



UNIVERSIDAD
NACIONAL
AUTÓNOMA DE
NICARAGUA,
MANAGUA

UNAN - MANAGUA

Recinto RURD

Facultad de Ciencias e Ingeniería

Departamento de Computación

Ingeniería en Sistemas de la Información

Tema:

Ejercicios Introdutorios en Windows Forms de Arboles Binarios y Grafos

Estudiante:

Josefina del Carmen López Cruz

Docente: Kener Salinas



Índice

Introducción	3
Desarrollo.....	4
Estructura del Proyecto.....	4
Ejercicio Árbol Binario	4
Ejercicio Grafo	10
.....	11
Conclusión	16



Introducción

El desarrollo de software es un proceso intrincado que se sustenta en la sólida comprensión de conceptos fundamentales de estructuras de datos. En el marco de mi proyecto en el curso, he abordado de manera exhaustiva elementos esenciales que constituyen la base de la programación, destacando la importancia de comprender la estructura de un árbol binario. Este conocimiento no solo nos proporciona una visión clara de su funcionamiento, sino que también sienta las bases para aplicaciones prácticas en el desarrollo de software.

Adicionalmente, quise también enfocarme en los grafos, utilizando un caso de estudio concreto que ilustra la aplicabilidad directa de estas estructuras de datos en situaciones del mundo real. Tomando como base el escenario de paradas de autobuses, creando dinámicamente un grafo que permite a los usuarios agregar vértices personalizados y establecer conexiones entre ellos. Esta aproximación no solo consolida entendimiento teórico de grafos, sino que también nos sumerge en la resolución práctica de problemas mediante la representación gráfica de relaciones entre entidades.

Este proyecto busca, por tanto, no solo sentar las bases teóricas, sino también proporcionar una experiencia práctica y dinámica que fortalezca nuestra capacidad para aplicar estos conceptos en contextos de desarrollo de software.



Desarrollo

Estructura del Proyecto

Ejercicio Árbol Binario

El proyecto implementado sigue una estructura sencilla. A continuación, lo describiré tanto visualmente como contextualmente, de manera que pueda servir también como un manual de usuario.



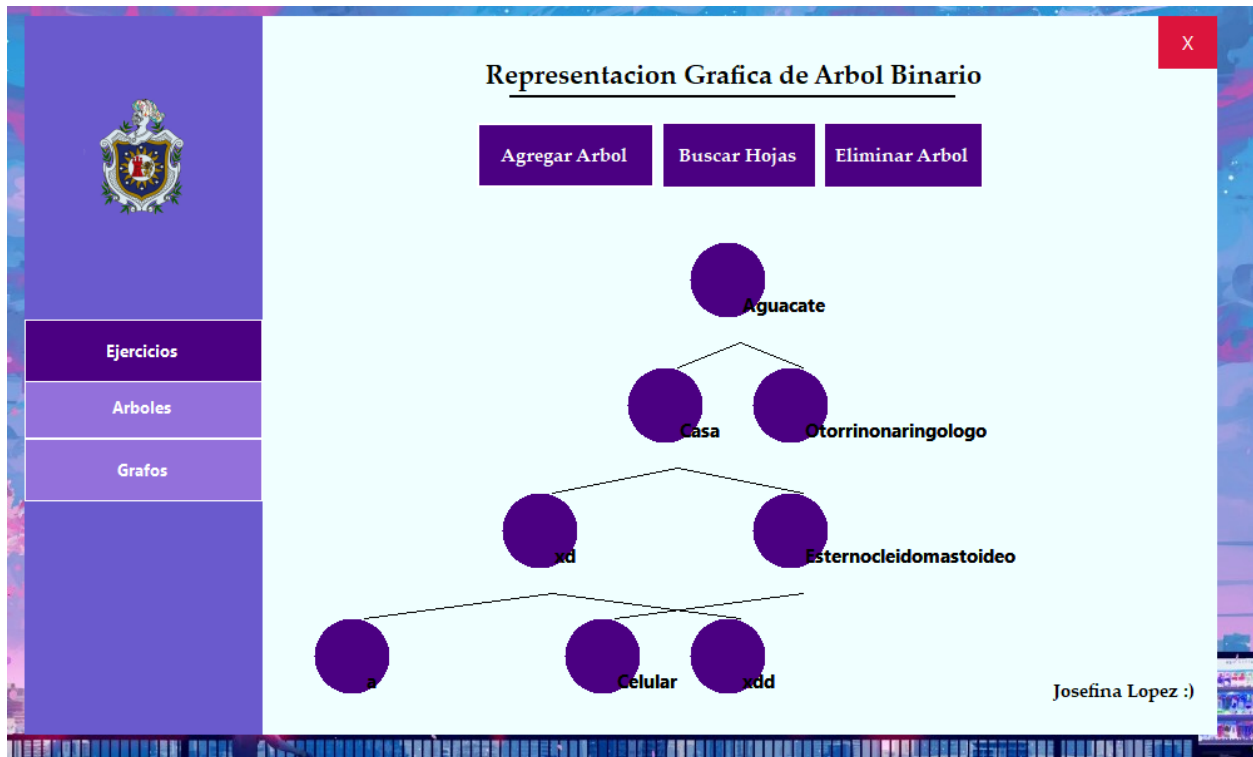
Inicialmente, se presenta un menú que contiene dos ejercicios: árboles y grafos. A continuación, proporcionaré una breve explicación sobre la implementación de los conceptos aprendidos en árboles dentro del entorno de desarrollo de Windows Forms.



Como se puede observar, hay tres botones principales, de los cuales destaco que estos tres son de los cuatro componentes principales. El cuarto y más significativo es el botón "Mostrar", donde se dibujará el árbol binario de manera dinámica e interactiva. Esto se debe a que la representación visual del árbol binario, independientemente de a qué nodo apunte, se adapta según las preferencias del usuario.



Establecemos la primera palabra como raíz y, en función de la longitud de la palabra, creamos un árbol simple. Para lograrlo, empleé una librería de Windows Forms que facilita la representación dinámica de círculos y líneas, mejorando la experiencia del usuario al visualizar gráficamente la estructura del árbol.



En la codificación, empleé un código similar al proporcionado por el profesor en la plantilla oficial de árboles binarios, adaptándolo a Windows Forms. La funcionalidad del botón consiste en:

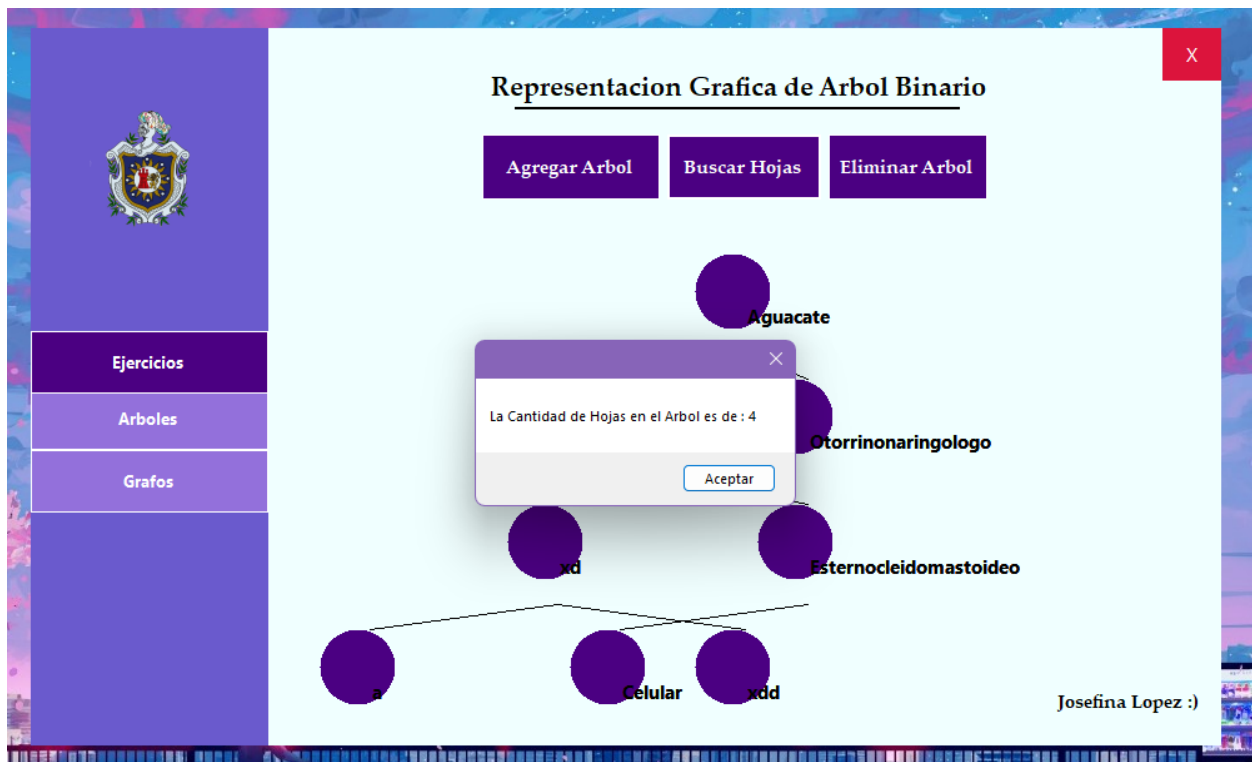
```

private void InsertarNodo(NodoArbol padre, NodoArbol nuevoNodo)
{
    if (padre.Texto.Length > nuevoNodo.Texto.Length)
    {
        if (padre.Izquierdo == null)
        {
            padre.Izquierdo = nuevoNodo;
        }
        else
        {
            InsertarNodo(padre.Izquierdo, nuevoNodo);
        }
    }
    else
    {
        if (padre.Derecho == null)
        {
            padre.Derecho = nuevoNodo;
        }
        else
        {
            InsertarNodo(padre.Derecho, nuevoNodo);
        }
    }
}

```



En otro componente clave, se encuentra un ejercicio propuesto al inicio del curso que contribuirá a una comprensión más profunda de la estructura de un árbol binario. Este ejercicio consiste en determinar cuántas ramas del árbol se consideran hojas. A continuación, adjuntaré una imagen que refuerce este concepto.

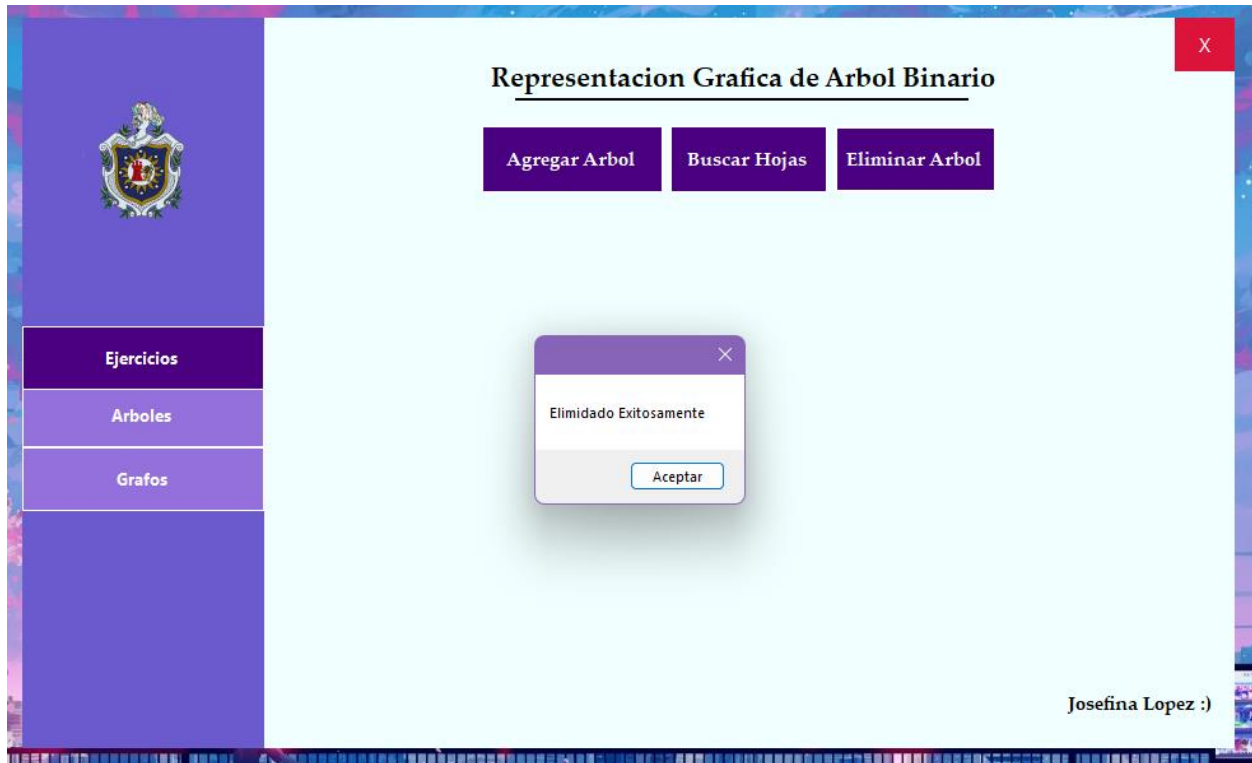


Método programado y utilizado para su funcionamiento: A continuación, se presenta el código correspondiente a la implementación de la funcionalidad.

```
private int ContarHoja(NodoArbol nodo, int cont)
{
    if (nodo == null)
    {
        return 0;
    }
    else
    {
        if (nodo.Izquierdo == null && nodo.Derecho == null)
        {
            cont += 1;
        }

        cont += ContarHoja(nodo.Izquierdo, 0);
        cont += ContarHoja(nodo.Derecho, 0);
    }
    return cont;
}
```


Para concluir, y no menos importante, procedemos a limpiar la estructura eliminando los vértices en la clase y restableciendo la pantalla en Windows Forms.



A continuación, adjunto un fragmento del código que especifica como fue que configure el botón de eliminar, para limpiar los elementos del árbol binario:

```
private void BtnEliminarArbol_Click(object sender, EventArgs e)
{
    nodo.Clear();
    raiz = null;
    this.Invalidate();
    MessageBox.Show("Eliminado Exitosamente");
    this.Close();
}
```

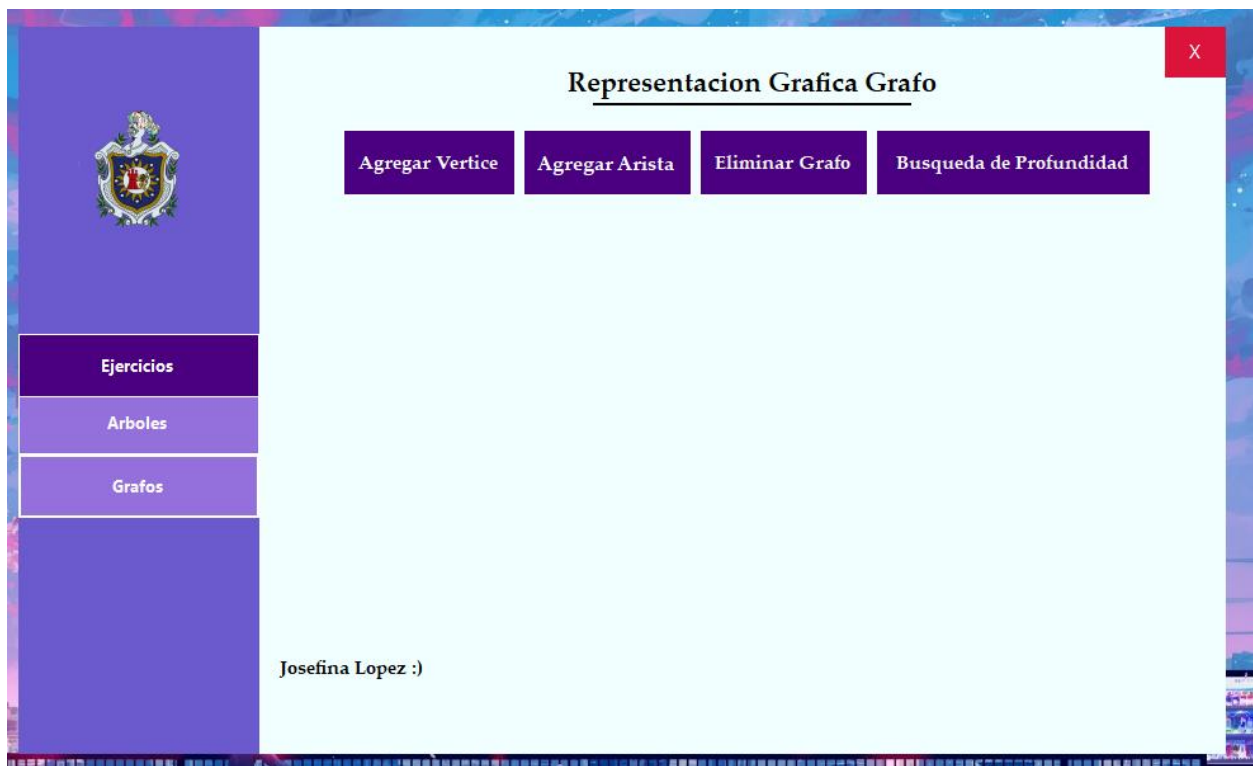


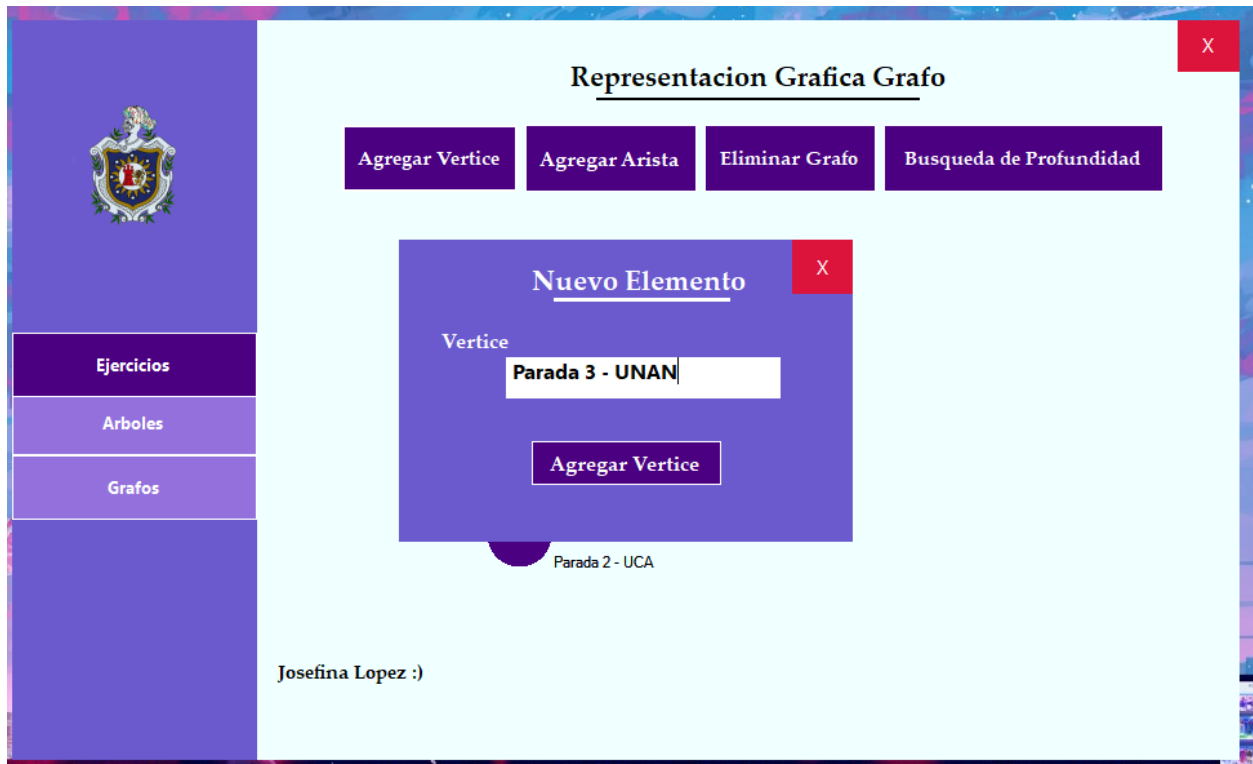
Ejercicio Grafo

Los grafos son una estructura de datos fundamental, en la estructura de mi proyecto, tome como referencia el ejercicio propuesto de las paradas de autobuses, en donde teníamos que diseñar un grafo en donde dependiendo del tipo de búsqueda o recorrido pudiéramos tener una conexión coherente y que cubra ciertas paradas o todas.

Es por ello que en este caso me dispuse a que el usuario pudiera crear su propio grafo a su gusto, pudiendo deslizar los vértices en pantalla y acomodarlos a como mejor le convenga.

Siento las claves los siguientes botones:

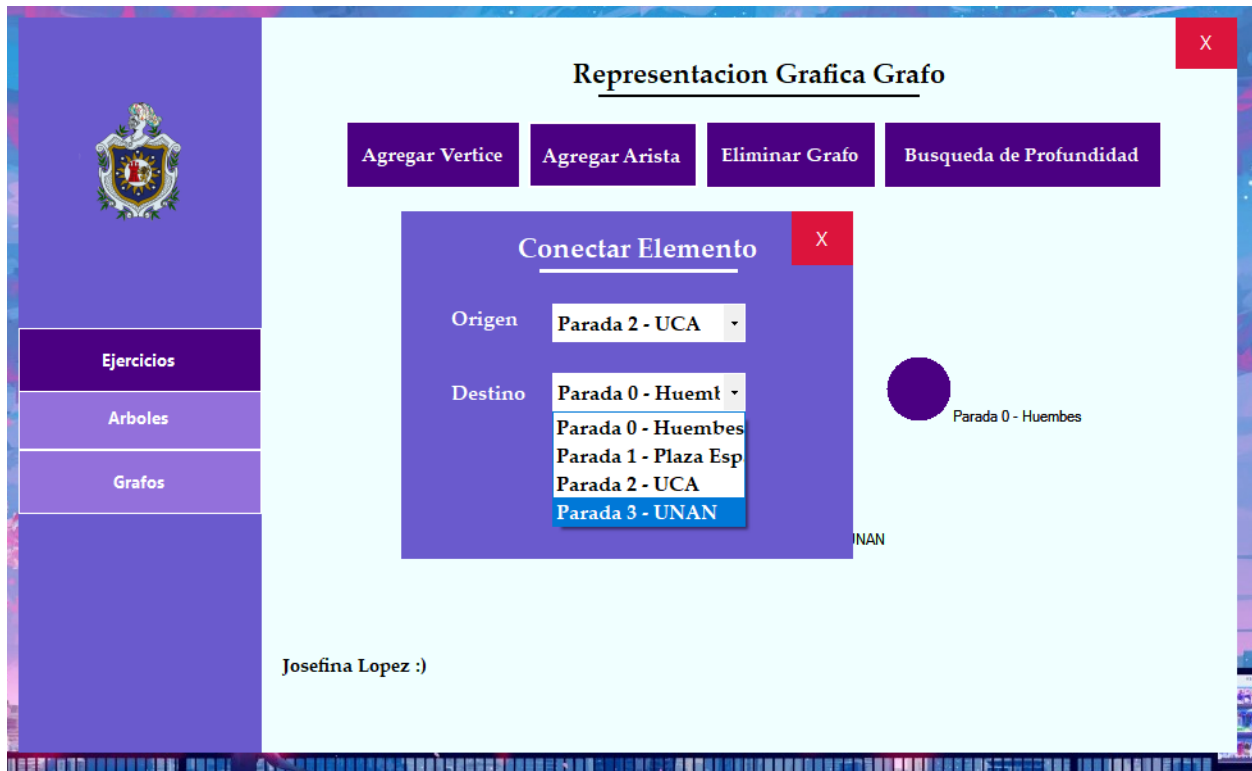




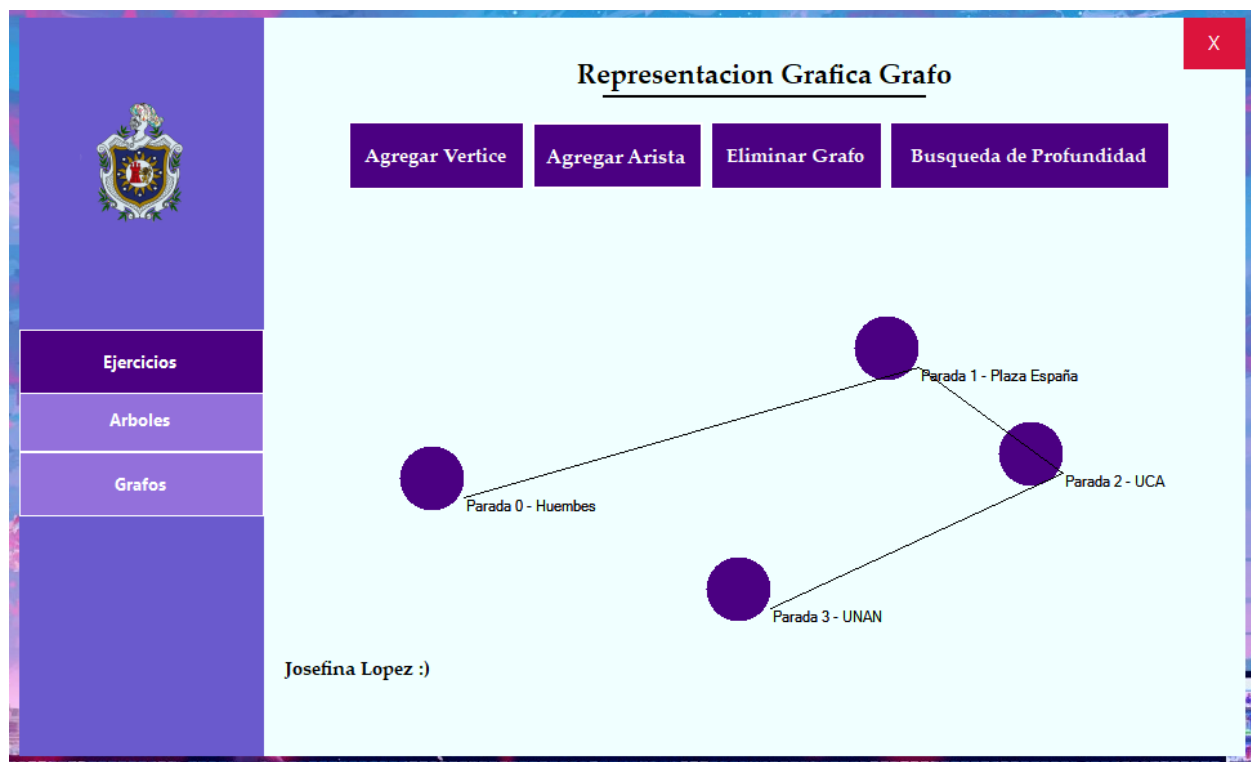
Como se puede observar, se han incorporado vértices al proyecto, donde el usuario proporciona el identificador para cada vértice. En el contexto de la programación, las conexiones entre estos vértices se representan como "Nodo A apunta a Nodo B" o simplemente como la tupla (A, B). En la implementación realizada con Windows Forms, se optó por utilizar dos ComboBox. Estos elementos permiten al usuario seleccionar los vértices A y B respectivamente, estableciendo así la relación donde el vértice A apunta al vértice B. Esta elección aporta una interfaz visual amigable que simplifica la creación y gestión de conexiones entre los diferentes vértices en el proyecto.



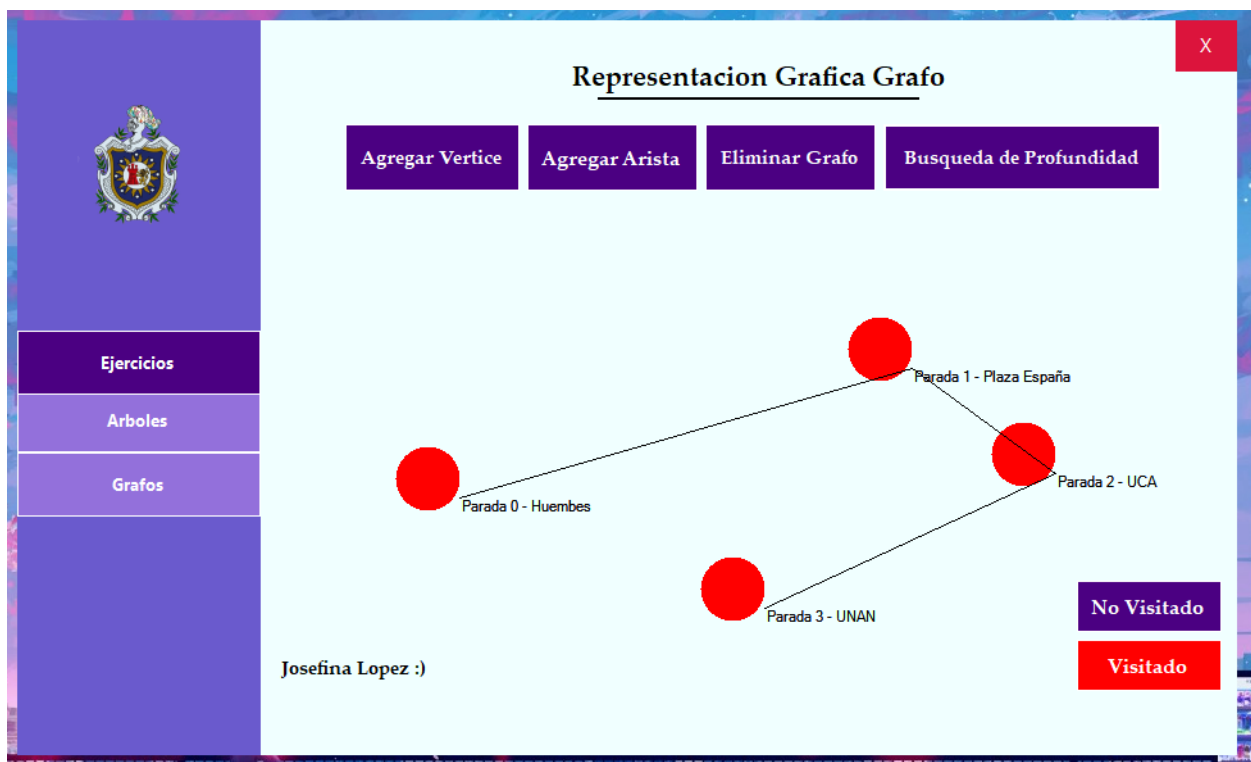
El usuario tiene la capacidad de seleccionar las conexiones de aristas dentro del grafo de acuerdo con sus necesidades y preferencias. Esto le permite aplicar su propia lógica y coherencia al establecer las relaciones entre los vértices del grafo. La interactividad proporcionada permite al usuario comprender y participar activamente en la construcción y configuración del grafo, adaptándolo a sus objetivos específicos. Esta flexibilidad no solo facilita la personalización del grafo, sino que también fomenta una comprensión más profunda de su uso y funcionalidades.



El resultado de este ejemplo generó el siguiente gráfico de paradas de autobuses. Ahora, mediante el método de Búsqueda por Profundidad, examinaremos la coherencia de la red. Comenzando con el primer vértice del grafo agregado como referencia, verificaremos si todos los demás vértices están conectados y han sido visitados. Este proceso permitirá determinar si la red forma una estructura coherente, asegurando que todas las paradas de autobuses estén interconectadas de manera eficiente.



La última opción de búsqueda, seleccionada como Búsqueda por Profundidad, arrojó como resultado la conexión coherente de nodos visitados. Este proceso asegura que, a partir del primer vértice del grafo utilizado como punto de inicio, se exploran y conectan de manera sistemática todos los nodos subsiguientes. El resultado final es una red de nodos visitados y conectados de manera coherente, lo que confirma la integridad de la estructura del grafo.





Se proporciona el código y la configuración del botón correspondiente a la búsqueda por profundidad. Dicho código será presentado en el documento, evitando que su inclusión parezca forzada.

```
private void BtnBuscar_Click(object sender, EventArgs e)
{
    nodosVisitados.Clear();
    DFS(vertice.First()); // Iniciara DFS desde el primer verticexd
    this.Invalidate(); // Vuelve a dibujar el formulario luego de la
    busqueda
    button1.Visible = true;
    button2.Visible = true;
    //MessageBox.Show("Busqueda Finalizada");
}
private void DFS(Vertice inicio)
{
    nodosVisitados.Clear(); // Limpiara xd nodos visitados antes de la
    búsqueda
    DFSUtil(inicio);
}
private void DFSUtil(Vertice v)
{
    nodosVisitados.Add(v); // Marcara el vertice como visitadoxd
    this.Invalidate(); // Vuelve a dibujar el formulario para reflejar el
    cambio de color

    foreach (var vecino in vertice.Where(x => !nodosVisitados.Contains(x)
    && ExisteArista(x, v)))
    {
        DFSUtil(vecino);
    }
}
```



Conclusión

En el presente proyecto se abordó de manera exhaustiva elementos esenciales de programación, centrándose en la comprensión de la estructura de un árbol binario. Se destacó la importancia de este conocimiento no solo para comprender su funcionamiento, sino también para aplicaciones prácticas en el desarrollo de software.

Se incluyó el estudio de grafos, utilizando un caso práctico de paradas de autobuses. Se creó dinámicamente un grafo que permitía a los usuarios agregar vértices personalizados y establecer conexiones entre ellos. Esta aproximación consolidó tanto el entendimiento teórico de grafos como la resolución práctica de problemas mediante representaciones gráficas de relaciones entre entidades del mundo real.

El proyecto no solo se centró en sentar bases teóricas, sino también en proporcionar una experiencia práctica y dinámica. El objetivo era fortalecer la capacidad para aplicar estos conceptos en contextos reales de desarrollo de software.

Reflexión sobre aprendizajes durante la realización del proyecto

Gracias a la oportunidad brindada, pude aprender y experimentar no solamente sobre las estructuras de datos planteadas, sino también para el estudio de ciertas librerías de dibujo gráfico para poder plasmar de forma gráfica el funcionamiento de cada uno.

Si tomamos como ejemplo de reflexión en los grafos, mediante el caso de estudio de paradas de autobuses, permitió no solo consolidar el entendimiento teórico, sino también sumergirse en la resolución práctica de problemas. Se destacó la utilidad de las representaciones gráficas de relaciones entre entidades en situaciones del mundo real.