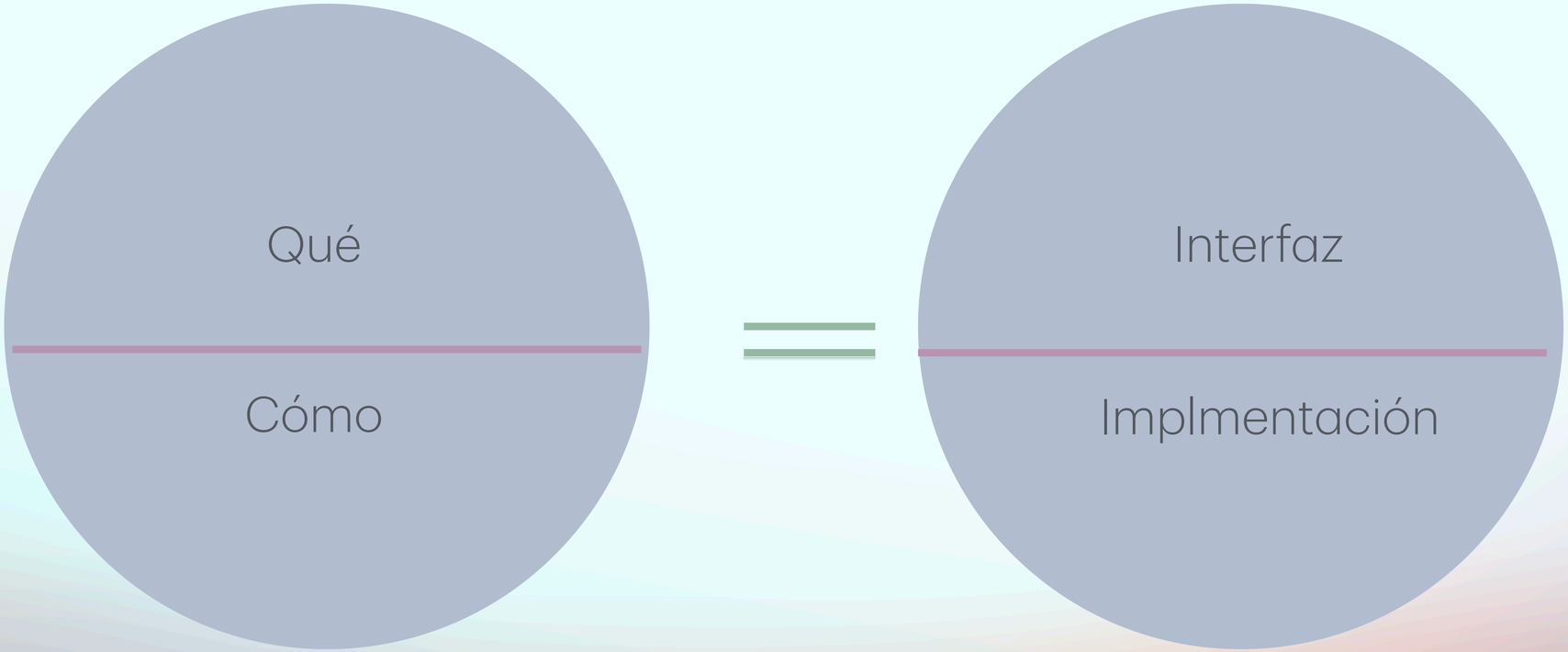
Tipos Abstractos de Datos y Asertos

Ana Lilia Laureano Cruces
Universidad Autónoma Metropolitana-Azcapotzalco

TAD´s

Tipos Abstractos de Datos

- Los tipos abstractos de datos (TAD´s) y los asertos son una técnica que obliga al diseñador a pensar primero en qué es lo que se supone desea haga el programa, antes de lanzarse a la computadora, lo que podría originarte dolores de cabeza muy fuertes.
- Un TAD, es una estructura de datos a la cual se le asocia un conjunto de operaciones .
- Ejemplos :
  - Un **grafo G** = <v,e>; un conjunto de: vertices y estados (arcos y ligas).
  - Una **matriz**; un conjunto de enteros, con operciaones (inversa, transpuesta, multiplicación, suma).
  - Una **lista** (secuencia); una colección ordenada de cero o más enteros; con significado en la posición.


Qué
Cómo
=
Interfaz
Implmentación

Especificación

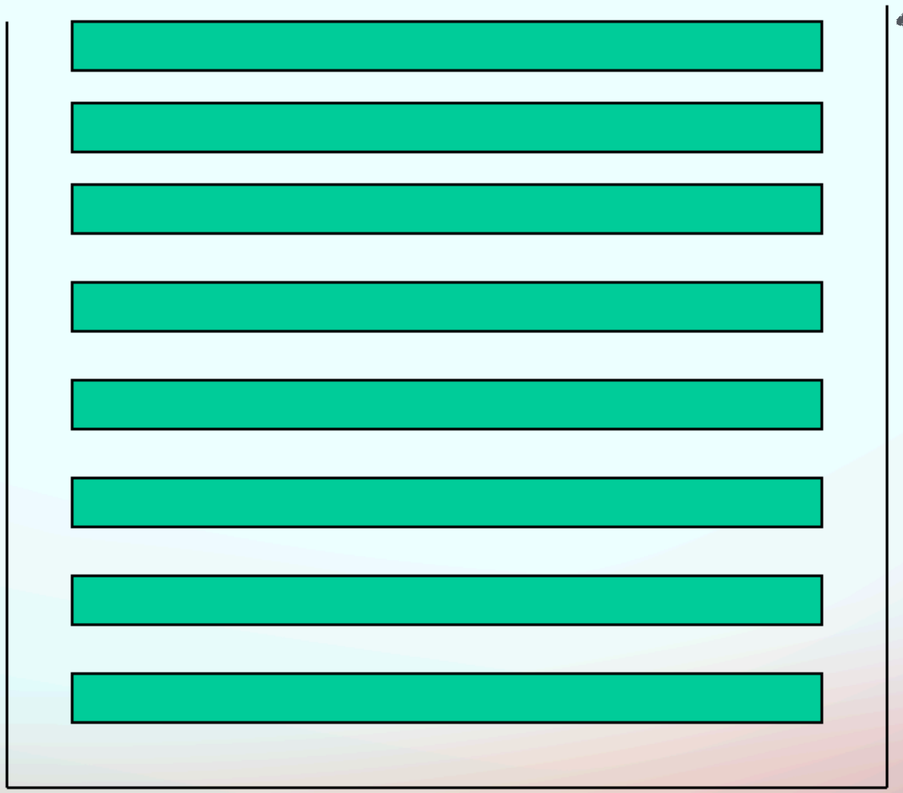
Ejemplo: qué caracteriza a una Plia

- Esta relacionada con la descripción de que es lo que se va a producir.
- Son llevadas a cabo en notación matemática.
- Nos ayuda a comparar la implentación final (Validación).

Una pila es una colección de datos secuenciales, donde cada localidad aloja un dato del mismo tipo. Los datos son agregados y retirados por un lado específico, llamado tope y solo es posible accesar los datos a través de ese tope.

El siguiente elemento a ser insertado va en el tope de la pila.

Elemento que se encuentra en el tope de la pila



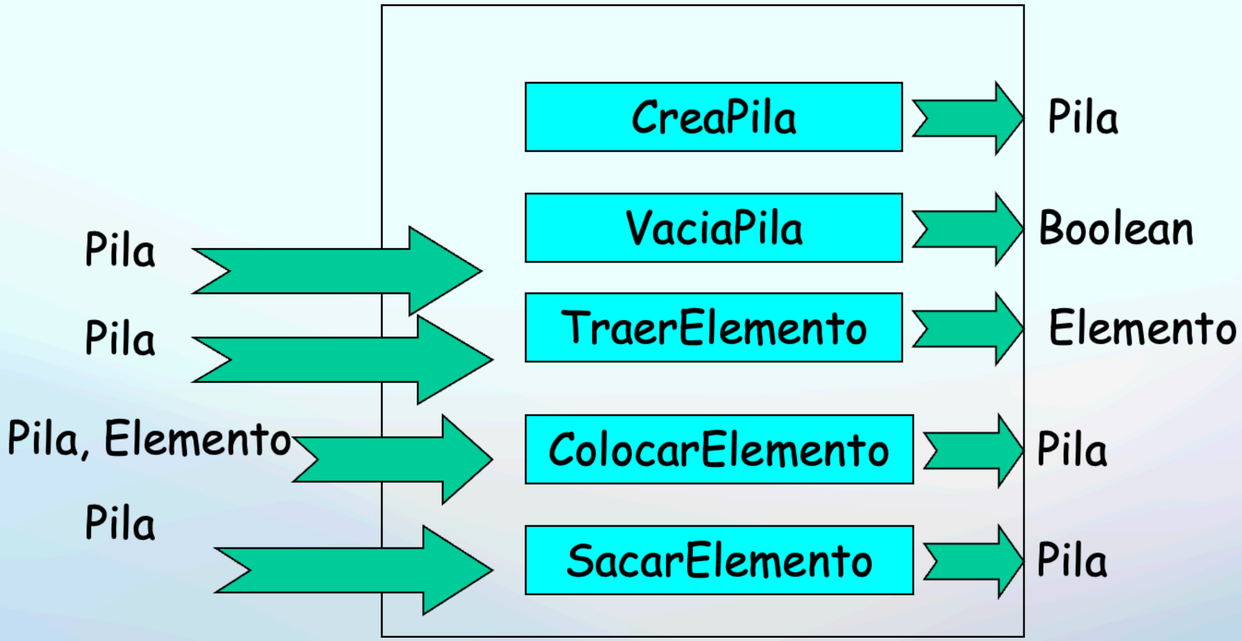
TAD Pila

Comportamiento de las Operaciones

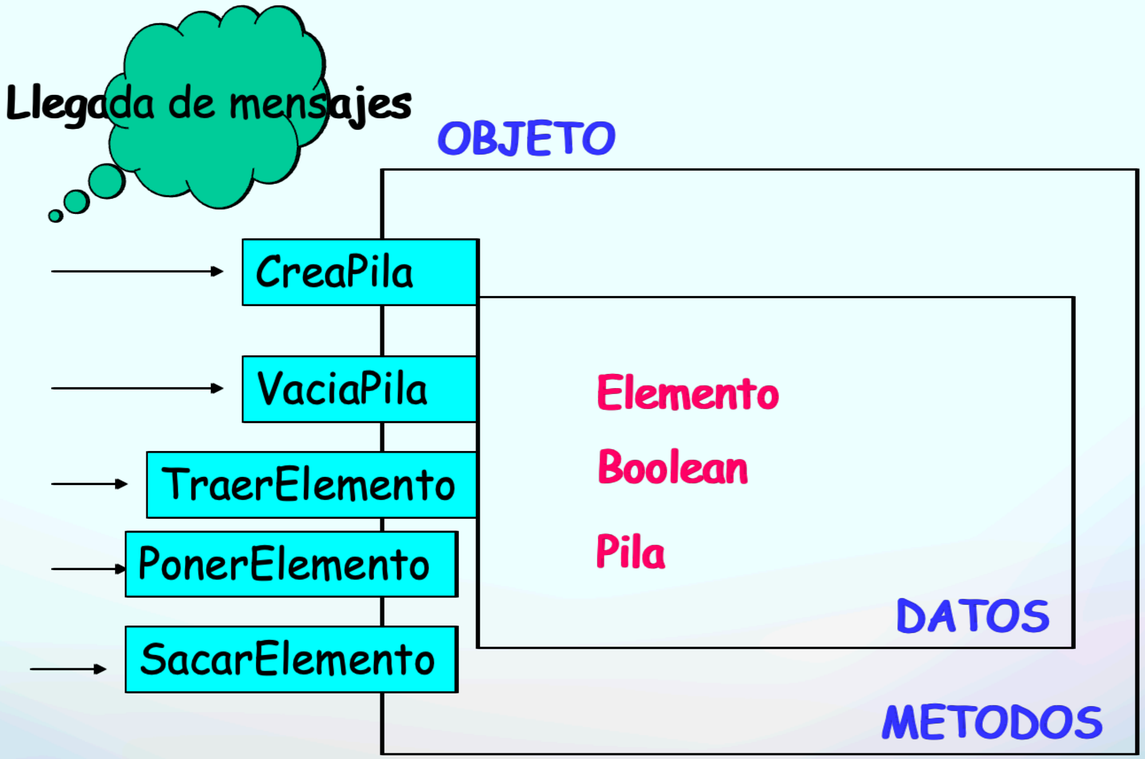
- TraerElemento: regresa el valor del tope de la Pila.
- ColocarElemento: introduce un elemento y regresa una pila con un elemento mas.
- Sacar Elemento: regresa una pila con un elemento menos (el del tope).
- CreaPila: regresa una pila nueva y vacía.
- VaciaPila: regresa un valor verdadero cuando la pila esta vacia y falso en caso contrario.
- AnulPlia: libera una estructura pila.

Una caja Negra llamada Pila

- Se deben incluir a la descripción anterior el tipo de datos que se recibe y el tipo de datos que se produce (dominio y codominio).

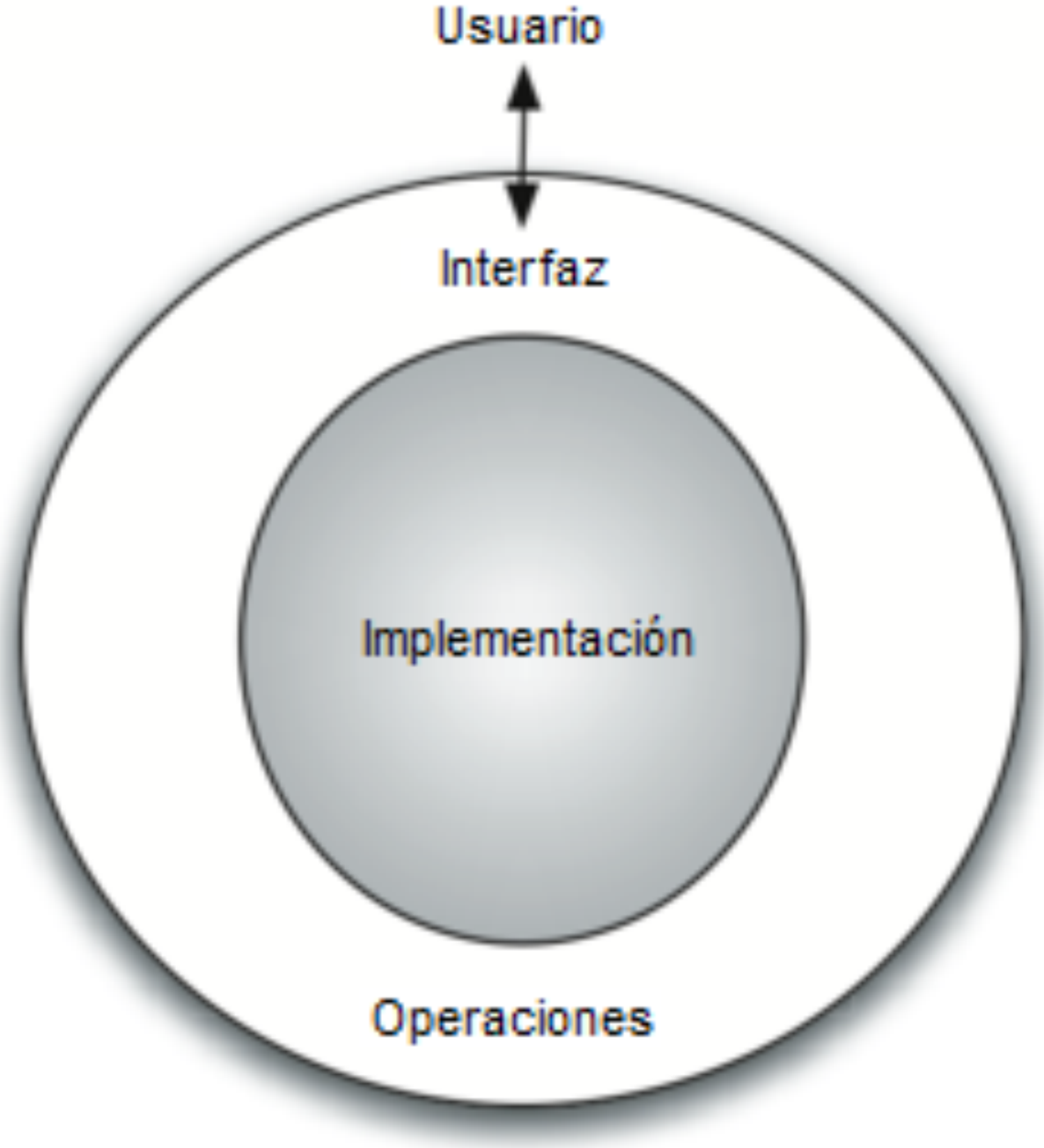


Anatomia del objeto Pila



Definición de un TAD

- Un TAD, es un conjunto de operaciones; la ejecución de un TAD, sólo puede ser alcanzada a través de esas operciones y los resultados de estas.



Verificación Formal

TAD - Ejemplos

- Esta representada por técnicas para preevenir errores en la etapa de diseño y de codificación, sin tener que ejecutar el programa.
- El objetivo es probar programas de manera análoga a las demostraciones en otras areas de matemáticas.
  - **Lenguaje Z**; Se basa en la teoría de conjuntos y la lógica de predicados.
  - **VDM**; en base a conjuntos de tipos de datos.
  - **Larch**; combina especificaciones axiomáticas y algebraicas.

Los asertos

Una técnica basada en afirmaciones que se hacen con

respecto al estado de un programa, en un punto (crítico)

y que son representadas por un **aserto**.

Suma Parte + 1

aserto { Suma > Parte }

Beneficios:

- Demostrar mediante argumentos lógicos que un diseño o implementación satisface los asertos.

- En lugar de ejecutar el programa.

Asertos

Características

- PreCondición: deben ser verdad antes de ejecutar un módulo lógico.
- PostCondición: que resultados son esperados, después de ejecutado un módulo lógico.
- Invariante de Ciclo: el estado de las variables antes de la I-ésima iteración, durante la ejecución y después de la ejecución.

TIP’s para descubrir componentes reusable y/o TAD’s

- Alguno de los componentes del sistema puede ser útil para otros (un componente reusable).
- El componente reusable puede ser usado con diferentes tipos de datos básicos y operaciones relacionadas con ellos (un TAD)

Cómo construir un componente reusable

- Especificarlo en dos partes: la interfaz o lo que ofrece **(Qué)** conocido como definición y la implentación **(Cómo)**.
- La instancia de un componente reusble es realizada al instanciar la parte de implementación, quien a su vez instancia a la parte de definición.

Etapas de diseño de los TAD's

1. Especificación.
2. Funcionalidad.
3. Verificación.
4. Implementación.

Especificación

- Realizar una descripción en función de las operciones que necesita para:

1. Crear objetos (inicializadoras y constructoras).
2. Transformar objetos (simplificadoras).
3. Analizar el estado de un objeto (analizadoras).
4. Destruir objetos (inicializadoras y destructoras).

Funcionalidad

En esta etapa descubriremos el dominio y el codominio de los datos utilizados por las operaciones que componen al TAD.

Verificación

A esta etapa le corresponde la formalización del comportamiento de las operaciones, se basan en la **pre y post condicion** y en la **invariante** del ciclo. Además se utilizan las estructuras de control permitidas para la *programación estructurada*.