

G-VOICE

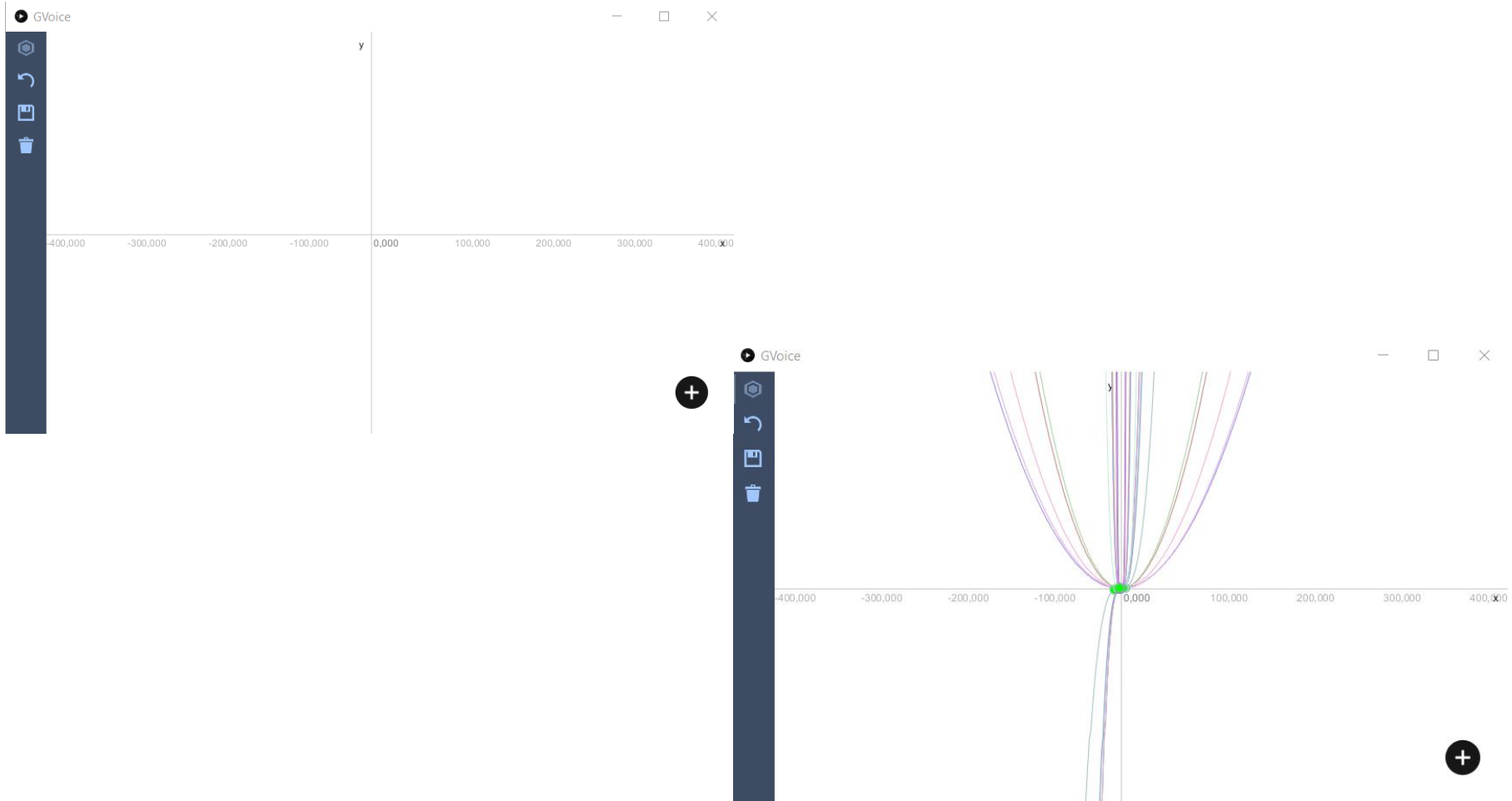
Morales Martínez Andrés, Ramírez Montes
Juan, Rodríguez Fuentes Miguel, Hernández
Triana Iván



UNIVERSIDAD
NACIONAL
DE COLOMBIA
SEDE BOGOTÁ



Avance Interfaz de Usuario





Nuevas Implementaciones

- Stack Boton Undo:

```
case 1:
    if (!g.inScreenStack.empty()) {
        g.inScreenStack.pop().nullify();
        g.generateImage();
    }
    break;
```



MinHeap:

```
RefQueue<PVector> getPC(Polinomio pol, GraphController graph){
    Heap h;
    Polinomio deripol = pol.derivate();
    h = new Heap(deripol, graph);
    PVector aux = new PVector();
    float[] points = deripol.getPoints(graph);
    for(int i = 0; i < points.length; i++){
        aux.x = i-(graph.getDimension().x)/2;
        aux.y = abs(deripol.y(aux.x));
        h.insertItem(new PVector(i-(graph.getDimension().x)/2, abs(deripol.y(aux.x))));
    }

    RefQueue<PVector> min;
    PVector aux2;
    min = new RefQueue<PVector>();
    aux2 = h.removeMin();
    while((aux2.y - 0) <= pow(0.1, 1/pol.grado)){
        min.enqueue(aux2);
        System.out.printf("removeMin ... aux.x: %.2f, aux.y: %.2f\n", aux2.x, aux2.y);
        aux2 = h.removeMin();
    }
    int cant = min.numInside();
    System.out.printf("%d", cant);

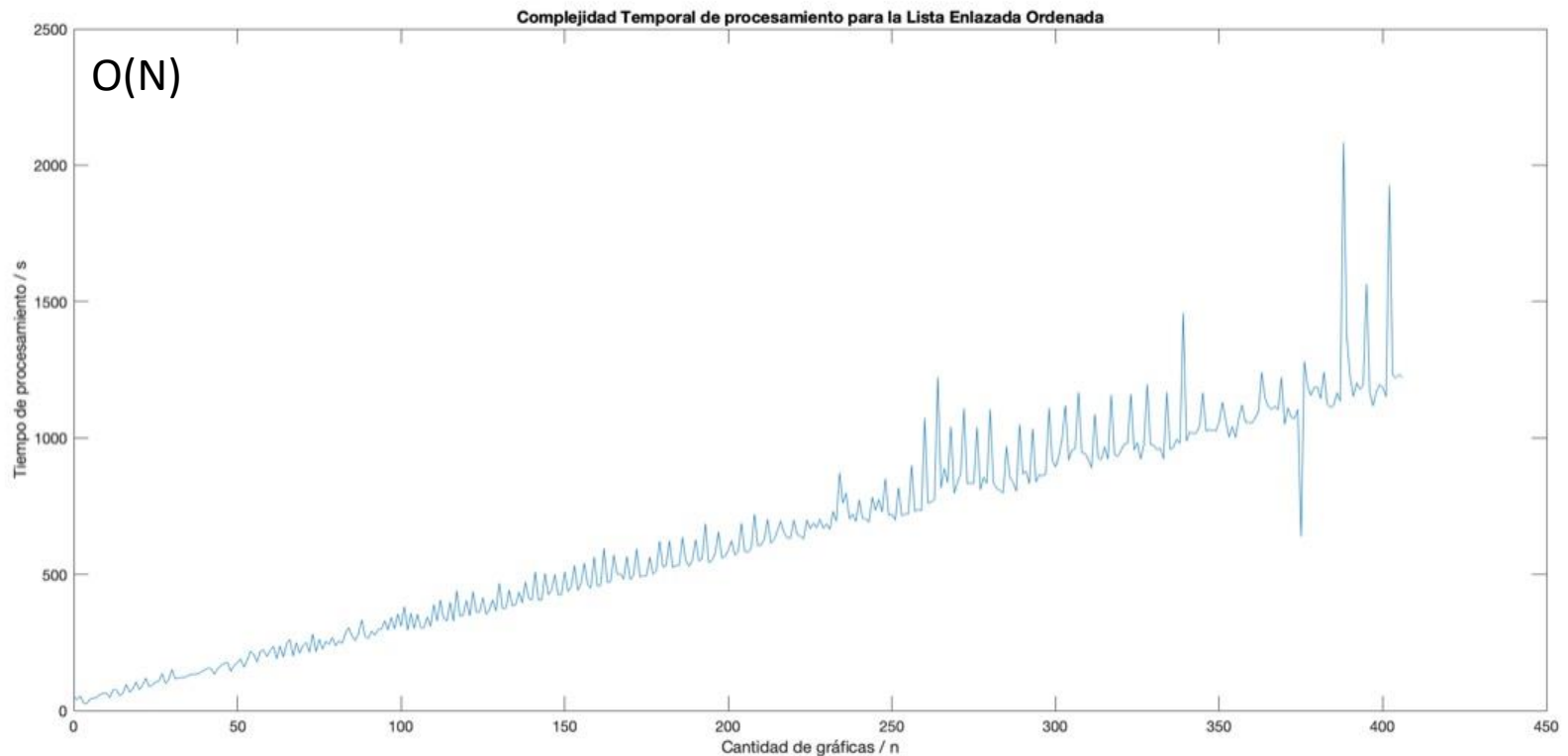
    return min;
}
```



Expression tree:

```
void calc(Node root) {  
    Node p = root;  
    while (p.chlds[0].chlds[0] != null) {  
        p = p.chlds[0];  
    }  
  
    if (p.val == "+") {  
        p.val = String.valueOf(Integer.parseInt(p.chlds[0].val) + Integer.parseInt(p.chlds[1].val));  
    } else if (p.val == "*") {  
        p.val = String.valueOf(Integer.parseInt(p.chlds[0].val) * Integer.parseInt(p.chlds[1].val));  
    } else if (p.val == "/") {  
        p.val = String.valueOf(Integer.parseInt(p.chlds[0].val) / Integer.parseInt(p.chlds[1].val));  
    } else if (p.val == "-") {  
        p.val = String.valueOf(Integer.parseInt(p.chlds[0].val) - Integer.parseInt(p.chlds[1].val));  
    }  
  
    p.chlds[0] = null;  
    p.chlds[1] = null;  
  
    this.printTree();  
    println("");  
  
    if (root.chlds[0] != null)  
        this.calc(root);  
}  
}
```

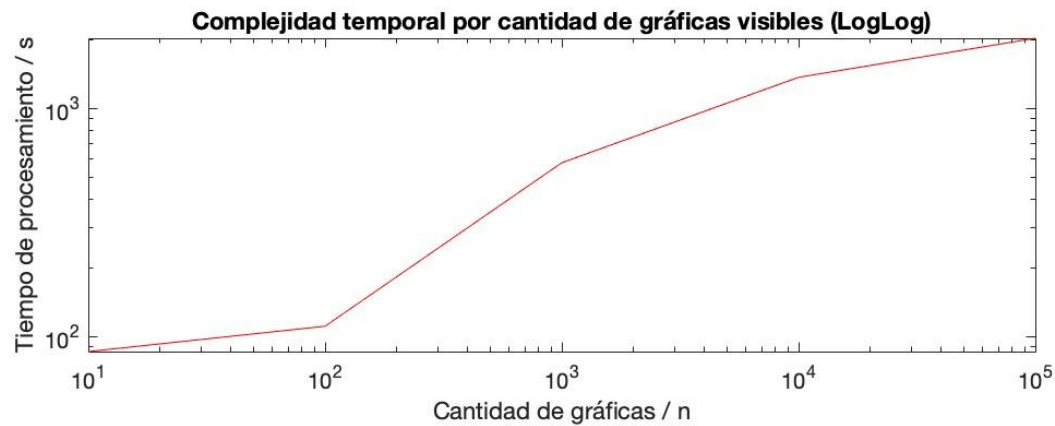
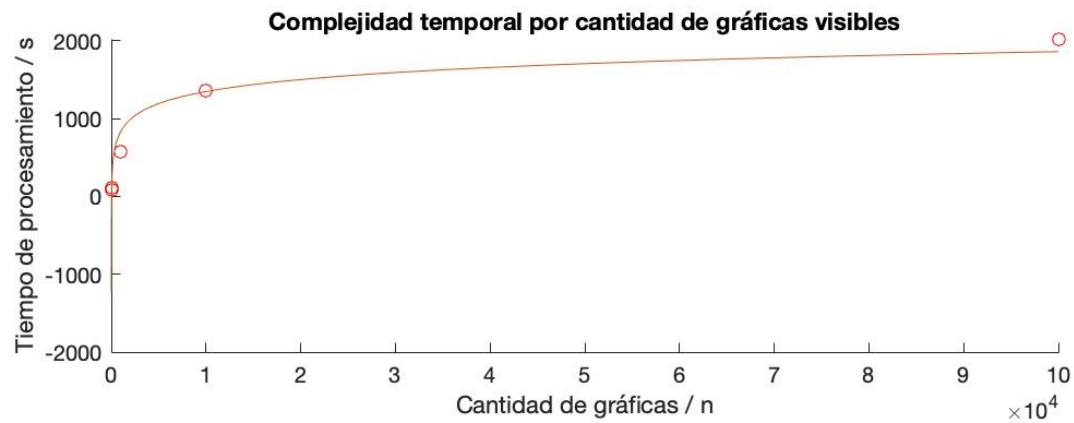
Pruebas y análisis comparativo del uso de las estructuras de datos





Complejidad Heap:

$\log(N)$



Conclusiones

- Se evidencio que el Heap es muy eficiente al organizar los datos
- Aunque el heap proporciona una solución muy eficiente al ser esta una aplicación grafica esta colapsa los recursos del computador luego de las 100.000 graficas.

