos Orientados a Objetos.

**Fase de Implementación:** Construcción de modelos computacionales empleando un lenguaje formal (Lenguaje de programación)

```
//Principal.cpp
#include "Vendedor.h"
#include "InterfaceVendedor.h"

int main ()
{
    Vendedor ven;           // Creación del objeto ven de la clase Vendedor
    InterfaceVendedor intVen; // Creación del objeto intVen de la clase InterfaceVendedor

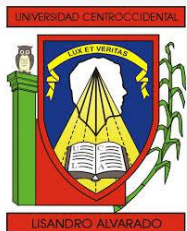
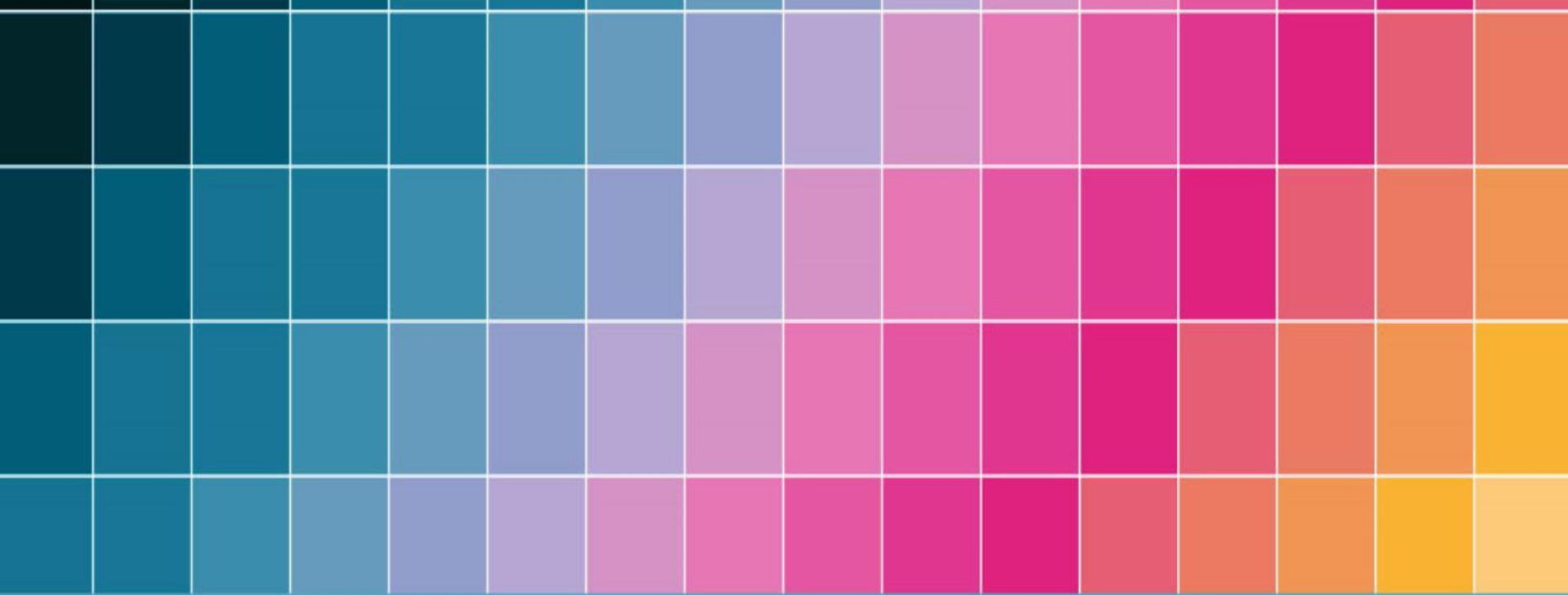
    string n;
    float mV,sB;

    // Se solicita al objeto de la clase InterfaceVendedor, nombre, monto de las ventas y sueldo base
    n = intVen.leerNombre();
    mV = intVen.leerMontoVentas();
    sB = intVen.leerSueldoBase();

    // Se le da estado al objeto de la clase Vendedor, usando el método set de cada atributo
    ven.setNombre(n);
    ven.setMontoVenta(mV);
    ven.setSueldoBase(sB);

    // Se pasa información al método del objeto de la clase InterfaceVendedor, para que imprima el reporte
    intVen.reporteVendedor(ven.getNombre(), ven.calcularComision(), ven.calcularSueldoFinal());

    return 0;
}
```



# **Unidad IV Modelado Orientado a Objetos con Estructuras de Control Secuenciales y Selectivas**

**Tema 02 Metodología para la construcción de modelos Orientados a Objetos.**

**Introducción a la Computación**