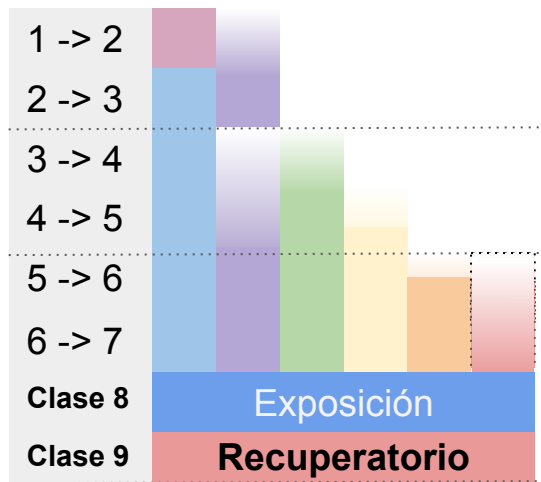


Carrera de Especialización en Sistemas Embebidos

Sistemas Operativos en Tiempo Real II

Clase 4: Ejercitación EAN-13

Cronograma general



Ejercicio introductorio

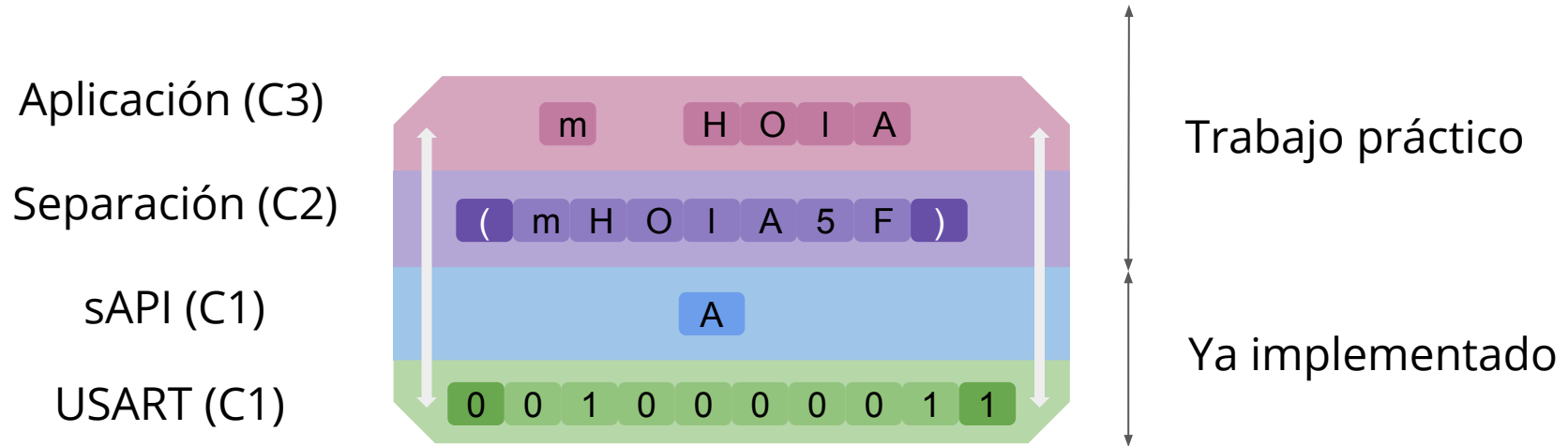
Trabajo práctico parte 1

Trabajo práctico parte 2

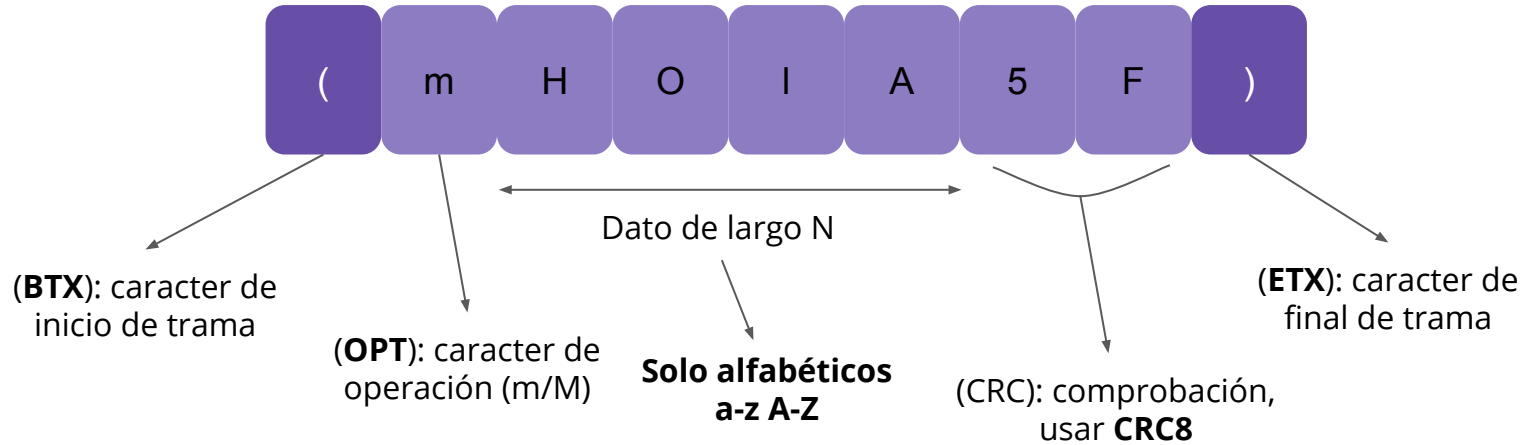
➡ [Link](#)

- malloc()/free()
- modularizar
- Alocación de memoria
- C2 Recepción de paquetes
- C2 Envío de respuesta
- Objeto activo
- Timer HW

Funcionamiento general



Paquete a procesar



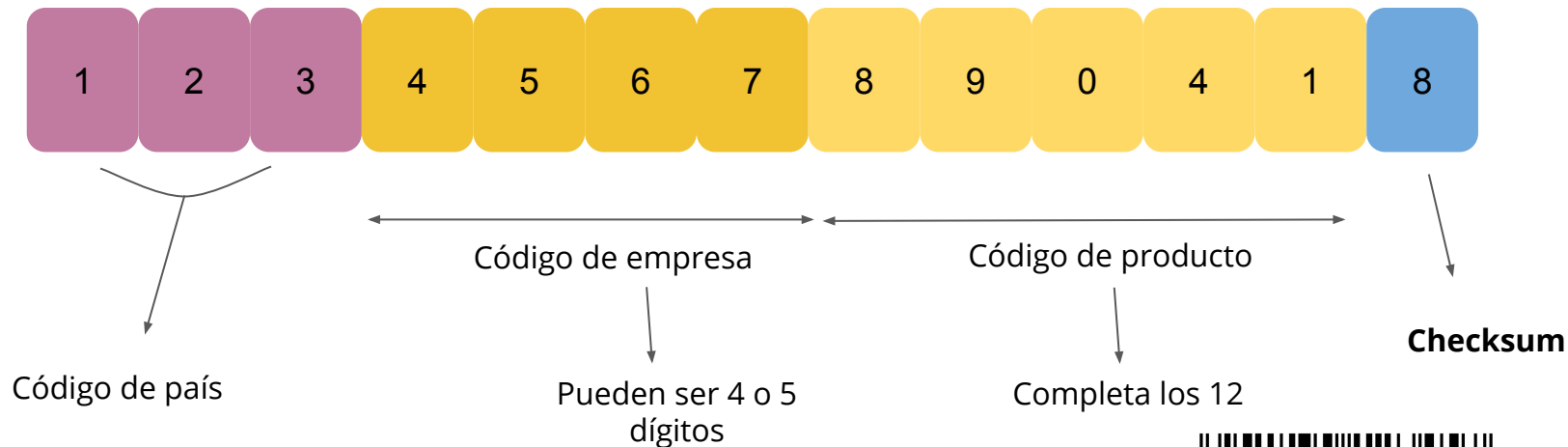
El paquete total tiene un **largo de N+5**.

N es **variable!** → Almacenar con **memoria dinámica**.

Imponer un **máximo** de tamaño.

EAN-13

European Article Number (EAN), consta de 13 dígitos dividido en 4 partes:

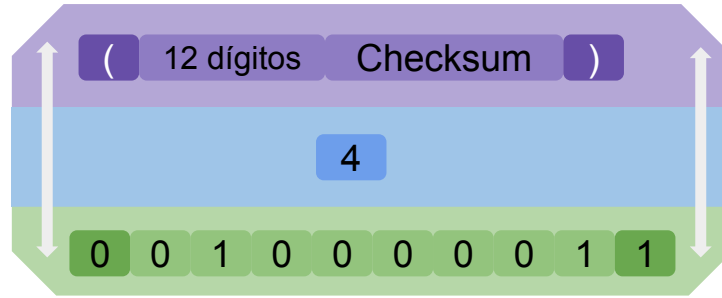


Propuesta de TP

Detección (C2)

sAPI (C1)

USART (C1)

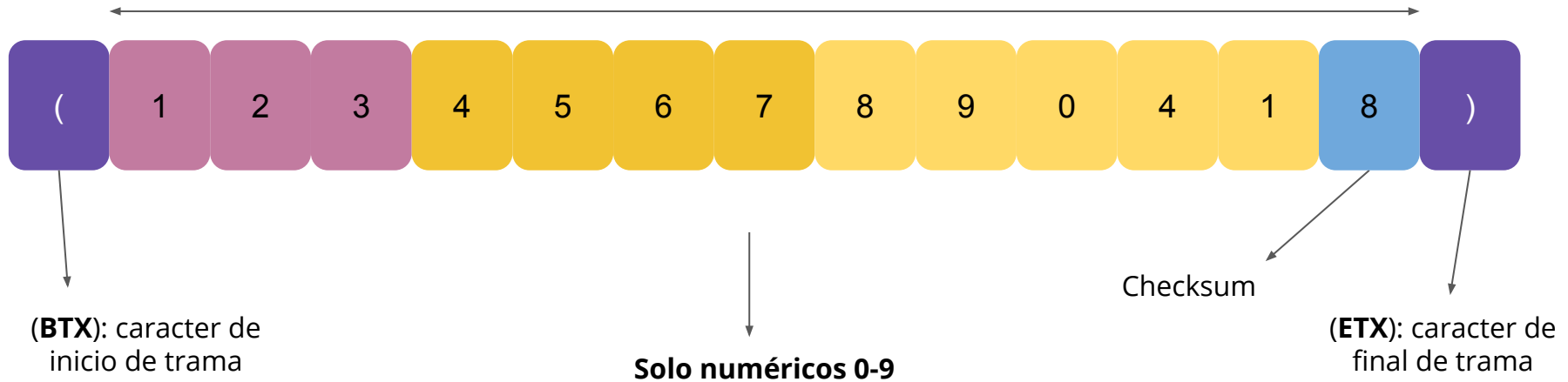


Ejercicio

Ya implementado

Paquete a procesar

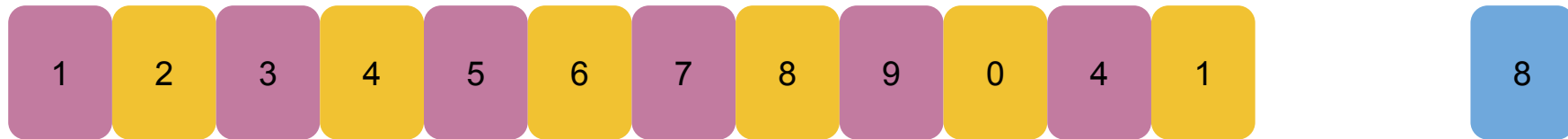
Dato de largo $N = 13$



El paquete total tiene un largo de $N+2$.

N es **Fijo!** → Igualmente usemos memoria dinámica.

Calculo del checksum



$$1 + 3 + 5 + 7 + 9 + 4 = 29$$

$$2 + 4 + 6 + 8 + 0 + 1 = 21 \times 3$$

¿Cuánto le falta para llegar a 100?



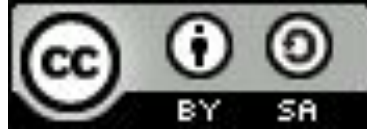
return ((10 - (checksum % 10)) % 10);

Crterios

- BTX erróneo? Descartar trama, imprimir ERROR_B.
- ETX erróneo? Descartar trama, imprimir ERROR_E.
- No todos son digitos? Descartar trama, imprimir ERROR_N.
- Checksum erróneo? No imprimir nada

Se confeccionó un script de **python** para realizar **ensayos** más extensos

Licencia



"Ejercitación EAN-13"

Por Mg. Ing. Martín Menéndez, se distribuye bajo una [licencia de Creative Commons Reconocimiento-CompartirIgual 4.0 Internacional](https://creativecommons.org/licenses/by-sa/4.0/)