Estructuras de Datos Dinámicas: Tipo de Dato Abstracto

parte I

Programación I
Departamento de Informática
Universidad Nacional de San Luis
Argentina

Progreso de la Abstracción

- Los diferentes niveles de abstracción que posee un lenguaje, dependen de los mecanismos proporcionados por el lenguaje elegido:

  - Ensamblador
  - Procedimientos
  - Módulos

  Perspectiva Funcional

  - Paquetes
  - Tipos de datos abstractos (TDA)

  Perspectiva de Datos

  - Objetos
    - TDA
    - + paso de mensajes
    - + herencia
    - + polimorfismo

  Perspectiva de Servicios

Definición de la abstracción

- **¿ Que es la Abstracción?**
- Supresión intencionada (u ocultación) de algunos detalles de un proceso o artefacto, con el fin de destacar más claramente otros aspectos, detalles o estructuras.
- En cada nivel de detalle cierta información se muestra y cierta información se omite.
  - Ejemplo: Diferentes escalas en mapas.
- Mediante la abstracción creamos **MODELOS** de la realidad.

Lenguajes de programación y niveles de abstracción

- Los lenguajes de programación proporcionan abstracciones.

- **Lenguajes Ensamblador**
- **Lenguajes Imperativos**
  - (C, Fortran, BASIC)
- **Lenguajes Específicos**
  - (LISP, PROLOG)

Simular

Abstracción

Directa, Dificultosa

- **Lenguajes Orientados a Objetos puros**
  - (Smalltalk,Eiffel)

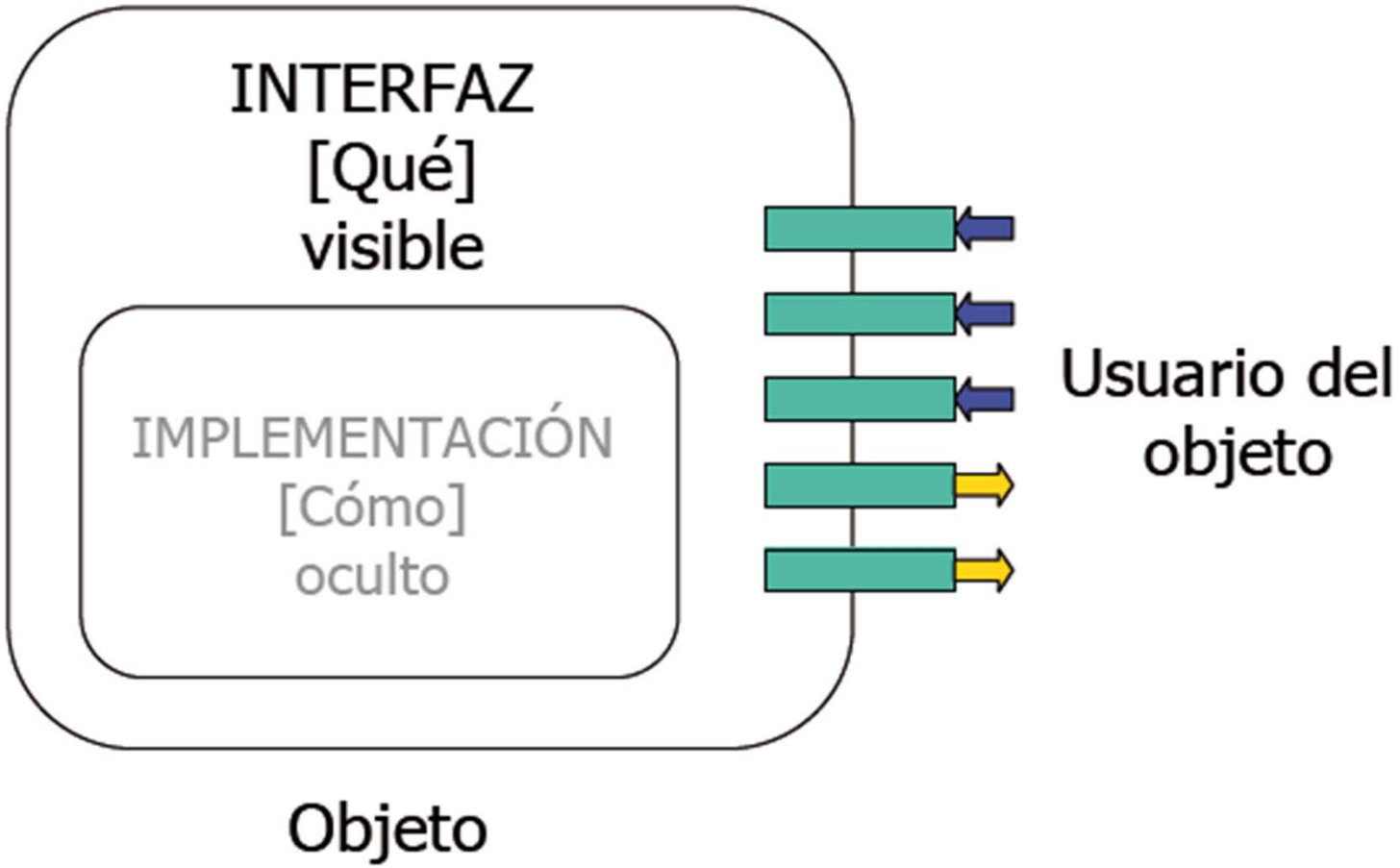
Abstracción

Directa

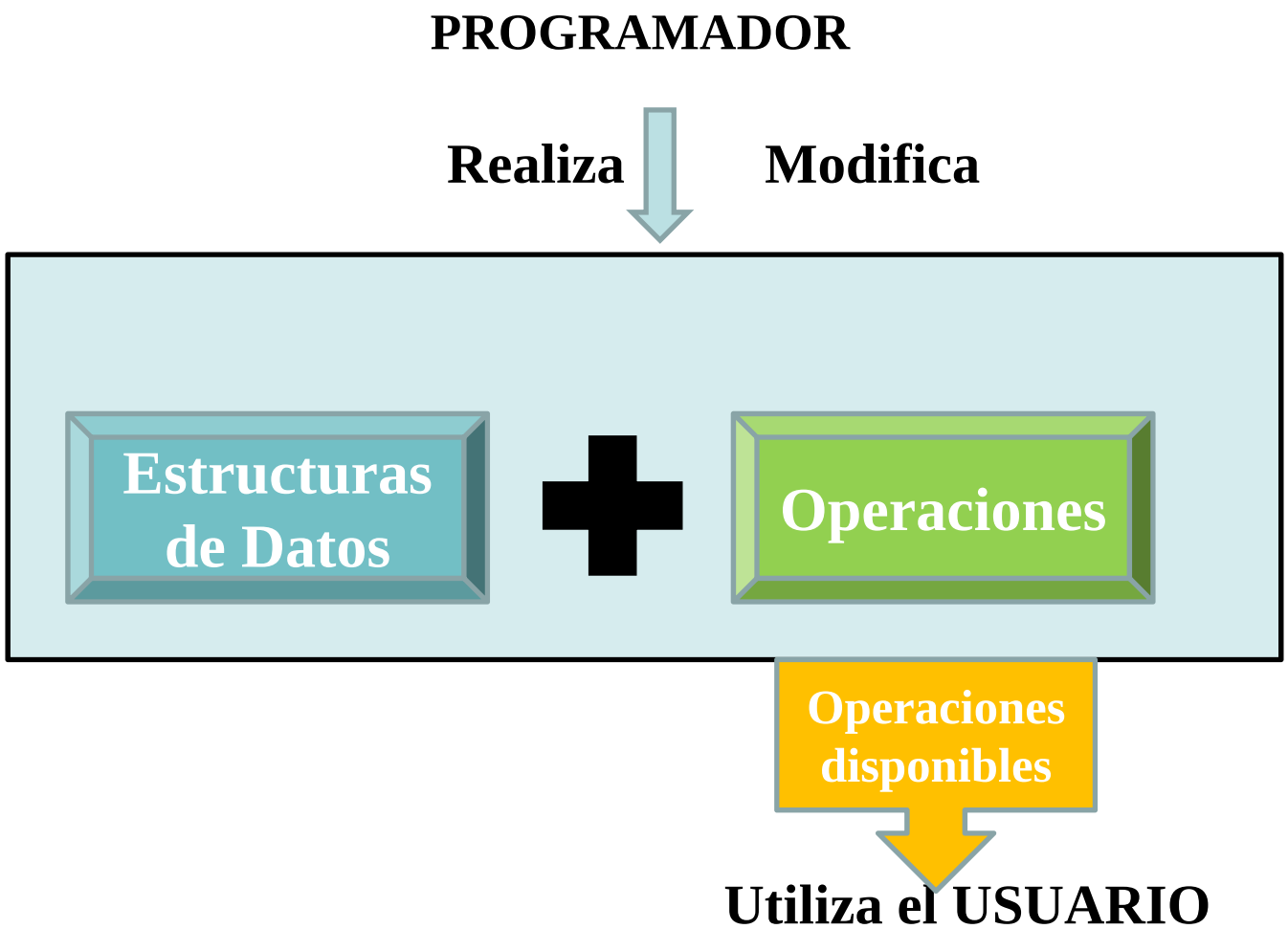
- **LOO Híbridos (Multiparadigma)**
  - C++, Object Pascal, Java,…

Abstracción

Mecanismos de Abstracción en los lenguajes de programación



¿Qué es un TDA ?


PROGRAMADOR
Realiza
Modifica
Estructuras de Datos
Operaciones
Operaciones disponibles
Utiliza el USUARIO

Tipos de Datos Abstractos (TDA)

- **Tipos básicos:**
  - Ej: entero, real, carácter, etc.
- **Tipos definidos por el usuario:**
  - Ej: Estado civil (soltero, casado, viudo, separado)
- Un tipo es el conjunto de valores que puede tomar un dato (Pascal).
- Un tipo es un conjunto de valores y un conjunto de operaciones.
- **Tipo de dato abstracto:**
  - Ej: pila, fila, lista, irracional, persona
- Un tipo de dato abstracto es una entidad que encapsula valores y operaciones. Los valores son sólo accesibles a través de las operaciones definidas por el usuario. El usuario desconoce la representación o implementación del tipo .

Ejemplo: TDA auto

- 1° definir la/s estructura/s de datos necesarias para almacenar los datos que necesito

- ¿Qué datos representan un auto?

- ¿Cuáles me sirven según mi problema a resolver?

- ¿Qué tipo de dato es el mas apropiado para cada uno de ellos?

Pensemos!!

Ejemplo: TDA auto

- 2° definir las operaciones que necesito para trabajar con la estructura de datos definida en el paso anterior
- Operaciones básicas:
  - Inicializar auto: colocar valores por defecto (una función si fuera necesario inicializar valores)
  - Cargar y Modificar auto: cambiar los valores del auto de a uno por vez. (Varias funciones modificar)
  - Mostrar auto: obtener los datos del auto de a uno. (varias funciones para obtener los datos)

Ejemplo: TDA auto

- Operaciones adicionales/solicitadas
  - Definir funciones necesarias para calcular/operar con la estructura según el problema a resolver
  - El programador es el «dueño» de la estructura, el usuario no conoce la implementación del TDA. El programador puede modificar la implementación del TDA.

Consideraciones TDA en C

¿Dónde defino el TDA?

En una librería «auto.h»

¿Cómo utilizo el TDA ?

En un archivo «usoauto.c» donde incluyo la librería auto.h (#include "auto.h") y utilizo solo las operaciones definidas.

NO UTILIZO DIRECTAMENTE LA ESTRUCTURA desde el programa **usoauto.c**

En resumen:
¿Qué debe contener un TDA?

- Una estructura de datos que permita almacenar la información (valores del cada elemento).
- Operaciones que permitan:
  - Inicializar el TDA (construye un elemento del TDA)
  - Establecer valores al TDA (valores a cada parte)
  - Modificar el TDA (solo los valores posibles)
  - Mostrar el TDA (cada uno de los valores)

IMPLEMENTACIÓN DE ESTRUCTURAS DE DATOS CON TDA

Repaso de Estructuras de Datos (1)

- Nociones Generales
  - Agrupación de valores que por razones lógicas, se quiere conservar ‘juntos’.
  - Cómo **se incorpora** un nuevo elemento a la estructura.
  - Cómo **se elimina o cambia** un elemento que ya está en la estructura.
  - Cómo **se inspecciona** un elemento que ya está en la estructura.

Repaso de Estructuras de Datos (2)

- Estos tres últimos aspectos tienen que ver con las **operaciones** que pueden hacerse sobre la estructura de datos: **Poner**, **Sacar**, **Inspeccionar**.
- Orden, con respecto a las operaciones: **Cronológico**, **No Cronológico**.
  - Cómo **se guardan** los elementos en la estructura, es decir en qué **orden** se encuentran unos con respecto a los otros. Deben tener algún orden o estructura.

Repaso de Estructuras de Datos (3)

- Cuántos elementos se pueden guardar, **capacidad de almacenamiento** de la estructura: **Estática / Dinámica**.
- Identificar o seleccionar los elementos de la estructura: *sin ambigüedad*. **Selector** de la estructura, **Unívoco**. **Explícito / Implícito**.
- Tipo de los datos de los elementos de la estructura. A este tipo de dato le llamaremos tipo de dato **base** de la estructura. **Simples / Compuestos**. **Homogéneo / Heterogéneo**.

Estructuras de Datos

PILAS Y FILAS

Estructuras de Datos Dinámicas: Pila (stack) (1)

- Capacidad:

  Dinámica, crece y disminuye con las inserciones y supresiones

  Por eso se habla de que la estructura puede estar vacía

- Orden: **LIFO** **L**ast **I**n **F**irst **O**ut.

Estructuras de Datos Dinámicas: Pila (stack) (2)

- Operaciones.
  - Inicialización (*init*)
  - Inserción (*insert* o *push*)
  - Supresión (*suppress* o *pop*) } tope
  - Copia (*copy* o *top* o *peek*)
  - Predicados: *isEmpty*, *isFull*.

- Selector: implícito ⇒ tope

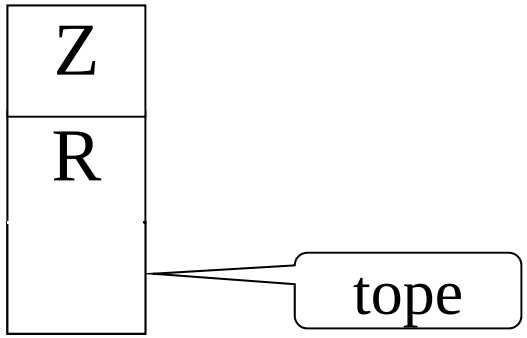
Estructuras de Datos Dinámicas: Pila (stack) (3)

suprimir (pop)

insertar (push) Z

insertar (push) R

insertar (push) Q



isEmpty = false (verdadero)

isEmpty = false (falso)

Estructuras de Datos Dinámicas: Fila o cola (queue) (1)

- Capacidad:

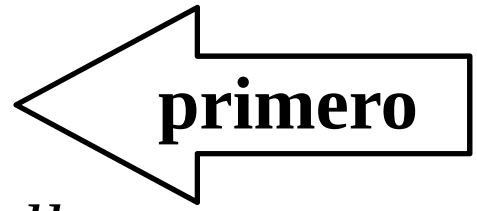
  Dinámica, crece y disminuye con las inserciones y supresiones

  Por eso se habla de que la estructura puede estar vacía.

- Orden: **FIFO** **F**irst **I**n **F**irst **O**ut.

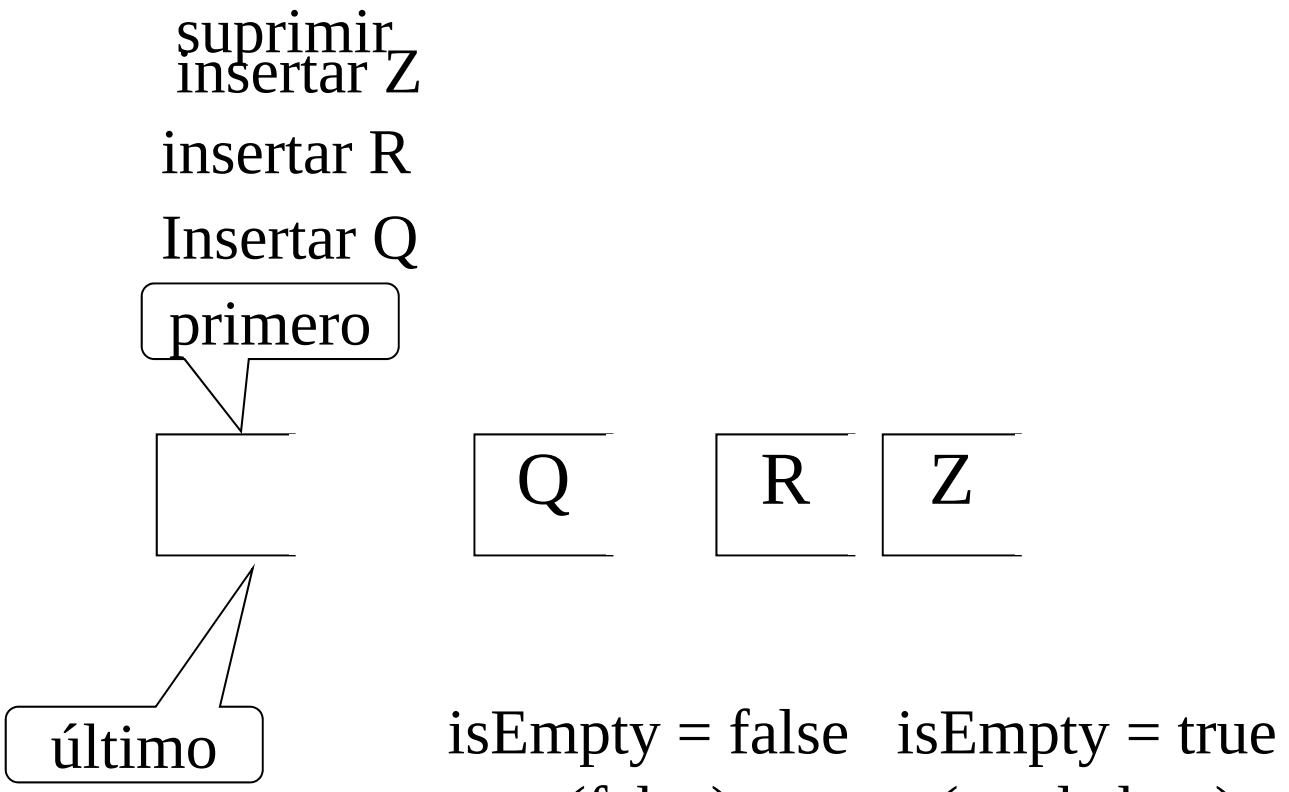
Estructuras de Datos Dinámicas: Fila o cola (queue) (2)

- Operaciones.
  - Inicialización (*init*)
  - Inserción (*insert*) ← **después** del último
  - Supresión (*suppress*)
  - Copia (*copy*)
  - Predicados: *isEmpty*, *isFull*.



- Selector: implícito ⇒ primero / último

Estructuras de Datos Dinámicas: Fila o cola (queue) (2)



Tipo de Datos Abstracto (TDA) (1)

- En inglés: Abstract Data Type (ADT).
- TDA se define a partir de las operaciones que pueden ser realizadas sobre los datos del tipo.
- Hay modelos matemáticos para los tipos de datos. Ej.: enteros y suma.
- Ocultamiento de la información (information hiding).

Tipo de Datos Abstracto (TDA) (2)

Ejemplo: Stack (Pila) podría ser definido por solo tres operaciones (funciones) sobre las pilas: init, push, pop, y un predicado: isEmpty.

Operaciones (funciones) :

init: $\rightarrow$ Stack

push: Elem $\times$ Stack $\rightarrow$ Stack

pop: Stack $\rightarrow$ Stack $\times$ Elem

isEmpty: Stack $\rightarrow$ boolean

Tipo de Datos Abstracto (TDA) (3)

init: $\rightarrow$ Stack

push: Elem $\times$ Stack $\rightarrow$ Stack

pop: Stack $\rightarrow$ Stack $\times$ Elem

isEmpty: Stack $\rightarrow$ boolean

pop(init()) = ERROR

pop(push(e, s)) = (s, e)

isEmpty(init()) = true

isEmpty(push(e, s)) = false

Tipo de Datos Abstracto (TDA) (4)

- Lenguaje de programación: se implementa un nuevo tipo no provisto por el lenguaje, sobre la base de los tipos y operaciones existentes.
- TDA en C para prácticas de pilas, filas, listas cuyo tipo base puede ser enteros (`int`) y carácter (`char`).
- Declaraciones de variables y operaciones.
  - Ej.: `pila_char pila;`
- Apunte TDA.

Tipo de Datos Abstracto (TDA) (5)

Ejemplo de uso (1)

```
/*  ******  Print fila int  ****** */
void printFilaInt (fila_int x) {
    while (!isEmpty(x)) {
        printf("%d ", copy(x));
        suppress(&x);
    }
    printf("\n");
} /* fin de printFilaInt */
```

Tipo de Datos Abstracto (TDA) (6)

Ejemplo de uso (2)

```
/*  ******  Buscar en fila int  ****** */
void buscaFilaInt (fila_int x, int y) {
    while (!isEmpty(x) && copy(x) != y) {
        suppress(&x);
    }
    if (!isEmpty(x))
         printf("%d esta en la fila\n", y);
    else
         printf("%d NO esta en la fila\n", y);
} /* fin de buscaFilaInt */
```

Tipo de Datos Abstracto (TDA) (7)

Ejemplo de uso (3)

```
/*  ******  Copiar fila int  ****** */
void copiaFilaInt (fila_int *x,
                   fila_int y){
     while (!isEmpty(y)) {
          insert(x, copy(y));
          suppress(&y);
     }
} /* fin de copiaFilaInt */
```

Fin … por ahora ;-)