

**Universidad Nacional de Ingeniería**

.- Facultad de Electrotecnia y Computación -.

Ingeniería en Computación

*Algoritmización y estructuras de datos*

**Laboratorio No. 1**

Alumno:

Marcel Enmanuel Díaz Largaespada (2020-1384U)

Docente:

Adilson González López

Fecha de entrega:

Lunes 23 de agosto del 2021

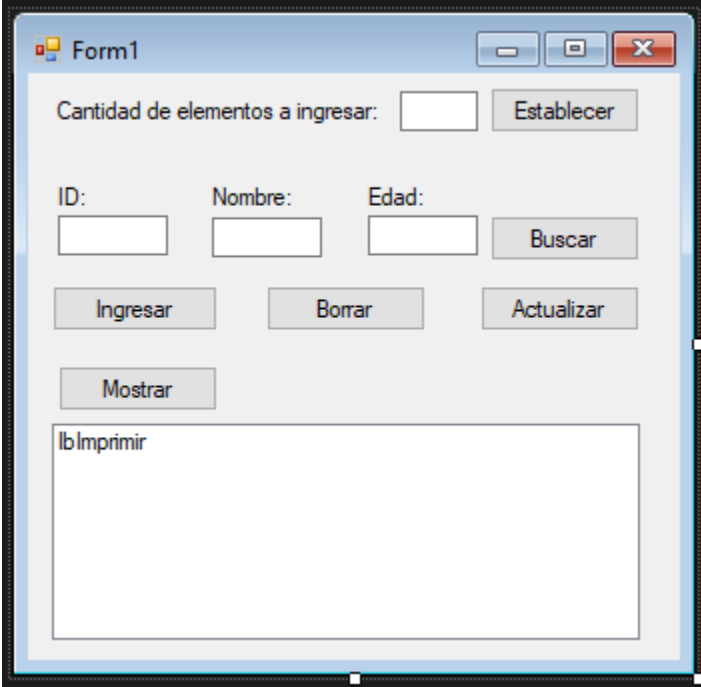
## 1. Introducción

Como forma de introducirnos a las prácticas de laboratorio que estaremos llevando a cabo en la asignatura de *Algoritmización y Estructuras de Datos*, realizamos un laboratorio bastante sencillo.

Dicho laboratorio consistía en una interfaz simple diseñada con las potentes herramientas proporcionadas por el IDE de Visual Studio, con las que pusimos en práctica la lectura y entrada de datos, haciendo uso de arreglos desordenados.

## 2. Diseño del *form*

Se trató de diseñar una interfaz compacta y homogénea.



The image shows a screenshot of a Windows application window titled "Form1". The window has a standard Windows XP-style title bar with minimize, maximize, and close buttons. The main content area is light gray and contains the following elements:

- A label "Cantidad de elementos a ingresar:" followed by a text input field and an "Establecer" button.
- A row of three labels: "ID:", "Nombre:", and "Edad:", each followed by a text input field. To the right of these fields is a "Buscar" button.
- A row of three buttons: "Ingresar", "Borrar", and "Actualizar".
- A single "Mostrar" button.
- A large rectangular area at the bottom, labeled "lbImprimir" in the top-left corner, which appears to be a placeholder for a list or a large text area.

### 3. Código Fuente

#### 3.1. Declaraciones globales.

Las siguientes variables serán las que nos permitirán almacenar los datos que lleguemos a digitar. Entre ellas, también figuran variables de iteración para controlar los ciclos.

```
//Declaraciones globales
int N = 0, Tam, i;
int[] Edad;
int[] Id;
string[] Nombres;
```

---

#### 3.2. Métodos de interacción con la interfaz

Botón Establecer: Nos permitirá establecer el tamaño de los arreglos con el tamaño indicado en la caja de texto anterior.

```
1 referencia
private void btnEstablecer_Click(object sender, EventArgs e)
{
    Tam = int.Parse(txtCantidad.Text);
    Id = new int[Tam];
    Edad = new int[Tam];
    Nombres = new string[Tam];
    N = 0;
}
```

Botón Ingresar: Evalúa si existe espacio en el arreglo para ingresar un nuevo elemento. De ser así, rellena los diferentes arreglos con la información en sus casillas correspondientes. De otro modo, muestra un mensaje de notificación.

```
1 referencia
private void btnIngresar_Click(object sender, EventArgs e)
{
    if (N <= Tam - 1)
    {
        Id[N] = int.Parse(txtId.Text);
        Edad[N] = int.Parse(txtEdad.Text);
        Nombres[N] = txtNombre.Text;
        N = N + 1;
        MessageBox.Show("Elemento Ingresado.");
    }
    else
    {
        MessageBox.Show("No hay espacio");
    }
}
```

Botón Borrar: Lee los datos ingresados en la casilla *ID* en búsqueda de una coincidencia. Si no la encuentra, muestra una notificación. De otro modo, se procede a borrar el elemento y mueve a los elementos posteriores un índice menos.

```
1 referencia
private void btnBorrar_Click(object sender, EventArgs e)
{
    int i = 0;
    int x;
    x = int.Parse(txtId.Text);

    while (i < N && x != Id[i])
    {
        i++;
    }

    if (i >= N)
    {
        MessageBox.Show(x + "No se encuentra registrado.");
    }
    else
    {
        for (int k = i; k < N - 1; k++)
        {
            Id[k] = Id[k + 1];
            Nombres[k] = Nombres[k + 1];
            Edad[k] = Edad[k + 1];
        }
        N--;
        MessageBox.Show("La persona con ID: " + x + " ha sido eliminada.");
    }
}
```

Botón Actualizar: Busca en el arreglo por una coincidencia con los datos ingresados en la casilla *ID*, de no encontrarlo, se notifica. Si lo hace, se procede a rellenar con los datos en las casillas a sus arreglos correspondientes.

```

1 referencia
private void btnActualizar_Click(object sender, EventArgs e)
{
    int i = 0;
    int x;
    x = int.Parse(txtId.Text);

    while (i < N && x != Id[i])
    {
        i++;
    }

    if (i >= N)
    {
        MessageBox.Show(x + "No se encuentra registrado.");
    }
    else
    {
        Id[i] = int.Parse(txtId.Text);
        Edad[i] = int.Parse(txtEdad.Text);
        Nombres[i] = txtNombre.Text;
        MessageBox.Show("La persona con ID: " + x + " ha sido actualizada");
    }
}

```

Botón Mostrar: Limpia el ListBox de impresión y, mediante un ciclo *for*, imprime uno por uno los elementos de todos los arreglos.

```

1 referencia
private void btnMostrar_Click(object sender, EventArgs e)
{
    lbImprimir.Items.Clear();
    for (i = 0; i < N; i++)
    {
        lbImprimir.Items.Add(Id[i] + "\t" + Nombres[i] + "\t" + Edad[i]);
    }
}

```

Botón Buscar: Explora entre los elementos del arreglo ID en búsqueda de una coincidencia con los datos ingresados en la casilla ID. De no lograrlo, se notifica. Si lo hace, se muestra en el panel de impresión el elemento encontrado.

1 referencia

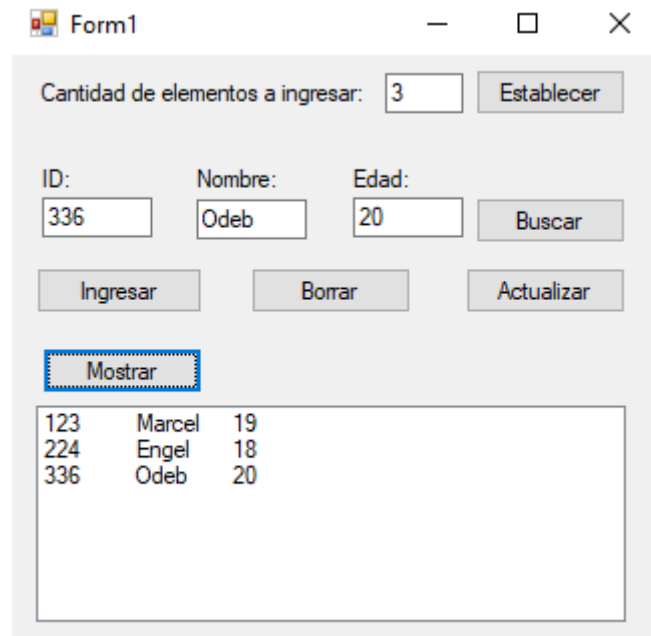
```
private void btnBuscar_Click(object sender, EventArgs e)
{
    int i = 0;
    int x;
    x = int.Parse(txtId.Text);

    while (i < N && x != Id[i])
    {
        i++;
    }

    if (i >= N)
    {
        MessageBox.Show(x + "No se encuentra registrado.");
    }
    else
    {
        lbImprimir.Items.Clear();
        lbImprimir.Items.Add(Id[i] + "\t" + Nombres[i] + "\t" + Edad[i]);
    }
}
```

## 4. Ejecución del programa

El código fue escrutado con detenimiento, con el fin de darnos una idea de cómo funcionaba cada una de sus partes. Luego, y después de habernos cerciorado de que no había ningún error, el resultado fue el siguiente:



Form1

Cantidad de elementos a ingresar: 3 Establecer

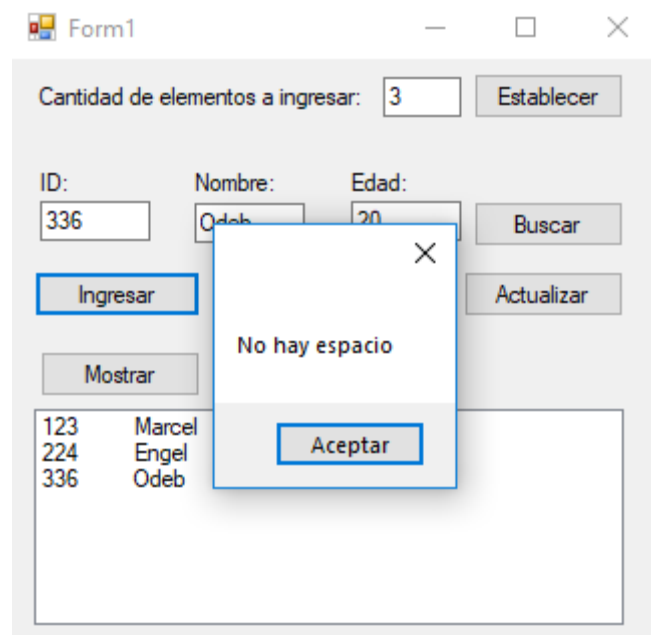
ID: 336 Nombre: Odeb Edad: 20 Buscar

Ingresar Borrar Actualizar

Mostrar

123	Marcel	19
224	Engel	18
336	Odeb	20

*Se estableció una cantidad máxima de 3 elementos y se rellenaron los arreglos*



Form1

Cantidad de elementos a ingresar: 3 Establecer

ID: 336 Nombre: Odeb Edad: 20 Buscar

Ingresar Actualizar

Mostrar

123	Marcel	19
224	Engel	18
336	Odeb	20

No hay espacio

Aceptar

*Cerciorándonos de que la validación funcione apropiadamente*



Form1

Cantidad de elementos a ingresar:  Establecer

ID:  Nombre:  Edad:

La persona con ID: 123 ha sido eliminada.

Aceptar

123  
224  
336

Odeb 20

*Eliminando un elemento*

Form1

Cantidad de elementos a ingresar:  Establecer

ID:  Nombre:  Edad:  Buscar

Ingresar Borrar Actualizar

Mostrar

224	Engel	18
336	Odeb	20

*Mostrando los datos nuevamente en pantalla*

Form1

Cantidad de elementos a ingresar: 3 Establecer

ID: 336 Nombre: Edad: 19

La persona con ID: 336 ha sido actualizada

Aceptar

224  
336

*Actualizando a la persona con ID 336*

Form1

Cantidad de elementos a ingresar: 3 Establecer

ID: 336 Nombre: Isai Edad: 19 Buscar

Ingresar Borrar Actualizar

Mostrar

224	Engel	18
336	Isai	19

*Volviendo a mostrar los datos en pantalla*

## **5. Conclusiones**

Como primera aproximación al entorno de desarrollo con el cual vamos a trabajar la asignatura, es una buena forma de principiar. Se nos dieron las bases con las que podremos construir algoritmos más complejos posteriormente con el lenguaje de C# y el IDE de Visual Studio.

Todo lo aprendido aquí fue aprehendido satisfactoriamente y su operación fue sencilla, didáctica, intuitiva y entretenida.