

**Universidad La Salle**



**INGENIERIA DE SOFTWARE**

**CURSO:**

**ANÁLISIS Y DISEÑO DE ALGORITMOS**

**DOCENTE:**

**CRISTIAN LÓPEZ DEL ALAMO**

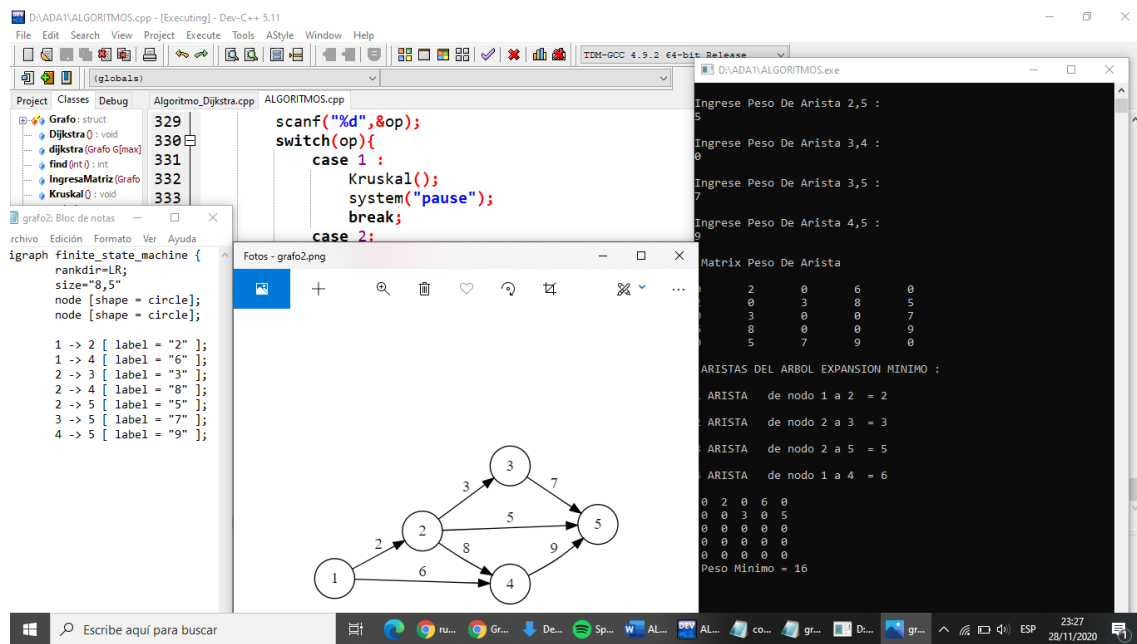
**NOMBRE:**

**GABRIELA PACCO HUAMANI**

**AREQUIPA- PERÚ**

**2021**

## KRUSKALL:



The screenshot shows a C++ IDE with the following components:

- Project Explorer:** Shows a project named 'grafo2' with files 'Bloc de notas', 'finite\_state\_machine', and 'grafo2.png'.
- Code Editor:** Displays the implementation of Kruskal's algorithm in 'ALGORITMO\_KRUSKAL.cpp'. The code includes a function to find the minimum spanning tree.
- Output Window:** Shows the input edges and the resulting minimum spanning tree matrix.
- Diagram:** A graph with 5 nodes (1, 2, 3, 4, 5) and weighted edges. The edges are: (1,2) weight 2, (1,4) weight 6, (2,3) weight 3, (2,4) weight 8, (2,5) weight 5, (3,5) weight 7, (4,5) weight 9.

**Matrix Peso De Arista:**

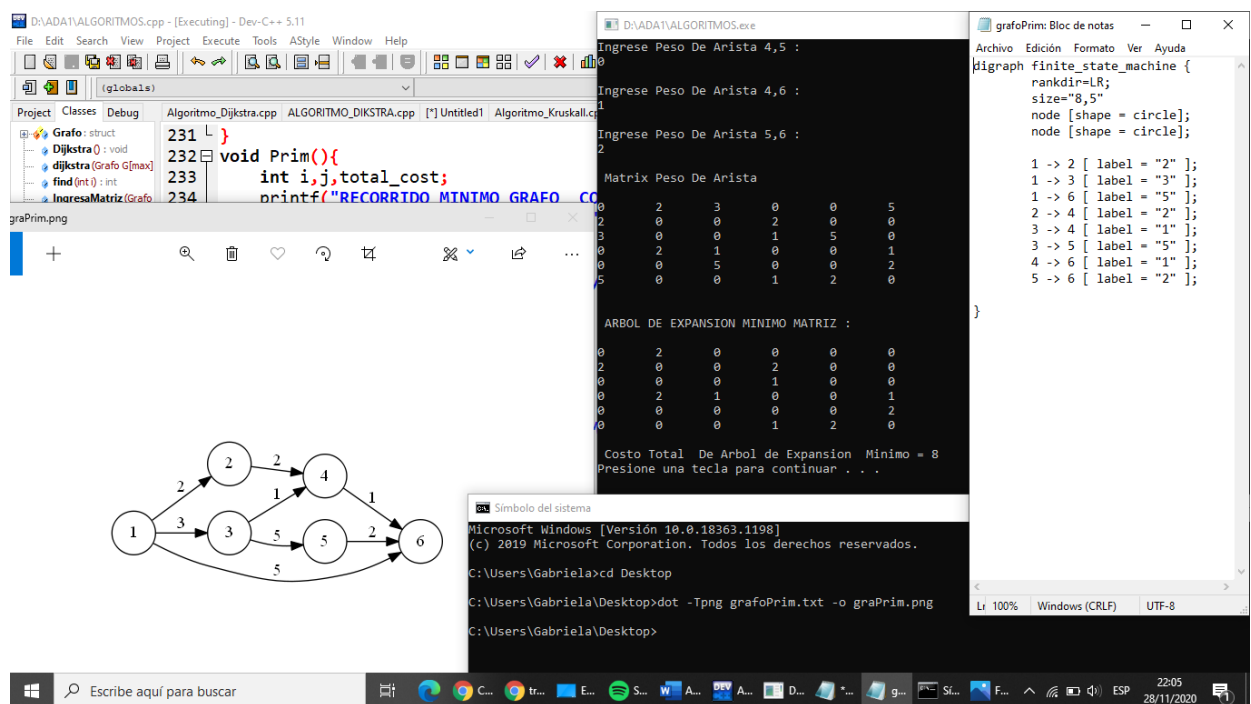
2	0	6	0	0
0	3	8	5	0
3	0	0	7	0
8	0	0	0	9
5	7	9	0	0

**ARISTAS DEL ARBOL EXPANSION MINIMO:**

- ARISTA de nodo 1 a 2 = 2
- ARISTA de nodo 2 a 3 = 3
- ARISTA de nodo 2 a 5 = 5
- ARISTA de nodo 1 a 4 = 6

**Peso Minimo = 16**

## PRIM:



The screenshot shows a C++ IDE with the following components:

- Project Explorer:** Shows a project named 'grafo2' with files 'Bloc de notas', 'finite\_state\_machine', and 'grafo2.png'.
- Code Editor:** Displays the implementation of Prim's algorithm in 'ALGORITMO\_PRIM.cpp'. The code includes a function to find the minimum spanning tree.
- Output Window:** Shows the input edges and the resulting minimum spanning tree matrix.
- Diagram:** A graph with 6 nodes (1, 2, 3, 4, 5, 6) and weighted edges. The edges are: (1,2) weight 2, (1,3) weight 3, (2,4) weight 2, (3,4) weight 1, (3,5) weight 5, (4,5) weight 1, (5,6) weight 2.

**Matrix Peso De Arista:**

2	3	0	0	0	5
0	0	2	0	0	0
0	0	0	1	5	0
2	1	0	0	0	1
0	5	0	0	0	2
5	0	0	1	2	0

**ARBOL DE EXPANSION MINIMO MATRIZ:**

0	2	0	0	0	0
2	0	0	2	0	0
0	0	0	1	0	0
0	2	1	0	0	1
0	0	0	0	0	2
0	0	0	1	2	0

**Costo Total De Arbol de Expansion Minimo = 8**  
Presione una tecla para continuar . . .

**Simbolo del sistema:**

```
Microsoft Windows [Versión 10.0.18363.1198]
(c) 2019 Microsoft Corporation. Todos los derechos reservados.

C:\Users\Gabriela>cd Desktop
C:\Users\Gabriela\Desktop>dot -Tpng grafoPrim.txt -o grafoPrim.png
C:\Users\Gabriela\Desktop>
```

## DIJKSTRA:

Dev-C++ IDE showing the execution of the Dijkstra algorithm. The code is in `Algoritmo_Dijkstra.cpp`.

```
112 Graph graph;
113
114 cout << "Algoritmo Dijkstra" << endl;
115
116 while (!out)
117 {
118     programa.Detallar_Grafo(graph); // cantidad v
119     programa.Cargar_Grafo(graph); // cargar aristas
```

The program prompts for input:

```
2
3 Inserte Origen:
4 1
5 Inserte Destino:
6 5
7 Inserte el Costo de la Arista:
8 5
9 Inserte Origen:
10 2
11 Inserte Destino:
12 4
13 Inserte el Costo de la Arista:
14 1
15 Inserte Origen:
16 0
17 Inserte Destino:
18 5
19 Inserte el Costo de la Arista:
20 4
21 Inserte Vertice Inicial:
22 0
23 Inserte Vertice Final:
24 5
25 Distancia del Camino: 14
26 Salir? (T/F)
```

The graph visualization shows the following structure:

```
graph LR
    0((0)) -- 3 --> 1((1))
    0((0)) -- 4 --> 2((2))
    1((1)) -- 6 --> 3((3))
    1((1)) -- 5 --> 4((4))
    2((2)) -- 4 --> 4((4))
    3((3)) -- 7 --> 5((5))
    4((4)) -- 12 --> 5((5))
```

The Windows Command Prompt shows the execution of the program:

```
Microsoft Windows [Versión 10.0.18363.1198]
(c) 2019 Microsoft Corporation. Todos los derechos reservados.

C:\Users\Gabriela>cd Desktop
C:\Users\Gabriela\Desktop>dot -Tpng grafo.txt -o grafo.png
C:\Users\Gabriela\Desktop>
```