

Métodos de heurística Utilizados

```
private int Heuristica1(String estado, String objetivo2)
{
    int resultado = 0;

    for (int i = 0; i < estado.length(); i++) {
        if (estado.charAt(i) != objetivo.charAt(i)) {
            resultado++;
        }
    }
    return resultado;
}

private int Heuristica2(String estado, String objetivo2)
{
    int resultado = 0;
    int num1, num2;

    for (int i = 0; i < estado.length(); i++) {
        num1 = Character.getNumericValue(estado.charAt(i));
        if (estado.charAt(i) == ' ') {
            num1 = 0;
        }

        num2 = Character.getNumericValue(objetivo2.charAt(i));
        if (estado.charAt(i) == ' ') {
            num2 = 0;
        }

        resultado += Math.abs(num1 - num2);
    }
    return resultado;
}

private int Heuristica3(String estado, String objetivo)
{
    int resultado = 0;
    int tamaño = (int) Math.sqrt(estado.length());

    for (int i = 0; i < estado.length(); i++) {
        char num = estado.charAt(i);
```

```

        if (num != '0') { // Ignorar la ficha vacía
            int j = objetivo.indexOf(num);
            int row = i / tamaño, col = i % tamaño;
            int targetRow = j / tamaño, targetCol = j % tamaño;
            resultado += Math.abs(row - targetRow) + Math.abs(col -
targetCol);
        }
    }
    return resultado;
}

```

Implementación de los métodos

```

public void busquedaPorAnchuraHeuristica(){
    Nodo nodoActual = raiz;
    Collection<String> estadosVisitados = new ArrayList<String>();
    PriorityQueue<Nodo> estadosPorVisitar = new PriorityQueue<Nodo>();
    long inicioTiempo = System.currentTimeMillis();
    while(!nodoActual.getEstado().equals(objetivo))
    {
        estadosVisitados.add(nodoActual.getEstado());
        //Generar a los Nodos Hijos
        Collection<String> hijos = nodoActual.generaHijos();    //<-- Cada
Equipo tiene que ingeniarselas para crear este metodo!
        for (String hijo : hijos) {
            if(!estadosVisitados.contains(hijo))
            {
                //System.out.println("---Metiendo nuevo Nodo");
                //Crear nuevo Nodo.
                Nodo nHijo = new Nodo(hijo);
                nHijo.costo = Heuristica1(nHijo.getEstado(), objetivo);
                nHijo.costo = Heuristica2(nHijo.getEstado(), objetivo);
                nHijo.costo = Heuristica3(nHijo.getEstado(), objetivo);

                nHijo.setPadre(nodoActual);
                estadosPorVisitar.add(nHijo);
            }
        }
        nodoActual = estadosPorVisitar.poll();
        a.imprimeSolucion(raiz, nodoActual);
    }
    long finTiempo = System.currentTimeMillis();
}

```

```
long duracion = (finTiempo - inicioTiempo);

System.out.println("YA SE ENCONTRO EL NODO OBJETIVO");
System.out.println(nodoActual.getEstado());

System.out.println("Tiempo de ejecución en milisegundos: " + duracion);

}
```