

Diseño

Algoritmos y estructuras de datos

Elías Cerdeira



¡Bienvenidos
una vez más!

¿Qué vamos a ver hoy?

- Breve introducción a diseño
- Diseñar una cola
- Diseñar un buffer circular



¿Qué implica diseñar?

- **Elegir estructuras** concretas para almacenar la información
- Escribir los **algoritmos de las operaciones** especificadas
- Contemplar el **contexto de uso**



¿El diseño es único?

- **No.** Según el contexto de uso puede variar la estructura que elija o cómo se implementan los algoritmos
- Todos los diseños deben **cumplir con la especificación** del TAD



¿Cómo luce un diseño?

```
Modulo <Impl> implementa <TAD> {  
    var campo1: Type1  
    var campo2: Type2  
  
    pred InvRep(i: <Impl>)  
        { ... }  
  
    pred Abs(i: <Impl>, t: <TAD>)  
        { ... }  
  
    proc Op1(inout i: <Impl>, in v: V)  
        { ... pseudocodigo ... }  
    proc Op1(in i: <Impl>): V  
        { ... pseudocodigo ... }  
}
```

Representan el **estado** del módulo. Se manipulan con los algoritmos. Se utilizan tipos de **implementación**.

Restricciones sobre los conjuntos de valores **válidos** para las variables de estado.

Establece relación entre una **instancia** del **TAD** y una del **módulo**. Asocia variables con observadores.

Pseudocódigo que implementa la operación. Se utilizan operaciones de los TADs.



Invariante de representación

```
pred InvRep(i: <Impl>)  
  { ... }
```

- Especifica qué **conjunto de valores** para las variables de la estructura representan una **instancia válida**
- Se debe cumplir siempre al entrar y salir de las operaciones
- Se especifica en **lógica**



“Función” de abstracción

```
pred Abs(i: <Impl>, t: <TAD>)  
  { ... }
```

- Indica si una instancia de la implementación (Diseño) corresponde con una instancia de la especificación (TAD)
- Es un predicado por lo que está especificado en lógica
 - Se asume que se cumple el invariante de representación



¡Vamos a ver el apunte!

Yeaaah!



¡A resolver ejercicios!



Ahora volvemos
a lo divertido...

Diseñar cola

Se nos pide diseñar una **Cola** utilizando como estructura una **Lista Enlazada**

- Dar el invariante de representación y la función de abstracción
 - ¿Qué propiedades de la LE hay que tener en cuenta?
- Escribir los **algoritmos de las operaciones** especificadas
- ¿Qué limitaciones en complejidad tiene este diseño?

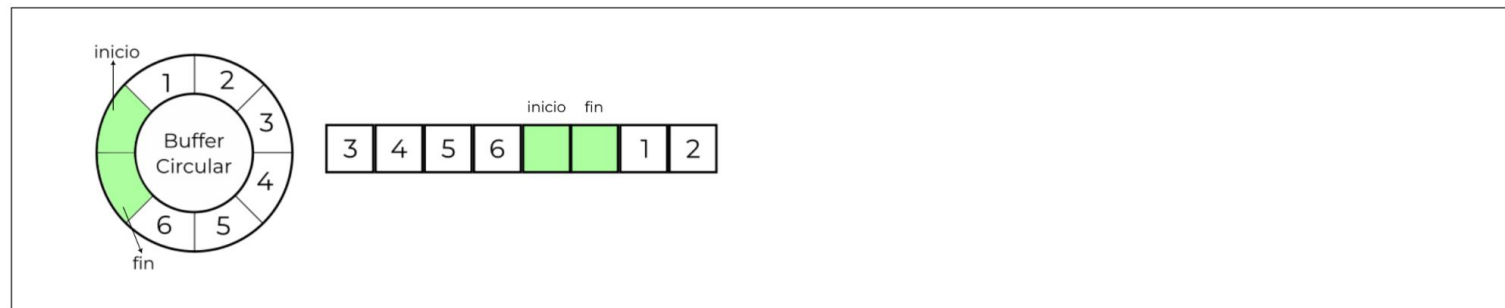
¡Intervalo!

¡Me hago pipí!



Diseñar conjunto acotado

Ejercicio 5. Una forma eficiente de implementar el TAD Cola en su versión acotada (con una cantidad máxima de elementos predefinida), es mediante un *buffer circular*. Esta estructura está formada por un array del tamaño máximo de la cola (n) y dos índices (*inicio* y *fin*), para indicar adonde empieza y adonde termina la cola. El chiste de esta estructura es que, al llegar al final del arreglo, si los elementos del principio ya fueron consumidos, se puede reusar dichas posiciones.



- Elija una estructura de representación
- Escriba el invariante de representación y la función de abstracción
- Escriba los algoritmos de las operaciones **encolar** y **desencolar**
- ¿Por qué tiene sentido utilizar un buffer circular para una cola y no para una pila?

¿Qué sigue?

- Con la clase de hoy pueden resolver **toda** la guía 6
- La clase que viene veremos diseños más complejos



¡Terminamos!

¡Hagan consultas!

Gracias por
acompañarnos

