# Análisis del Punto 7 - TP Listas Simples

Este punto presenta tres funciones recursivas en C++ que operan sobre listas simplemente enlazadas. Vamos a analizar qué hace cada función, cómo lo hace y cuál es su propósito.

## Función 1: enigma

void enigma(Nodo\* nodo)   
{   
 if(nodo != nullptr) {  
 if(nodo->getSiguiente() == nullptr)   
 cout << nodo->getDato() << endl;   
 else {   
 enigma(nodo->getSiguiente());   
 cout << nodo->getDato() << endl;   
 }   
 }   
}

✅ ¿Qué hace?  
- Recorre la lista hasta el último nodo.  
- Luego, imprime los datos desde el último hasta el primero (orden inverso).

🎯 Propósito:  
- Mostrar los valores de la lista en orden inverso, sin modificar la estructura de la lista.

🧠 Ejemplo:  
Para la lista:  
10 → 20 → 30 → nullptr  
Salida:  
30  
20  
10

## Función 2: misterio

int misterio(Nodo\* nodo)   
{   
 if(nodo == nullptr)   
 return 0;   
 else   
 return misterio(nodo->getSiguiente()) + 1;   
}

✅ ¿Qué hace?  
- Cuenta cuántos nodos tiene la lista enlazada.

🎯 Propósito:  
- Obtener la cantidad total de elementos de la lista.

🧠 Ejemplo:  
Para la lista:  
5 → 6 → 7 → nullptr  
Devuelve: 3

## Función 3: desconocido

Nodo\* desconocido(Nodo\* nodo)   
{   
 if(nodo == nullptr || nodo->getSiguiente() == nullptr)   
 return nodo;   
 else   
 return desconocido(nodo->getSiguiente());   
}

✅ ¿Qué hace?  
- Recorre la lista hasta encontrar el último nodo, y lo devuelve.

🎯 Propósito:  
- Obtener un puntero al último nodo de la lista.

🧠 Ejemplo:  
Para la lista:  
2 → 4 → 6 → nullptr  
Devuelve un puntero al nodo que contiene 6.

## Conclusión

|  |  |  |
| --- | --- | --- |
| Función | Acción que realiza | Propósito principal |
| enigma | Imprime los valores en orden inverso | Mostrar lista al revés sin modificarla |
| misterio | Cuenta los nodos | Obtener la cantidad de elementos |
| desconocido | Devuelve el último nodo | Acceder al nodo final de la lista |