

# Universidad Abierta Para Adultos



**Formación General**

**CARRERA**

**Ingeniería en Software**

**ASIGNATURA**

**Programación II**

**FACILITADOR**

**Diógenes Amaury Martínez Silverio**

**PARTICIPANTE**

**Vladimir Núñez Calderón**

**Lewis Sánchez**

**MATRICULA**

**100057484**

**100056056**

**TEMA**

**Proyecto Final**

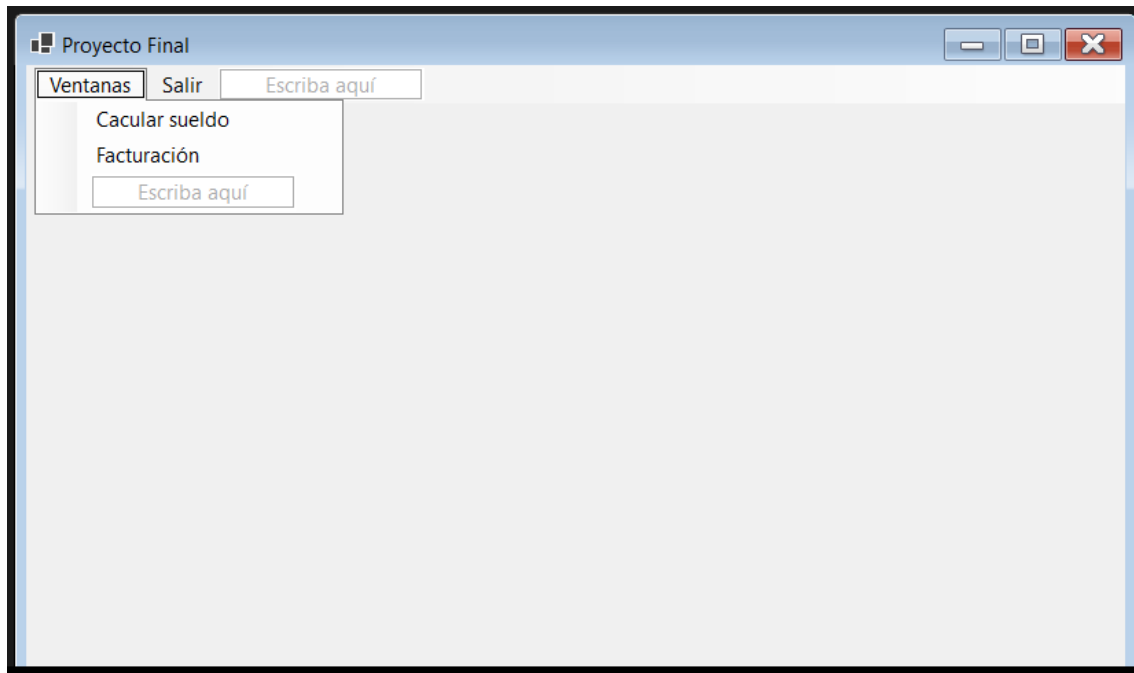
**FECHA**

**20 de marzo del 2023**

**Santiago de los Caballeros, República Dominicana**

## **Introducción**

Hablaremos sobre algunos fragmentos de código de un programa de Windows Forms para el cálculo de sueldo y facturación. Hemos revisado cómo se implementa el menú con dos pestañas y cómo se manejan los eventos de los botones para realizar los cálculos correspondientes y para limpiar los campos. También hemos explorado cómo se agregan y eliminan filas en una tabla de productos y cómo se formatea un campo de texto para mostrar números con separadores de miles.



El menú consta de dos pestañas: "Calcular sueldo" y "Facturación".

La pestaña "Calcular sueldo" puede contener opciones para calcular el sueldo de un empleado, como por ejemplo:

- Ingreso de salario.
- Cálculo del salario bruto.
- Deducción de impuestos y otros descuentos para obtener el salario neto.
- Visualización del salario neto calculado.

La pestaña "Facturación" puede contener opciones para facturar productos o servicios, como por ejemplo:

- Ingreso de datos de la factura, como código de producto.
- Ingreso de los productos o servicios vendidos y su precio.
- Cálculo del total de la factura en función de los productos o servicios vendidos y sus precios.

En general, un menú en Windows Form permite al usuario realizar diversas tareas en la aplicación de manera organizada y fácil de usar.

```

1 referencia
private void calcularSueldoToolStripMenuItem_Click(object sender, EventArgs e)
{
    frmCalcularSueldo formulario = new frmCalcularSueldo();
    formulario.Show()
}

```

Este código corresponde a un controlador de eventos en C# para el evento "Click" del elemento de menú "calcularSueldoToolStripMenuItem". Cuando el usuario hace clic en este elemento de menú, se ejecutará el código dentro de este controlador de eventos.

El código crea una instancia del formulario "frmCalcularSueldo" y lo muestra utilizando el método "Show()". Esto significa que cuando el usuario hace clic en "calcularSueldoToolStripMenuItem", se abrirá un nuevo formulario "frmCalcularSueldo".

El propósito del formulario "frmCalcularSueldo" es permitir al usuario ingresar información sobre un empleado y calcular su salario. Es posible que este formulario tenga controles como cajas de texto para ingresar la información del empleado y botones para realizar los cálculos y mostrar los resultados.

En resumen, este código permite al usuario abrir el formulario "frmCalcularSueldo" desde el menú de la aplicación.

The screenshot shows a Windows application window titled "Calcular Sueldo". The window contains a form with two columns of input fields. The left column has three fields labeled "Ingresar sueldo bruto", "Total descuentos", and "Sueldo neto". The right column has four fields labeled "AFP", "ARS", "Cooperativa", and "ISR". At the bottom of the window, there are three buttons: "Salir", "Limpiar", and "Calcular".

```

private void btnCalcular_Click(object sender, EventArgs e)
{
    const float afp = 0.07F;
    const float ars = 0.03F;
    const float cooperativa = 0.02F;
    float isr = 0.00F, totalAFP, totalARS, totalCooperativa, totalISR, totalDescuentos;

    if (!string.IsNullOrEmpty(txtSueldoBruto.Text))
    {
        decimal number;
        bool isNumber = decimal.TryParse(txtSueldoBruto.Text, out number);

        if (isNumber)
        {
            float sueldoBruto = float.Parse(txtSueldoBruto.Text);
            if(sueldoBruto <= 34334)
            {
                isr = 0.00F;
            }
            if(sueldoBruto >= 34334 && sueldoBruto <= 50000)
            {
                isr = 0.15F;
            }
            if(sueldoBruto >= 51000 && sueldoBruto <= 100000)
            {
                isr = 0.20F;
            }
            if(sueldoBruto > 100000)
            {
                isr = 0.25F;
            }
            totalAFP = sueldoBruto * afp;
            totalARS = sueldoBruto * ars;
            totalCooperativa = sueldoBruto * cooperativa;
            totalISR = sueldoBruto * isr;
            totalDescuentos = totalAFP + totalARS + totalCooperativa + totalISR;

            txtAfp.Text = totalAFP.ToString();
            txtArs.Text = totalARS.ToString();
            txtCooperativa.Text = totalCooperativa.ToString();
            txtIsr.Text = totalISR.ToString();
            txtTotalDescuentos.Text = totalDescuentos.ToString();
            txtSueldoNeto.Text = (sueldoBruto - totalDescuentos).ToString();
        }

        if (!isNumber)
        {
            MessageBox.Show("Ingresa un numero valido");
        }
    }
}

```

Este código es el controlador de eventos para el evento "Click" del botón "btnCalcular". Cuando el usuario hace clic en este botón, se ejecutará el código dentro de este controlador de eventos.

El código comienza definiendo constantes para las tasas de descuento de AFP, ARS y cooperativa. Luego inicializa variables para los totales de AFP, ARS, cooperativa, ISR, descuentos y sueldo neto.

El código luego verifica si el campo de texto "txtSueldoBruto" no está vacío y si su contenido es un número válido. Si se cumple esta condición, se realiza el cálculo de descuentos para el sueldo bruto ingresado. Si el sueldo bruto está dentro de ciertos rangos, se aplica una tasa de ISR específica. Luego, se calculan los totales de descuento para AFP, ARS, cooperativa e ISR y se calcula el total de descuentos.

Finalmente, se actualizan los campos de texto correspondientes con los totales calculados y el sueldo neto. Si el campo de texto "txtSueldoBruto" no contiene un número válido, se muestra un mensaje de error utilizando el método "MessageBox.Show()".



```
private void btnLimpiar_Click(object sender, EventArgs e)
{
    txtAfp.Clear();
    txtArs.Clear();
    txtCooperativa.Clear();
    txtIsr.Clear();
    txtSueldoBruto.Clear();
    txtSueldoNeto.Clear();
    txtTotalDescuentos.Clear();
}
```

El código simplemente borra el contenido de todos los campos de texto en el formulario, utilizando el método "Clear()". Esto significa que cuando el usuario hace clic en "btnLimpiar", se borrará todo el contenido ingresado anteriormente en los campos de texto, lo que permitirá al usuario ingresar nuevos valores para calcular los descuentos y el sueldo neto.

```

private void txtSueldoBruto_TextChanged(object sender, EventArgs e)
{
    if (!string.IsNullOrEmpty(txtSueldoBruto.Text))
    {
        decimal number;
        bool result = decimal.TryParse(txtSueldoBruto.Text, out number);

        if (result)
        {
            txtSueldoBruto.Text = number.ToString("N0");
            txtSueldoBruto.SelectionStart = txtSueldoBruto.Text.Length;
            txtSueldoBruto.SelectionLength = 0;
        }
    }
}

```

Este código se encarga de dar formato de separador de miles al valor ingresado en el campo de texto "txtSueldoBruto" al momento de ser cambiado por el usuario. El formato separa los dígitos del número en grupos de tres, utilizando una coma como separador de miles, lo que hace más fácil su lectura.

En la primera línea, se verifica que el campo de texto no esté vacío. Luego, se utiliza el método TryParse para intentar convertir el valor del campo de texto a un número decimal. Si la conversión es exitosa, el valor del campo de texto es actualizado con el número formateado utilizando el método ToString con la especificación "N0", lo que indica que se debe usar separador de miles. Finalmente, se selecciona todo el texto del campo de texto para que el usuario pueda seguir ingresando sin tener que borrar el valor anteriormente ingresado.

Codigo	Nombre Producto	Precio	Cantidad	Total
*				

Eliminar Fila

Total Factura: 0

```

private void btnAgregar_Click(object sender, EventArgs e)
{
    float totalFactura, totalPrecio;
    totalFactura = 0;

    if (txtCantidad.Text == "" || txtCodigo.Text == "" || txtPrecio.Text == "" ||
    txtProducto.Text == "")
    {
        MessageBox.Show("Todos los campos son obligatorios");
        return;
    }

    decimal number;
    bool isCantidadNumber = decimal.TryParse(txtCantidad.Text, out number);
    bool isPrecioNumber = decimal.TryParse(txtPrecio.Text, out number);

    if (isCantidadNumber && isPrecioNumber)
    {
        totalPrecio = float.Parse(txtCantidad.Text) * float.Parse(txtPrecio.Text);
        totalFactura = totalPrecio;
        dgvProductos.Rows.Add(txtCodigo.Text, txtProducto.Text, txtPrecio.Text,
        txtCantidad.Text, totalPrecio);
        txtTotalFactura.Text = (float.Parse(txtTotalFactura.Text) + totalFactura).ToString();

        txtCantidad.Clear();
        txtCodigo.Clear();
        txtPrecio.Clear();
        txtProducto.Clear();
    }
    else
    {
        MessageBox.Show("La cantidad y el precio deben ser un numero");
    }
}

```

Este código es el controlador de eventos para el evento "Click" del botón "btnAgregar" en una aplicación de facturación. Cuando el usuario hace clic en este botón, se ejecutará el código dentro de este controlador de eventos.

El código primero declara dos variables float, "totalFactura" y "totalPrecio", y las inicializa en 0.

Luego, el código comprueba si todos los campos obligatorios están completos. Si alguno de los campos obligatorios (cantidad, código, precio o producto) está vacío, se muestra un mensaje de advertencia al usuario y se detiene la ejecución del código.

Después, el código comprueba si tanto la cantidad como el precio ingresados son números válidos. Si alguno de ellos no es un número válido, se muestra un mensaje de advertencia al usuario.

Si tanto la cantidad como el precio son números válidos, el código calcula el precio total de los productos (multiplicando la cantidad por el precio), actualiza el



precio total de la factura en el campo "txtTotalFactura", agrega los detalles del producto (código, producto, precio, cantidad y precio total) a un DataGridView llamado "dgvProductos", y finalmente borra el contenido de los campos de texto para que el usuario pueda ingresar los detalles de un nuevo producto.



Al hacer clic en este botón, se eliminará la fila seleccionada actualmente del DataGridView "dgvProductos" y se actualizará el campo "txtTotalFactura" restando el valor del producto eliminado.

En detalle, la variable "totalProducto" almacena el valor de la columna de "Total" en la fila seleccionada del DataGridView "dgvProductos". Si el valor es nulo o vacío, se asigna cero. Luego, se calcula la nueva suma total de los productos restando el valor del producto eliminado del total actual. Finalmente, se actualiza el campo "txtTotalFactura" con el nuevo valor y se elimina la fila seleccionada del DataGridView "dgvProductos".

#### Link del video:

<https://drive.google.com/file/d/1pa8ATwT0bL7DZQU9x9PgtD-dyeuaUWYO/view?usp=sharing>

## **Conclusión**

En esta sesión se ha presentado código en lenguaje C# relacionado con el cálculo de sueldo y facturación en una aplicación de Windows Forms. Se han analizado las diferentes funciones y métodos utilizados en cada uno de los códigos, brindando una explicación detallada de su funcionamiento.

En el caso del cálculo de sueldo, se ha mostrado cómo se realiza el cálculo de los diferentes descuentos (AFP, ARS, Cooperativa e ISR) a partir del ingreso del sueldo bruto por parte del usuario. Además, se ha presentado un método para limpiar los campos de la aplicación.

Por otro lado, en el caso de la facturación, se ha explicado cómo se agrega y elimina una fila en la tabla de productos, y cómo se realiza el cálculo del total de la factura. También se ha mostrado cómo se formatea el campo de la cantidad de productos para que solo acepte números enteros.

En conclusión, la aplicación presentada en estos códigos permite al usuario calcular su sueldo neto después de aplicar los diferentes descuentos y también permite al usuario generar una factura a partir de la lista de productos. El código presentado puede ser utilizado como base para el desarrollo de aplicaciones similares.