

A stylized city skyline is visible in the background, rendered in dark blue and purple silhouettes. The main title 'NEON' is centered within a rounded rectangular frame with a glowing pink border. Below the title, the subtitle 'ALGO-RHYTHM' is flanked by horizontal lines. The entire composition is surrounded by various geometric shapes like triangles, circles, and a diamond, some of which are glowing with a neon effect.

# NEON

—ALGO-RHYTHM—

01

# Estructuras de datos

# Lista

Mercury is the closest planet to the Sun and the smallest one in the Solar System—it's only a bit larger than the Moon

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

16 referencias
public class DoubleNode<T>
{
    3 referencias
    public T Data { get; set; }
    12 referencias
    public DoubleNode<T> Next { get; set; }
    9 referencias
    public DoubleNode<T> Previous { get; set; }

    2 referencias
    public DoubleNode(T data)
    {
        Data = data;
        Next = this;
        Previous = this;
    }
}

9 referencias
public class CircularDoublyLinkedList<T>
{
    private DoubleNode<T> head;
    private int count = 0;
}
```

# Cola

La cola fue usada para contener las notas

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

16 referencias
public class DoubleNode<T>
{
    3 referencias
    public T Data { get; set; }
    12 referencias
    public DoubleNode<T> Next { get; set; }
    9 referencias
    public DoubleNode<T> Previous { get; set; }

    2 referencias
    public DoubleNode(T data)
    {
        Data = data;
        Next = this;
        Previous = this;
    }
}

9 referencias
public class CircularDoublyLinkedList<T>
{
    private DoubleNode<T> head;
    private int count = 0;
```

# Cola de prioridad

La cola fue usada para ordenar los puntajes obtenidos

```
using System.Collections;
using System.Collections.Generic;
using UnityEngine;

16 referencias
public class DoubleNode<T>
{
    3 referencias
    public T Data { get; set; }
    12 referencias
    public DoubleNode<T> Next { get; set; }
    9 referencias
    public DoubleNode<T> Previous { get; set; }

    2 referencias
    public DoubleNode(T data)
    {
        Data = data;
        Next = this;
        Previous = this;
    }
}

9 referencias
public class CircularDoublyLinkedList<T>
{
    private DoubleNode<T> head;
    private int count = 0;
```

02

# Ordenamiento

# Insertion Sort

Eficaz para lista pequeñas y semi ordenadas

```
0 referencias
public void SortByDuration()
{
    for (int i = 1; i < songsList.Count; i++)
    {
        SongsSO key = songsList.Get(i);
        int j = i - 1;

        while (j ≥ 0 && songsList.Get(j).musicDurationInSeconds > key.musicDurationInSeconds)
        {
            songsList.Set(j + 1, songsList.Get(j));
            j--;
        }
        songsList.Set(j + 1, key);
    }

    UpdateMenu();
}
```