

Reporte Técnico: Decodificador de Protocolo PRT-7

Autor: David Alejandro Ibarra Castañeda **Fecha:** 6 de Noviembre de 2025

Catedrático: Dr. Said Polanco Martagon

2. ¿Qué hace este programa?

El proyecto consiste en dos programas que trabajan juntos:

1. **Arduino (transmisor):** Envía letras y comandos a través del puerto serial
2. **Programa en C++ (receptor):** Recibe esas letras, las decodifica y arma el mensaje secreto

La idea es similar a una máquina de cifrado antigua, donde las letras se reemplazan por otras dependiendo de cómo esté configurado el sistema en ese momento.

3. ¿Cómo funciona este programa?

3.1 El Rotor de Mapeo (disco de cifrado)

Esto funciona como un disco con todas las letras del alfabeto (A-Z) escritas alrededor. Este disco puede girar.

- Cuando el disco está en posición inicial, la letra **A** se queda como **A**, la **B** como **B**, etc.
- Si giramos el disco 5 posiciones hacia adelante, entonces:
 - La **A** ahora se convierte en **F**
 - La **B** se convierte en **G**
 - Y así sucesivamente

Este disco está programado como una **lista circular**, donde cada letra está conectada con la siguiente y la última con la primera, formando un círculo.

3.2 Tipos de mensajes que envía el Arduino

El Arduino envía dos tipos de instrucciones:

a) Mensajes LOAD (L,X)

- Formato: **L**, **H** significa "carga la letra H"
- El programa toma esa letra y la decodifica según cómo esté girado el disco en ese momento
- La letra decodificada se guarda en una lista para formar el mensaje final

b) Mensajes MAP (M,N)

- Formato: **M**, **5** significa "gira el disco 5 posiciones adelante"
- Formato: **M**, **-5** significa "gira el disco 5 posiciones atrás"
- Esto cambia cómo se van a decodificar las siguientes letras

c) Mensaje FIN

- Cuando el Arduino envía "FIN", el programa para de recibir datos y muestra el mensaje completo

3.3 Ejemplo práctico con el código

El Arduino envía esta secuencia:

SISTEMA PRT-7 ACTIVO

L,H → decodifica "H" (sin rotación) = H
L,E → decodifica "E" (sin rotación) = E
L,L → decodifica "L" (sin rotación) = L
L,L → decodifica "L" (sin rotación) = L
L,O → decodifica "O" (sin rotación) = O
L, → decodifica " " (espacio) = espacio
M,5 → ¡gira el disco 5 posiciones!
L,W → decodifica "W" (con rotación +5) = B
L,O → decodifica "O" (con rotación +5) = T
M,-5 → regresa el disco a posición original
L,R → decodifica "R" (sin rotación) = R
L,L → decodifica "L" (sin rotación) = L
L,D → decodifica "D" (sin rotación) = D
FINn

Mensaje final: HELLO WORLD (no muestra el world)

4. Estructuras de datos utilizadas

4.1 Lista Circular (para el rotor)

Se usó una lista donde cada letra apunta a la siguiente y la última apunta de regreso a la primera. Esto permite "girar" el disco simplemente cambiando desde dónde empezamos a contar.

4.2 Lista Doblemente Enlazada (para los caracteres decodificados)

Cada letra decodificada se guarda en orden, donde cada letra conoce a la anterior y a la siguiente.

5. Características del programa

5.1 Compatibilidad multiplataforma

El programa funciona tanto en:

- **Linux/Mac** (puertos /dev/ttyUSB0, /dev/ttyACM0, etc.)

5.2 Uso de programación orientada a objetos

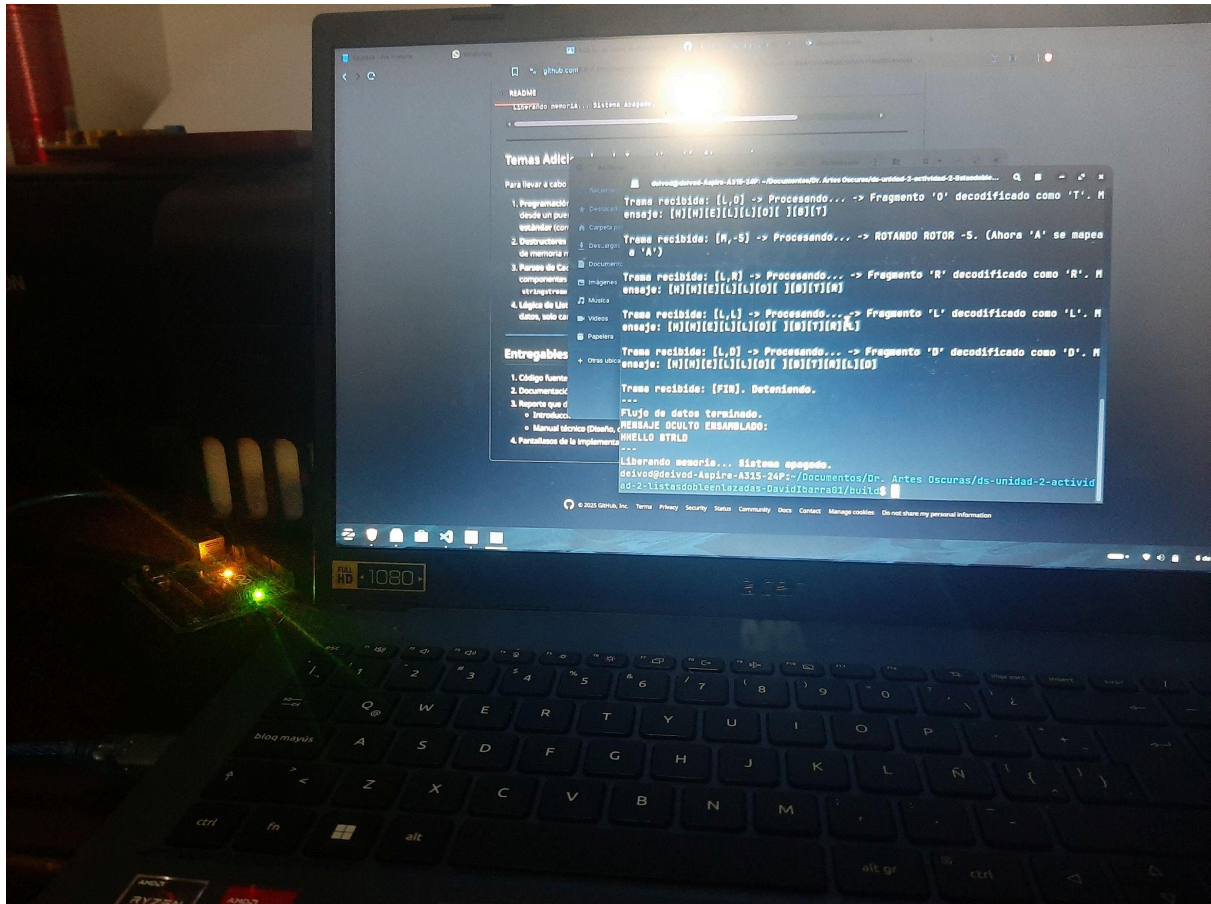
Se crearon diferentes "clases" (plantillas) para organizar el código:

- **TramaBase:** Clase principal de la que heredan los otros tipos de mensajes
- **TramaLoad:** Mensajes que contienen letras para decodificar
- **TramaMap:** Mensajes que indican cuánto girar el rotor
- **RotorDeMapeo:** El disco de cifrado que puede girar
- **ListaDeCarga:** La lista donde se guardan las letras decodificadas

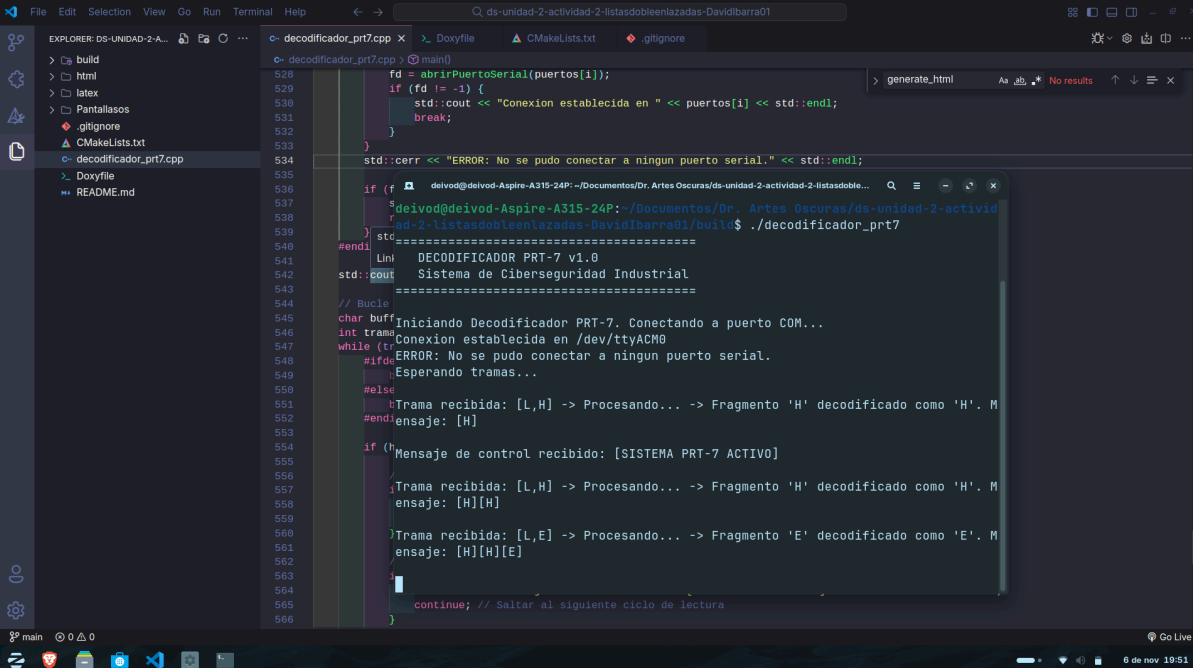
5.3 Manejo de memoria

El programa crea y destruye correctamente todos los objetos que usa, evitando desperdiciar memoria.

6. Resultados:



```
deivod@deivod-Aspire-A315-24P: ~/Documentos/Dr. Artes Oscuras/ds-unidad-2-actividad-2-listasdoblee...  
Trama recibida: [L,0] -> Procesando... -> Fragmento '0' decodificado como 'T'. M  
ensaje: [H][E][L][L][O][ ][B][T]  
  
Trama recibida: [M,-5] -> Procesando... -> ROTANDO ROTOR -5. (Ahora 'A' se mapea  
a 'A')  
  
Trama recibida: [L,R] -> Procesando... -> Fragmento 'R' decodificado como 'R'. M  
ensaje: [H][E][L][L][O][ ][B][T][R]  
  
Trama recibida: [L,L] -> Procesando... -> Fragmento 'L' decodificado como 'L'. M  
ensaje: [H][E][L][L][O][ ][B][T][R][L]  
  
Trama recibida: [L,D] -> Procesando... -> Fragmento 'D' decodificado como 'D'. M  
ensaje: [H][E][L][L][O][ ][B][T][R][L][D]  
  
Trama recibida: [FIN]. Deteniendo.  
---  
Flujo de datos terminado.  
MENSAJE OCULTO ENSAMBLADO:  
HELLO BTRLD  
---  
Liberando memoria... Sistema apagado.  
deivod@deivod-Aspire-A315-24P:~/Documentos/Dr. Artes Oscuras/ds-unidad-2-actividad-2-listasdoblee.../build$
```



```
528 fd = abrirPuertoSerial(puertos[i]);
529 if (fd != -1) {
530     std::cout << "Conexion establecida en " << puertos[i] << std::endl;
531     break;
532 }
533 }
534 std::cerr << "ERROR: No se pudo conectar a ningun puerto serial." << std::endl;
535
536 if (f
537     deivod@deivod-Aspire-A315-24P: ~/Documentos/Dr. Artes Oscuras/ds-unidad-2-listasdob
538     ad-2-listasdobleenlazadas-DavidIbarra01/built$ ./decodificador_prt7
539 }
540 #endl
541 Lin1
542 std::cout << "DECODIFICADOR PRT-7 v1.0\n";
543     Sistema de Ciberseguridad Industrial
544     =====
545 // Bucle
546 char buff;
547 int trama;
548 while (true) {
549     if (fd == -1) {
550         std::cerr << "ERROR: No se pudo conectar a ningun puerto serial.\n";
551         continue;
552     }
553     if (fd != -1) {
554         Trama recibida: [L,H] -> Procesando... -> Fragmento 'H' decodificado como 'H'. M
555         ensaje: [H]
556     }
557     if (fd != -1) {
558         Mensaje de control recibido: [SISTEMA PRT-7 ACTIVO]
559     }
560     if (fd != -1) {
561         Trama recibida: [L,H] -> Procesando... -> Fragmento 'H' decodificado como 'H'. M
562         ensaje: [H][H]
563     }
564     if (fd != -1) {
565         Trama recibida: [L,E] -> Procesando... -> Fragmento 'E' decodificado como 'E'. M
566         ensaje: [H][H][E]
567     }
568     continue; // Saltar al siguiente ciclo de lectura
569 }
```

```

File Edit Selection View Go Run Terminal Help
c- decodificador_pt7.cpp x Doxyfile
Captura de pantalla Ahora mismo
Captura de pantalla realizada
Puede pegar la imagen desde el portapapeles.
generate_html Aa No results

EXPLORER: DS-UNIDAD-2-A...
> build
> html
> latex
> Pantallasos
> .gitignore
> CMakeLists.txt
> decodificador_pt7.cpp
> Doxyfile
> README.md

528 fd = abrirPuertoSerial(puertos[i]);
529 if (fd != -1) {
530     std::cout << "Conexion establecida en " << puertos[i] << std::endl;
531     break;
532 }
533
534 std::cerr << "ERROR: No se pudo conectar a ningun puerto serial." << std::endl;
535
536 if (f
537     mensaje: [H][H]
538 }
539 Trama recibida: [L,E] -> Procesando... -> Fragmento 'E' decodificado como 'E'. M
540 mensaje: [H][H][E]
541 #endif
542 Trama recibida: [L,L] -> Procesando... -> Fragmento 'L' decodificado como 'L'. M
543 mensaje: [H][H][E][L]
544 // Bucla
545 char buff
546 int trama
547 while (t
548     #ifd
549     #else
550 Trama recibida: [L,0] -> Procesando... -> Fragmento '0' decodificado como '0'. M
551 mensaje: [H][H][E][L][0]
552 #endif
553 Trama recibida: [L, ] -> Procesando... -> Fragmento ' ' decodificado como ' '. M
554 mensaje: [H][H][E][L][ ][ ]
555 if (
556 Trama recibida: [M,5] -> Procesando... -> ROTANDO ROTOR +5. (Ahora 'A' se mapea
557 a 'F')
558 Trama recibida: [L,W] -> Procesando... -> Fragmento 'W' decodificado como 'B'. M
559 mensaje: [H][H][E][L][ ][ ][ ][B]
560
561 continue; // Saltar al siguiente ciclo de lectura
562
563
564
565
566
main 0 0 0
Go Live
6 de nov 19:51

```

```

File Edit Selection View Go Run Terminal Help
c- decodificador_pt7.cpp x Doxyfile
CMakeLists.txt .gitignore
generate_html Aa No results

EXPLORER: DS-UNIDAD-2-A...
> build
> html
> latex
> Pantallasos
> .gitignore
> CMakeLists.txt
> decodificador_pt7.cpp
> Doxyfile
> README.md

528 fd = abrirPuertoSerial(puertos[i]);
529 if (fd != -1) {
530     std::cout << "Conexion establecida en " << puertos[i] << std::endl;
531     break;
532 }
533
534 std::cerr << "ERROR: No se pudo conectar a ningun puerto serial." << std::endl;
535
536 if (f
537     mensaje: [L,0] -> Procesando... -> Fragmento '0' decodificado como 'T'. M
538 mensaje: [H][H][E][L][0][ ][B][T]
539 #endif
540 Trama recibida: [M,-5] -> Procesando... -> ROTANDO ROTOR -5. (Ahora 'A' se mapea
541 a 'A')
542 std::cout << 'A'
543
544 // Bucla
545 char buff
546 int trama
547 while (t
548     #ifd
549     #else
550 Trama recibida: [L,R] -> Procesando... -> Fragmento 'R' decodificado como 'R'. M
551 mensaje: [H][H][E][L][ ][ ][ ][B][T][R]
552 #endif
553 Trama recibida: [L,D] -> Procesando... -> Fragmento 'D' decodificado como 'D'. M
554 mensaje: [H][H][E][L][ ][ ][ ][B][T][R][D]
555 #endif
556 Trama recibida: [FIN]. Deteniendo.
557 ---
558 Flujo de datos terminado.
559 MENSAJE OCULTO ENSAMBLADO:
560 HHHELLO BTRLDD
561 ---
562 Liberando memoria... Sistema apagado.
563 deivod@deivod-Aspire-A315-24P: ~/Documentos/Dr. Artes Oscuras/ds-unidad-2-actividad-2-listasdoblenlazaradas-DavidIbarra01/build$
564 continue; // Saltar al siguiente ciclo de lectura
565
566
main 0 0 0
Go Live
6 de nov 19:51

```