

## Pseudocódigo Prim

```
void Prim (vector<vector<int,int,int>> conexiones, int inicio) {  
    vector<bool> visitado  
    int costoTotal = 0  
    int edges = 0 // Contador para v-1 edges  
    v = conexiones.size()  
    // Marcar nodo inicial como visitado  
    visitado[inicio] = true  
  
    // Mientras no tengamos v-1 edges  
    while (edges < v-1) {  
        int costoMin = ∞, int idx = -1  
        int nodoActual  
        int nodoSig  
  
        for (int i = 0; i < conexiones.size(); i++) {  
            // Indices de nodos  
            int idx1 = nodos[n1]  
            int idx2 = nodos[n2]  
            // Conexión actual  
            int n1 = conexiones[i][0]  
            int n2 = conexiones[i][1]  
            int costo = conexiones[i][2]  
  
            // Revisar si se conecta a un nodo no visitado  
            if (!visitado[idx1] && !visitado[idx2]) {  
                // Si el costo es menor actual, por mínimo  
                if (costo < costoMin) {  
                    costoMin = costo  
                    idx = i // Para agregar a MST  
                }  
  
                // Determinar sig. nodo  
                if (visitado[idx1]) {  
                    nodoActual = n1  
                    nodoSig = n2  
                } else {  
                    nodoActual = n2  
                    nodoSig = n1  
                }  
            }  
        }  
  
        if (idx != -1) { // Si se encontró un edge válido  
            T.push_back(conexiones[idx])  
            visitado[nodos[nodoSig]] = true  
            edges++  
        } else break;  
    }  
    print T, costoTotal  
}
```