



NOMBRE: _____ MAQUINA: _____ FECHA: _____

NOMBRE: _____ MAQUINA: _____ FECHA: _____

PRACTICA 2 “ESTRUCTURAS DE DATOS”

1. Escriba un programa en C++ que declare una estructura de datos para tres empleados de una compañía. Los miembros que debe contener la estructura son:

- a) Número de identificación.
- b) Tasa de pago.
- c) Número de horas trabajadas al mes.

El primer y tercer miembro será ingresado desde teclado por el usuario, el segundo miembro deberá inicializarse con un valor de 8.93 para los tres empleados. El programa debe requisitar debidamente la estructura de datos para cada empleado e imprimirla en pantalla.

2. Utilizando el programa del punto 1, complete el siguiente programa:

```
#include <iostream.h>
#include <iomanip.h>
```

Declaración de estructura EMPLEADO

```
double calc_neto(EMPLEADO );
```

```
int main( )
{
    double pago_neto;
```

Definición, inicialización e impresión de las estructuras Emp1, Emp2 y Emp3.

```
pago_neto = calc_neto(Emp1);
```

```
cout << setw(10)
      << setiosflags (ios::fixed)
      << setiosflags (ios::showpoint)
      << setprecision (2);
```

```
cout << “El pago neto para el empleado: “<< Emp1.num_id
```



**INSTITUTO POLITÉCNICO NACIONAL
ESCUELA SUPERIOR DE INGENIERÍA MECÁNICA Y ELÉCTRICA
INGENIERÍA EN COMUNICACIONES Y ELECTRÓNICA
LABORATORIO DE ESTRUCTURAS Y BASES DE DATOS**



NOMBRE: _____ MAQUINA: _____ FECHA: _____

NOMBRE: _____ MAQUINA: _____ FECHA: _____

```
        << " es $" << pago_neto << endl;

    pago_neto = calc_neto(Emp2);

    cout << setw(10)
        << setiosflags (ios::fixed)
        << setiosflags (ios::showpoint)
        << setprecision (2);

    cout << "El pago neto para el empleado: " << Emp2.num_id
        << " es $" << pago_neto << endl;

    pago_neto = calc_neto(Emp3);

    cout << setw(10)
        << setiosflags (ios::fixed)
        << setiosflags (ios::showpoint)
        << setprecision (2);

    cout << "El pago neto para el empleado: " << Emp3.num_id
        << " es $" << pago_neto << endl;

    return 0;

}

double calc_neto(EMPLEADO temp)
{
    return ( temp.tasa_pago * temp.horas);
}
```

3. Realice el mismo programa del punto 2, pero ahora utilizando la estructura union.