

# 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 Dblemente 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:

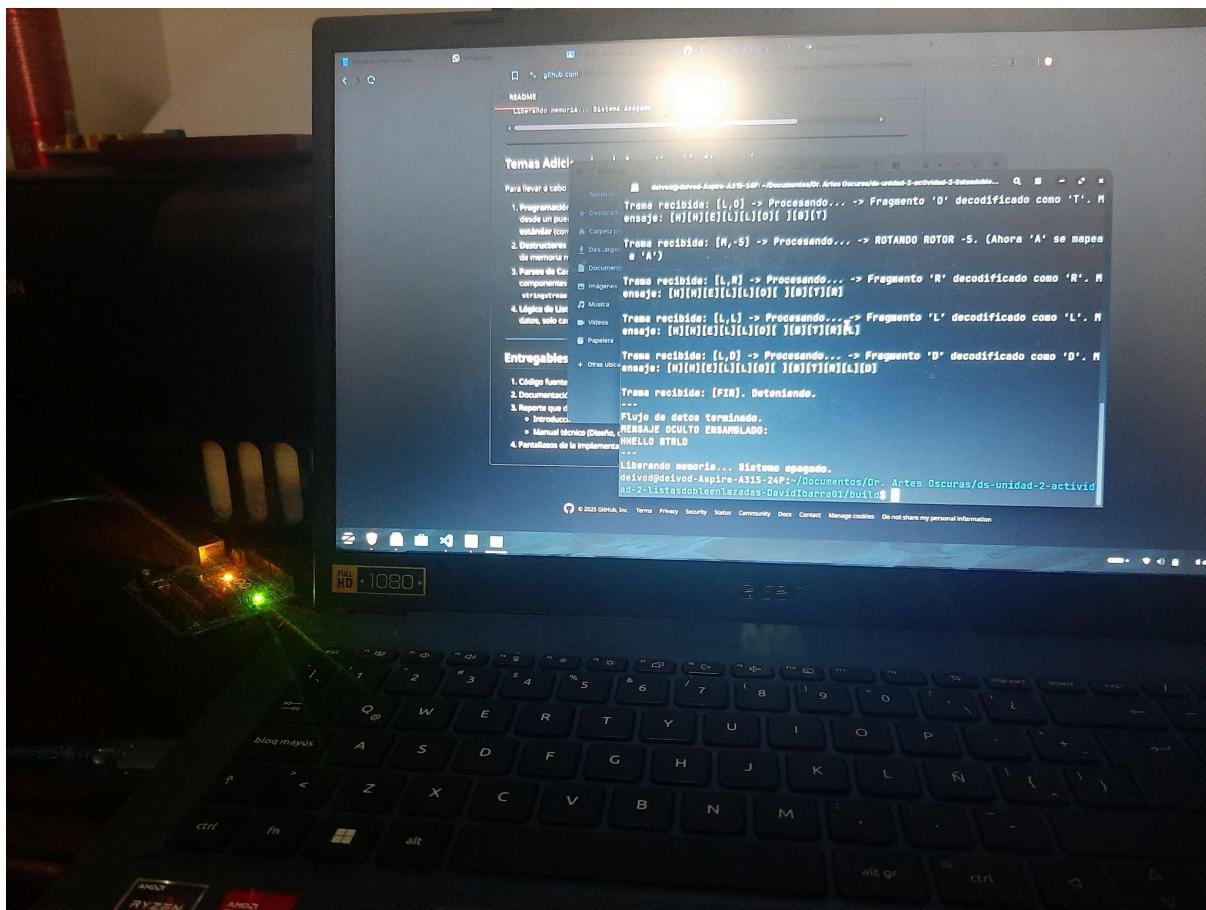
- **TramaBase**: Clase principal de la que heredan los otros tipos de mensajes
- **TramaLoad**: Mensajes que contienen letras para decodificar
- **TramaMap**: Mensajes que indican cuánto girar el rotor
- **RotorDeMapeo**: El disco de cifrado que puede girar
- **ListaDeCarga**: La lista donde se guardan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

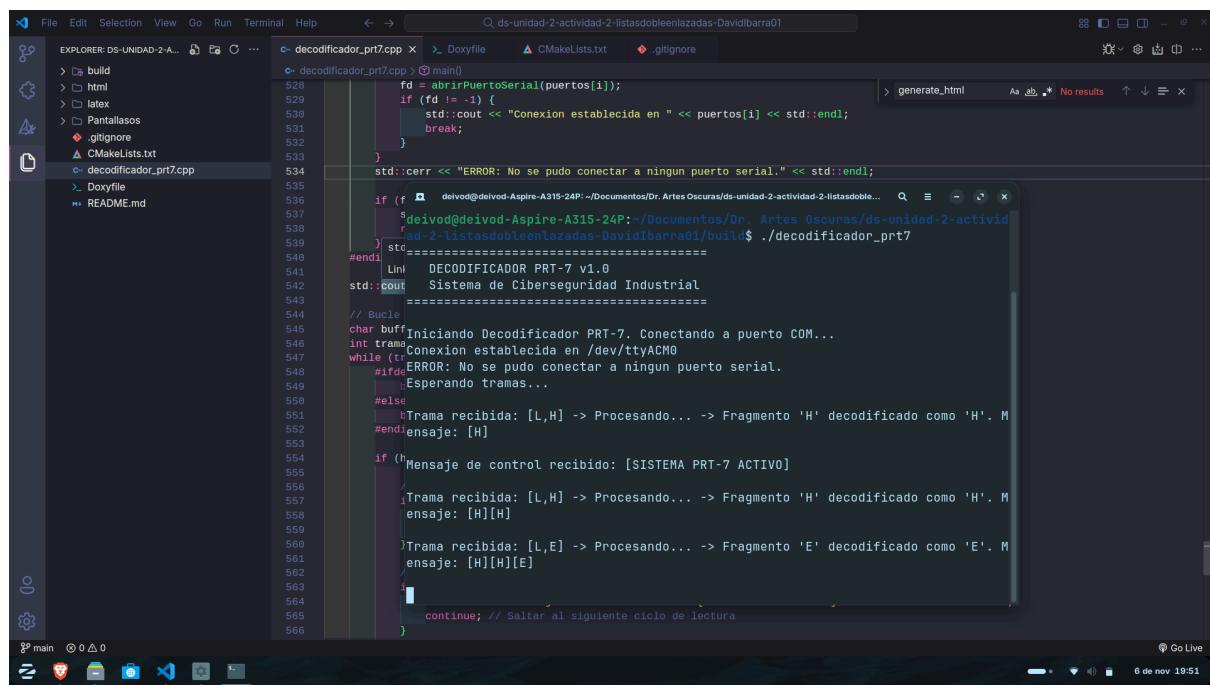
Trama recibida: [L,L] -> Procesando... -> Fragmento 'L' decodificado como 'L'. M
ensaje: [H][E][L][L][O][ ][B][T][R][L]

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-activid
ad-2-listasdobleenlazadas-DavidIbarra01/build$
```



```
File Edit Selection View Go Run Terminal Help <- > ds-unidad-2-actividad-2-listasdoblementazadas-Davidbarra01
EXPLORER: DS-UNIDAD-2-A...
build
html
latex
Pantallas
.gitignore
CMakeLists.txt
decodificador_prt7.cpp
Doxfile
README.md

decodificador_prt7.cpp : main()
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     ^devod@devod-Aspire-A315-24P:~/Documentos/Dr. Artes Oscuras/ds-unidad-2-actividad-2-listasdoblementazadas-Davidbarra01/build$ ./decodificador_prt7
538
539 } std::cout << "=====";
540 #endif
541 LIn DECODIFICADOR PRT-7 v1.0
542 std::cout << "SISTEMA DE CIBERSEGURIDAD INDUSTRIAL"
543 =====
544 // Bucle
545 char buff
546 int trama
547 while (true) {
548     #if fd < 0
549         ERROR: No se pudo conectar a ningun puerto serial.
550     #else
551         Trama recibida: [L,H] -> Procesando... -> Fragmento 'H' decodificado como 'H'. Mensaje: [H]
552     #endif
553
554     if (h
555         Mensaje de control recibido: [SISTEMA PRT-7 ACTIVO]
556
557     Trama recibida: [L,H] -> Procesando... -> Fragmento 'H' decodificado como 'H'. Mensaje: [H][H]
558
559     Trama recibida: [L,E] -> Procesando... -> Fragmento 'E' decodificado como 'E'. Mensaje: [H][H][E]
560
561
562
563
564
565
566     continue; // Saltar al siguiente ciclo de lectura
567 }
```

Captura de pantalla Ahora mismo

```

File Edit Selection View Go Run Terminal Help <- > Captura de pantalla realizada
EXPLORER: DS-UNIDAD-2-A...
> build
> html
> latex
> Pantallas
  .gitignore
  CMakeLists.txt
  decodificador_prt7.cpp
> Doxyfile
> README.md

cdecodificador_prt7.cpp : main()
528     fd = abrirPuertoSerial(puertos[i]);
529     if (fd != -1) {
530         std::cout << "Conexion establecida en " << puertos[i] << std::endl;
531         break;
532     }
533     std::cerr << "ERROR: No se pudo conectar a ningun puerto serial." << std::endl;
534
535     if (f
      ^ devivo@devivo-Aspire-A315-24P:~/Documentos/Dr. Artes Oscuras/ds-unidad-2-actividad-2-listasdobles
      $ mensaje: [H][H]
      r
      ]
    #endif Trama recibida: [L,E] -> Procesando... -> Fragmento 'E' decodificado como 'E'. M
      ensaje: [H][H][E]
      std::cout Trama recibida: [L,L] -> Procesando... -> Fragmento 'L' decodificado como 'L'. M
      // Bucle
      ensaje: [H][H][E][L]
      char buf
      int trama
      while (trama
      #if fd
      b
      #else Trama recibida: [L,O] -> Procesando... -> Fragmento 'O' decodificado como 'O'. M
      ensaje: [H][H][E][L][L][O]
      #endif
      Trama recibida: [L, ] -> Procesando... -> Fragmento ' ' decodificado como ' '. M
      ensaje: [H][H][E][L][L][O][ ]
      Trama recibida: [M,S] -> Procesando... -> ROTANDO ROTOR +5. (Ahora 'A' se mapea
      a 'F')
      Trama recibida: [L,W] -> Procesando... -> Fragmento 'W' decodificado como 'B'. M
      ensaje: [H][H][E][L][L][O][ ][B]
      i
      continue; // Saltar al siguiente ciclo de lectura
  
```

8 main 0 △ 0 6 nov 19:51

Captura de pantalla Ahora mismo

```

File Edit Selection View Go Run Terminal Help <- > Captura de pantalla realizada
EXPLORER: DS-UNIDAD-2-A...
> build
> html
> latex
> Pantallas
  .gitignore
  CMakeLists.txt
  decodificador_prt7.cpp
> Doxyfile
> README.md

cdecodificador_prt7.cpp : main()
528     fd = abrirPuertoSerial(puertos[i]);
529     if (fd != -1) {
530         std::cout << "Conexion establecida en " << puertos[i] << std::endl;
531         break;
532     }
533     std::cerr << "ERROR: No se pudo conectar a ningun puerto serial." << std::endl;
534
535     if (f
      ^ devivo@devivo-Aspire-A315-24P:~/Documentos/Dr. Artes Oscuras/ds-unidad-2-actividad-2-listasdobles
      $ Strama recibida: [L,O] -> Procesando... -> Fragmento 'O' decodificado como 'T'. M
      ensaje: [H][H][E][L][L][O][ ][B][T][R]
      ]
    #endif Trama recibida: [M,-5] -> Procesando... -> ROTANDO ROTOR -5. (Ahora 'A' se mapea
      std::cout a 'A')
      // Bucle Trama recibida: [L,R] -> Procesando... -> Fragmento 'R' decodificado como 'R'. M
      char buf
      int trama
      while (trama
      #if fd
      b
      #else Trama recibida: [L,L] -> Procesando... -> Fragmento 'L' decodificado como 'L'. M
      ensaje: [H][H][E][L][L][O][ ][B][T][R][L]
      #endif
      Trama recibida: [L,D] -> Procesando... -> Fragmento 'D' decodificado como 'O'. M
      ensaje: [H][H][E][L][L][O][ ][B][T][R][L][O]
      if (h
      Trama recibida: [FIN]. Deteniendo.
      ---
      Flujido de datos terminado.
      MENSAJE OCULTO ENSAMBLADO:
      HHELLO BTRLD
      )---
      Liberando memoria... Sistema apagado.
      devivo@devivo-Aspire-A315-24P:~/Documentos/Dr. Artes Oscuras/ds-unidad-2-actividad-2-listasdobles
      ad-2-listasdoblesenlazadas-DavidIbarra01/build$ 
      continue; // Saltar al siguiente ciclo de lectura
  
```

8 main 0 △ 0 6 nov 19:51