

1.º parcialito (1R) – 05/03/2021

Resolvé los siguientes problemas en forma clara y legible. Podés incluir tantas funciones auxiliares como creas necesarias.

1. Escribir en C una **función** que dada una cola y una función cuya firma es `void* transformar(void*)`, devuelva una **nueva** cola cuyos elementos sean los resultantes de aplicarle la función `transformar` a cada elemento de la cola original. Los elementos en la nueva cola deben tener el orden que tenían en la cola original, y la cola original debe quedar en el mismo estado al inicial (si bien puede ser modificada durante la ejecución de la función). Indicar y justificar la complejidad de la función implementada. La firma de la función debe ser `cola_t* transformar_cola(cola_t* original, void* (*transformar)(void*))`.

Por ejemplo, para la cola [ 1, 2, 3, 6, 2 ] (primero es el número 1), y la función `sumar_uno` (que devuelve la suma entre el número 1 y el número recibido), la cola resultante debe ser [ 2, 3, 4, 7, 3 ] (el primero es el número 2).

2. Implementar una **primitiva** `void cola_invertir(cola_t* cola)` para una *Cola Dinámica/Circular*, que invierta el orden de sus elementos, sin utilizar estructuras auxiliares (i.e. en espacio constante). Indicar y justificar la complejidad de la primitiva. La estructura interna de la cola es la siguiente:

```
typedef struct cola {  
    void** datos;  
    size_t capacidad, cantidad;  
    size_t inicio, fin;  
} cola_t;
```

3. Implementar una función `float suma_total(float arr[], size_t n)` que, **por división y conquista**, devuelva la suma de todos los elementos. Indicar y justificar adecuadamente la complejidad de la función implementada.