

Pseudocódigo Primm

```
void Primm (vector<vector<int>> conexiones, int inicio) {
    vector<bool> visitado
    int costoTotal = 0
    int edges = 0 // Contador para v-1 edges
    y = conexiones.size()
    // Marcar nodo inicial como visitado
    visitado[inicio] = true

    // Mientras no tengamos v-1 edges
    while (edges < y-1) {
        int costoMin = infinity, int idx = -1
        int nodoActual
        int nodoSig

        for (int i=0; i < conexiones.size(); i++) {
            // Indices de nodos           // Conexión actual
            int idx1 = nodos[n1]          int n1 = conexiones[i][0]
            int idx2 = nodos[n2]          int n2 = conexiones[i][1]
            int costo = conexiones[i][2]

            // Revisar si se conecta a un nodo no visitado
            if ((visitado[idx1] & !visitado[idx2]) || (visitado[idx2] & !visitado[idx1])) {
                // Si el costo es menor actualizar mínimo
                if (costo < costoMin) {
                    costoMin = costo
                    idx = i // Para agregar MST
                }
            }
        }

        if (idx != -1) { // Si se encontró un edge válido
            T.push_back (conexiones[idx])
            visitado[nodos[NodoSig]] = true
            edges++
        } else break;
    }

    print T, costoTotal
}
```