

Pseudocódigo Kmp

Función Prefijo (n)

```
Vector<int> prefijo (string patron) {
    vector<int> lista (patron.size()) // Para almacenar valores de n
    int len = 0 // Valores que coinciden de ambos lados

    for (int i=0 ; i < patron.size(); i++) {
        if (i==0) lista[i] = 0 // El primer elemento de n siempre es 0
        else {
            // Mientras no coincidan, retroceder
            while (len > 0 && patron[i] != patron[len]) {
                len = lista[len-1]
            }
            // Si son iguales, incrementar len
            if (patron[i] == patron[len]) {
                len++
            }
            lista[i] = len // Guardar longitud
        }
    }
    return lista
}
```

Función KMP → Índices de inicio : fin

```

vector<pair<int,int>> kmp ( string texto, string patron ) {
    // Primero obtenemos la función prefijo del patrón
    vector<int> lista = prefija ( patron )
    // Contadores para texto, patrón           // Índices (resultado)
    int i=0, j=0                                vector<pair<int,int>> indices
    while ( i < texto.size() ) {
        if ( texto [i] == patron [j+1] ) { // Si i==j+1, ambos avanzan
            i++, j++
        }
        if ( j == patron.size()-1 ) { // Si j==patrón, regresar res
            indices.push_back ( i - patron.size() + 1, i )
            j = lista [j]
        }
        if ( texto [i] != patron [j+1] ) {
            if ( j == 0 ) i++
            else j = lista [j]
        }
    }
    return indices
}

```