**Universidad Tecnológica de México**

**Verduzco López Rubén**

**Ing. Sistemas Computacionales**

**Ingeniería de Software**

**Miguel Ángel Rafael Arellano**

**Práctica 2**

****



UNIVERSIDAD TECNOLOGICA DE MÉXICO

|  |  |  |  |
| --- | --- | --- | --- |
| **Nombre de la Materia** | **Nombre del Profesor** | **No. Practica** | **Fecha** |
| Ingeniería de Software | Miguel Ángel Rafael Arellano | 2 | 02-Feb-2019 |

**Normas de seguridad del laboratorio de cómputo:**

No colocar sobre la mesa botellas de agua, No usar gorras, utilizar el celular solo en caso de emergencia, entregar el equipo de cómputo como lo recibimos, a esto se refiere conectar los cables que desconectamos (monitor, PC), colocar el mouse sobre el gabinete, acomodar las sillas, recoger la basura y depositarla en un recolector.

**Sistemas de Información.**

**Introducción**

La elaboración de programas para solucionar problemas permite al alumno fortalecer habilidades y conocimientos, con el propósito de llevar a cabo una metodología estructurada en cualquier lenguaje de programación.

**Objetivos:**

* El alumno analizará los enunciados planteados para identificar los requerimientos.
* El alumno diseñará con diagramas el funcionamiento de los modelos de sistemas
* El alumno aplicará sus conocimientos de programación estructurada para realizar los sistemas.
* El alumno comprobará el programa para evaluar si cumple con los requerimientos.

**Competencias**

1.-El alumno fortalece su pensamiento analítico, diseña y crea soluciones de acuerdo a sus conocimientos y experiencias adquiridas en su formación profesional.

**EJERCICIOS**

1. Una empresa busca tabular los impuestos de todos sus empleados, para que sean considerandos en el sueldo bruto correspondiente. Desarrolle un sistema que permita calcular el sueldo neto de cada perfil.
2. Agregando una compensación adicional al sueldo bruto.
3. Descontando los impuestos ISR, Asistencia Médica, vivienda y pensión alimenticia según sea el caso.
4. Los empleados se clasifican de menor a Mayor utilizando la Letra A para el más básico hasta el empleado E

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sueldo | Adicional | ISR | Asistencia  Medica | Vivienda | Pensión Alimenticia  (opcional) |
| Menor de 10000 | 10% | 5% | 5% | 5% | 10% y 20% |
| Sueldo 10000 entre 29999 | 20% | 10% | 7% | 5% |
| Sueldo 30000 entre 49999 | 20% | 15% | 7% | 5% |
| Sueldo 50000 entre 69999 | 25% | 20% | 10% | 7% | 20 y 25% |
| Sueldo 70000 hasta 100000 | 30% | 25% | 15% | 10% |

El sistema considera como entrada el sueldo del empleado para que pueda clasificarse y determinar los porcentajes de impuesto que debe pagar, además de solicitar el porcentaje de pensión alimenticia que se aplique si fuera el caso. La salida del sistema será el sueldo neto del empleado con los descuentos aplicados respectivamente y clasificando al tipo de empleado.

Para la entrega de la practica realize lo siguiente:

* Describa los procesos o tareas que realiza el sistema.

1. Se pedirá el nombre del empleado con su respectivo saldo, después pedir la pansión alimenticia si este tiene.
2. Después se guardará la información, este paso generará también los porcentajes de cada elemento en automático.
3. Al mostrar a los empleados se observará cada porcentaje dependiendo el saldo.

* Diseñe un diagrama del sistema.

|  |
| --- |
| Empleado |
| Nombre: string  Saldo: double  Pension: bool |
| Guardar();  Mostrar(); |

* Desarrolle el codigo del programa.

**// Métodos**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Practica\_2

{

public class Metodos

{

public static string[,] empleado = new string[25, 9];

public static byte num\_emp = 0;

public void Agregar\_Empleado()

{

num\_emp += 1;

}

public byte Numero\_Empleados()

{

if(num\_emp>=25)

{

MessageBox.Show("Se ha excedido el límite de usuarios para guardar.");

num\_emp = 24;

}

return num\_emp;

}

public void Guardar(string nombre,string saldo,string pen\_alim,byte n\_empleado)

{

empleado[n\_empleado, 0] = nombre; // Nombre

empleado[n\_empleado, 1] = saldo; // Saldo

empleado[n\_empleado, 2] = Tipo\_Saldo(Convert.ToDouble(saldo),pen\_alim,false,false,false,false,false,false); // Porcentaje de la pensión

empleado[n\_empleado, 3] = Tipo\_Saldo(Convert.ToDouble(saldo), pen\_alim, true,false,false,false,false,false); // Clase de empleado

empleado[n\_empleado, 4] = Tipo\_Saldo(Convert.ToDouble(saldo), pen\_alim, false, true, false, false, false, false); // Adicional

empleado[n\_empleado, 5] = Tipo\_Saldo(Convert.ToDouble(saldo), pen\_alim, false, false, true, false, false, false); // ISR

empleado[n\_empleado, 6] = Tipo\_Saldo(Convert.ToDouble(saldo), pen\_alim, false, false, false, true, false, false); // Asistencia médica

empleado[n\_empleado, 7] = Tipo\_Saldo(Convert.ToDouble(saldo), pen\_alim, false, false, false, false, true, false); // Vivienda

empleado[n\_empleado, 8] = Tipo\_Saldo(Convert.ToDouble(saldo), pen\_alim, false, false, false, false, false, true); // Pensión Alimenticia (Saldo)

}

public string Tipo\_Saldo(double s\_empl,string pension\_alim,bool clase,bool adicional,bool r\_isr,bool asis\_medica,bool vivienda,bool pen\_alim\_saldo)

{

string retorno = "";

if(s\_empl<0)

{

MessageBox.Show("Error: No se aceptan números negativos");

num\_emp -= 1;

} // Excepción número negativo

if(s\_empl<10000)

{

if (s\_empl > 5000 && Convert.ToBoolean(pension\_alim))

{

retorno = "0.2"; // 20 %

}

else if (Convert.ToBoolean(pension\_alim))

retorno = "0.1"; // 10 %

else

retorno = "0";

if (clase)

{

retorno = "Clase A";

MessageBox.Show("Empleado guardado con éxito.");

}

if(adicional)

{

s\_empl \*= 0.1;

retorno = Convert.ToString(s\_empl);

}

if(r\_isr)

{

s\_empl \*= 0.05;

retorno = Convert.ToString(s\_empl);

}

if (asis\_medica)

{

s\_empl \*= 0.05;

retorno = Convert.ToString(s\_empl);

}

if (vivienda)

{

s\_empl \*= 0.05;

retorno = Convert.ToString(s\_empl);

}

if((pen\_alim\_saldo)&&(Convert.ToBoolean(pension\_alim)))

{

s\_empl = s\_empl \* Convert.ToDouble(retorno);

retorno = Convert.ToString(s\_empl);

}

} // Menor de 10000

if ((s\_empl > 10000)&&(s\_empl < 30000))

{

if (s\_empl > 15000 && Convert.ToBoolean(pension\_alim))

{

retorno = "0.2"; // 20 %

}

else if (Convert.ToBoolean(pension\_alim))

retorno = "0.1"; // 10 %

else

retorno = "0";

if (clase)

{

retorno = "Clase B";

MessageBox.Show("Empleado guardado con éxito.");

}

if (adicional)

{

s\_empl \*= 0.2;

retorno = Convert.ToString(s\_empl);

}

if (r\_isr)

{

s\_empl \*= 0.1;

retorno = Convert.ToString(s\_empl);

}

if (asis\_medica)

{

s\_empl \*= 0.07;

retorno = Convert.ToString(s\_empl);

}

if (vivienda)

{

s\_empl \*= 0.05;

retorno = Convert.ToString(s\_empl);

}

if ((pen\_alim\_saldo) && (Convert.ToBoolean(pension\_alim)))

{

s\_empl = s\_empl \* Convert.ToDouble(retorno);

retorno = Convert.ToString(s\_empl);

}

} // Entre 10000 y 29999

if ((s\_empl >= 30000) && (s\_empl < 50000))

{

if (s\_empl > 45000 && Convert.ToBoolean(pension\_alim))

{

retorno = "0.2"; // 20 %

}

else if (Convert.ToBoolean(pension\_alim))

retorno = "0.1"; // 10 %

else

retorno = "0";

if (clase)

{

retorno = "Clase C";

MessageBox.Show("Empleado guardado con éxito.");

}

if (adicional)

{

s\_empl \*= 0.2;

retorno = Convert.ToString(s\_empl);

}

if (r\_isr)

{

s\_empl \*= 0.15;

retorno = Convert.ToString(s\_empl);

}

if (asis\_medica)

{

s\_empl \*= 0.07;

retorno = Convert.ToString(s\_empl);

}

if (vivienda)

{

s\_empl \*= 0.05;

retorno = Convert.ToString(s\_empl);

}

if ((pen\_alim\_saldo) && (Convert.ToBoolean(pension\_alim)))

{

s\_empl = s\_empl \* Convert.ToDouble(retorno);

retorno = Convert.ToString(s\_empl);

}

} // Entre 30000 y 49999

if ((s\_empl >= 50000) && (s\_empl < 70000))

{

if (s\_empl > 65000 && Convert.ToBoolean(pension\_alim))

{

retorno = "0.25"; // 25 %

}

else if (Convert.ToBoolean(pension\_alim))

retorno = "0.20"; // 20 %

else

retorno = "0";

if (clase)

{

retorno = "Clase D";

MessageBox.Show("Empleado guardado con éxito.");

}

if (adicional)

{

s\_empl \*= 0.25;

retorno = Convert.ToString(s\_empl);

}

if (r\_isr)

{

s\_empl \*= 0.2;

retorno = Convert.ToString(s\_empl);

}

if (asis\_medica)

{

s\_empl \*= 0.1;

retorno = Convert.ToString(s\_empl);

}

if (vivienda)

{

s\_empl \*= 0.07;

retorno = Convert.ToString(s\_empl);

}

if ((pen\_alim\_saldo) && (Convert.ToBoolean(pension\_alim)))

{

s\_empl = s\_empl \* Convert.ToDouble(retorno);

retorno = Convert.ToString(s\_empl);

}

} // Entre 50000 y 69999

if ((s\_empl >= 70000) && (s\_empl <= 100000))

{

if (s\_empl > 85000 && Convert.ToBoolean(pension\_alim))

{

retorno = "0.25"; // 20 %

}

else if (Convert.ToBoolean(pension\_alim))

retorno = "0.2"; // 20 %

else

retorno = "0";

if (clase)

{

retorno = "Clase E";

MessageBox.Show("Empleado guardado con éxito.");

}

if (adicional)

{

s\_empl \*= 0.3;

retorno = Convert.ToString(s\_empl);

}

if (r\_isr)

{

s\_empl \*= 0.3;

retorno = Convert.ToString(s\_empl);

}

if (asis\_medica)

{

s\_empl \*= 0.15;

retorno = Convert.ToString(s\_empl);

}

if (vivienda)

{

s\_empl \*= 0.1;

retorno = Convert.ToString(s\_empl);

}

if ((pen\_alim\_saldo) && (Convert.ToBoolean(pension\_alim)))

{

s\_empl = s\_empl \* Convert.ToDouble(retorno);

retorno = Convert.ToString(s\_empl);

}

} // Entre 70000 y 100000

if (s\_empl > 100000)

{

MessageBox.Show("Error: No se aceptan mayores a 100000");

num\_emp -= 1;

} // Excepción mayores a 100000

return retorno;

}

public string Mostrar()

{

byte no\_empleado = 1;

string retorno = "No. Empleado Nombre Clase Saldo Adicional ISR Asistencia Médica Vivienda Pensión Alimenticia\n";

// Protege el colapso del sistema

if (empleado[0, 0] == null)

MessageBox.Show("Error: No se han ingresado los valores");

else

{

// Mostrar empleados

for(byte v=0;v<25;v++)

{

if (num\_emp < v)

break;

if(v==0)

{

no\_empleado = v;

no\_empleado += 1;

}

if (v > 0)

no\_empleado += 1;

retorno = retorno +" " +no\_empleado+".- "+empleado[v,0]+" "+empleado[v,3]+" "+empleado[v,1]+" "+empleado[v,4]+" "+empleado[v,5]+" "+empleado[v,6]+" "+empleado[v,7]+" "+empleado[v,8]+"\n";

}

}

return retorno;

}

}

}

**// Ingresar**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Practica\_2

{

public partial class Ingresar : Form

{

Metodos m = new Metodos();

public Ingresar()

{

InitializeComponent();

no\_sist.Text = "Sistema 2";

no\_version.Text = "Versión 1.0";

txt\_nombre.Text = "Nombre:";

txt\_saldo.Text = "Saldo:";

ckb\_pension.Text = "Pensión Alimenticia";

creador.Text = "Elaborado por:\nRubén Verduzco López";

b\_guardar.Text = "Guardar";

b\_mostrar.Text = "Mostrar Usuarios";

b\_cerrar.Text = "Cerrar";

}

private void b\_guardar\_Click(object sender, EventArgs e)

{

m.Guardar(txb\_nombre.Text, txb\_saldo.Text, Convert.ToString(ckb\_pension.Checked),m.Numero\_Empleados());

m.Agregar\_Empleado();

}

private void b\_mostrar\_Click(object sender, EventArgs e)

{

Mostrar\_Empleados mu = new Mostrar\_Empleados();

mu.Show();

Hide();

}

private void b\_cerrar\_Click(object sender, EventArgs e)

{

Close();

}

}

}

**// Mostrar\_Empleados**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

namespace Practica\_2

{

public partial class Mostrar\_Empleados : Form

{

Metodos m = new Metodos();

public Mostrar\_Empleados()

{

InitializeComponent();

mostrar\_txt.Text = m.Mostrar();

b\_cerrar.Text = "Cerrar";

}

private void b\_cerrar\_Click(object sender, EventArgs e)

{

Ingresar i = new Ingresar();

i.Show();

Close();

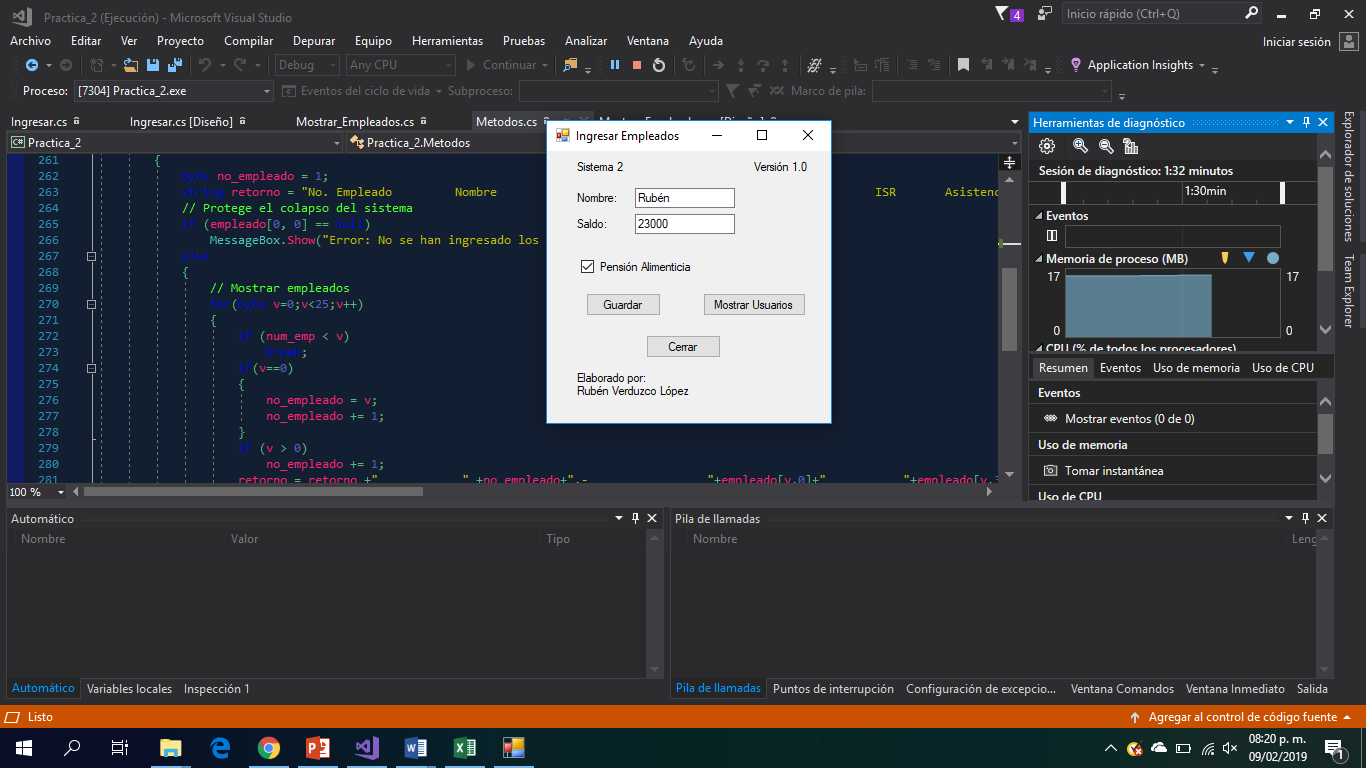
}

}

}

Documente las pruebas con capturas de pantalla y describa esos resultados

1. Se pedirá el nombre del empleado con su respectivo saldo, después pedir la pansión alimenticia si este tiene.
2. Después se guardará la información, este paso generará también los porcentajes de cada elemento en automático.



3.- Al mostrar a los empleados se observará cada porcentaje dependiendo el saldo.

