DEPARTAMENTO DE
INGENIERÍA
INFORMÁTICA
UNIVERSIDAD DE SANTIAGO DE CHILE

Análisis de Algoritmos y Estructura de Datos

Implementación de TDA pila

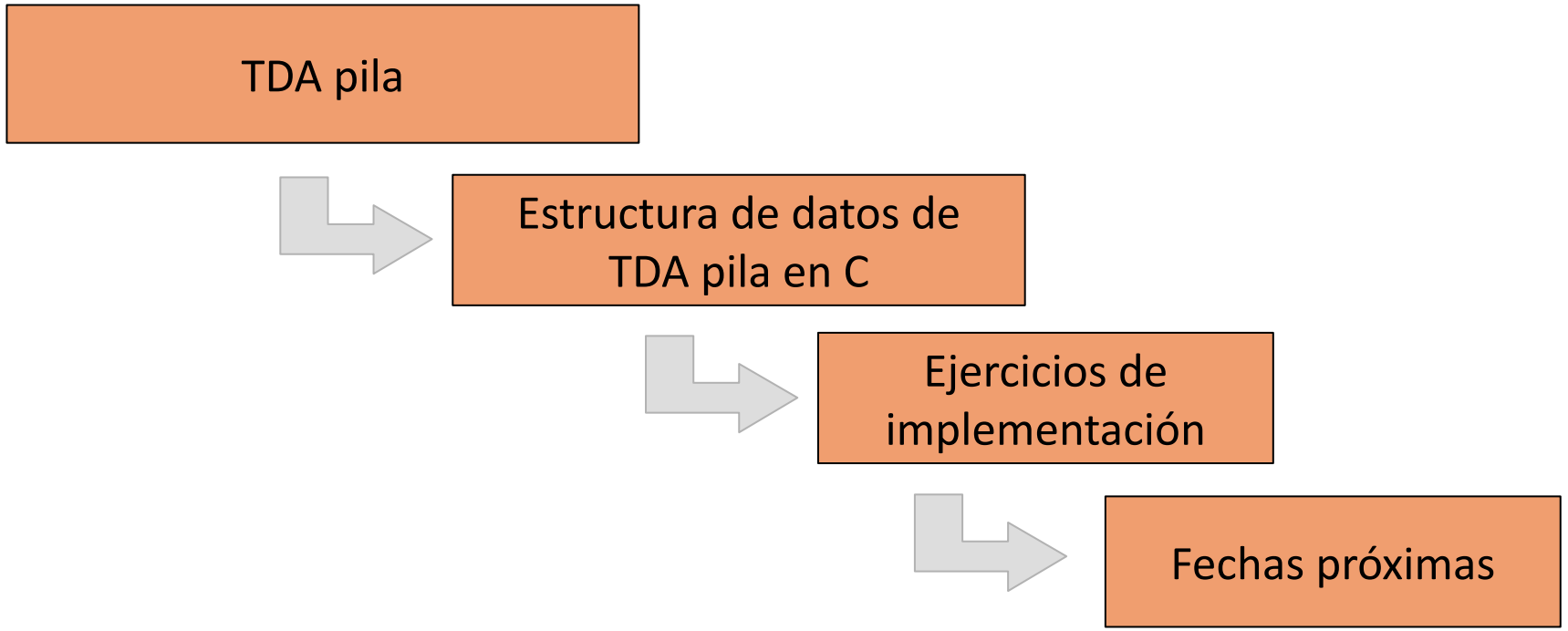
Prof. Violeta Chang C

Semestre 2 – 2023

TDA pila

- **Contenidos**:
  - Estructura de datos de TDA pila
  - Operaciones de TDA pila
- **Objetivos**:
  - Implementar TDA pila

Ruta de trabajo

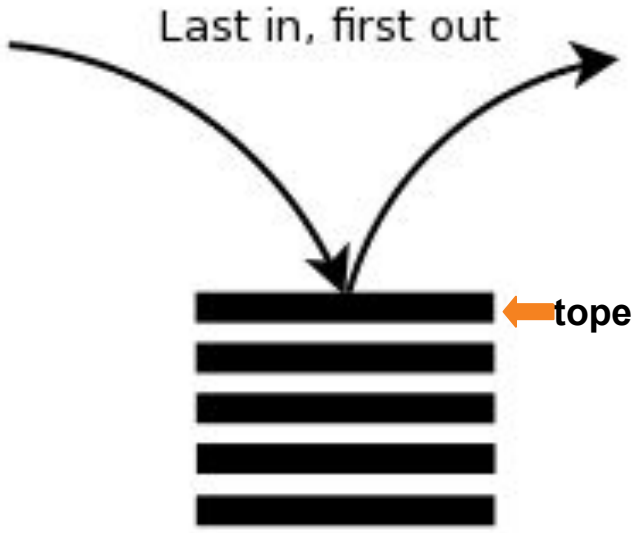


DEPARTAMENTO DE
INGENIERÍA
INFORMÁTICA
UNIVERSIDAD DE SANTIAGO DE CHILE

Especificación e implementación de TDA pila

Especificación de TDA pila

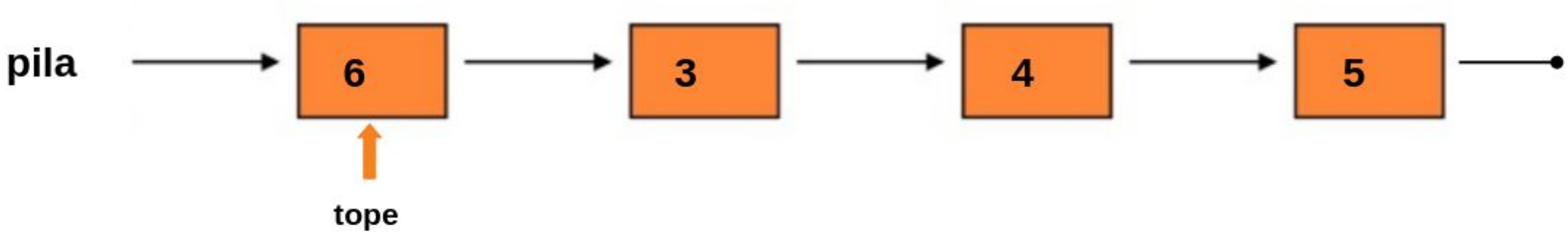
- **Estructura de datos**:
  - Una pila es una secuencia de elementos que cumple la propiedad **LIFO** (last in, first out), es decir, el último elemento que se introduce en la pila, será el primer elemento en salir de ella
  - **Tope de la pila**: extremo de la lista por donde se introducen y extraen los elementos



Representación de TDA pila

- **Dos opciones:**
  - Usando un arreglo:

  - **Usando una lista enlazada simple:**

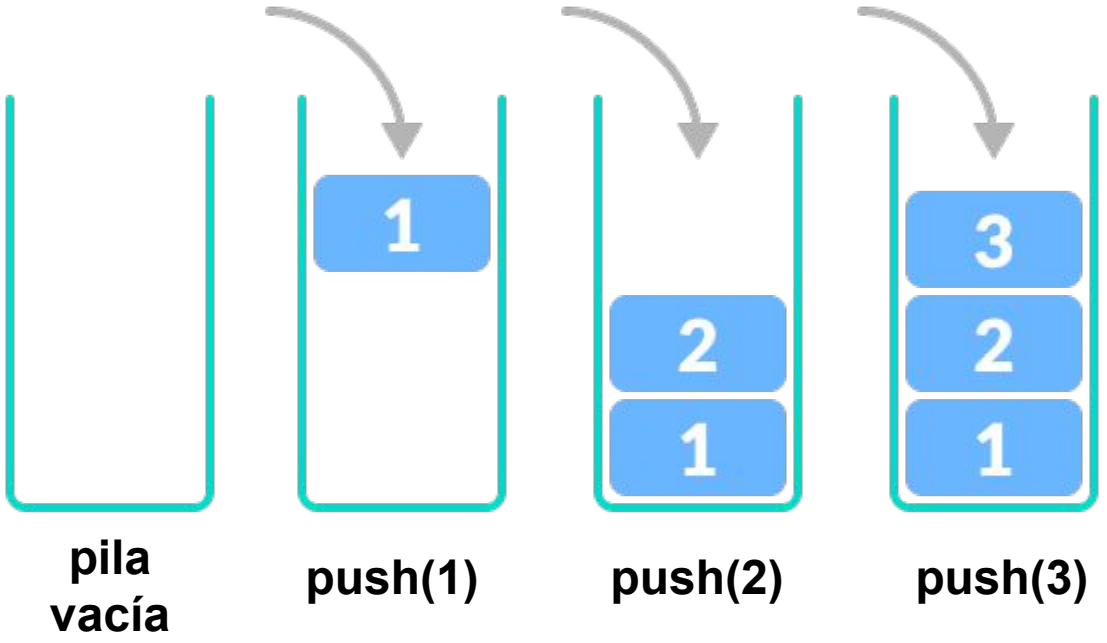


Especificación de TDA pila

- **Operaciones:**
  - **apilar(pila,dato)**: agrega un elemento al comienzo de la pila (tope) → **push**
  - **desapilar(pila)**: quita el primer elemento en el tope de la pila, sin devolverlo → **pop**
  - **tope(pila)**: devuelve el primer elemento de la pila, sin extraerlo → **cima**, **top**
  - **esPilaVacía(pila)**: comprueba si la pila está vacía

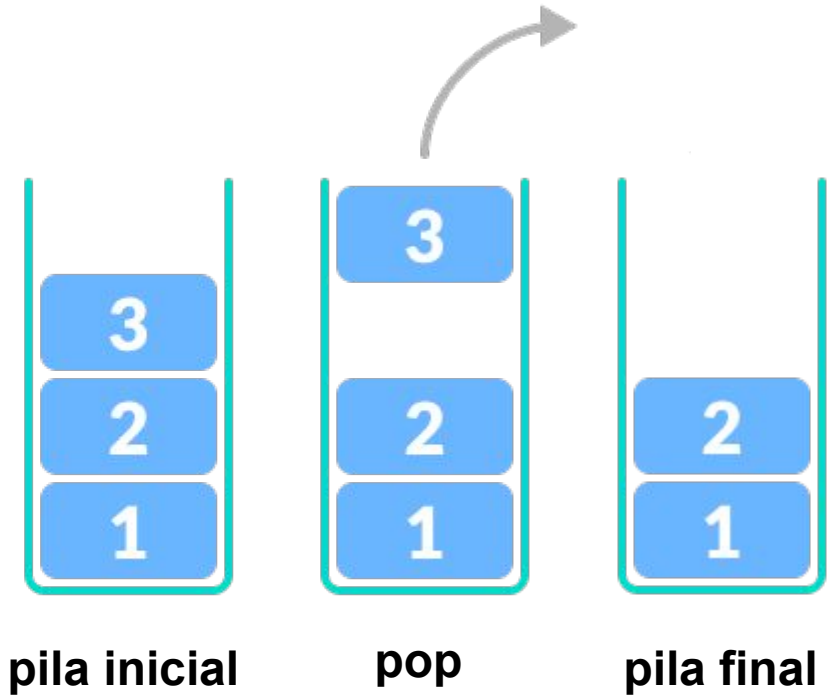
Operación *apilar*

- **Ejemplo:**
  - Supongamos que tenemos una pila de enteros vacía y queremos apilar los valores 1, 2 y 3:



Operación *desapilar*

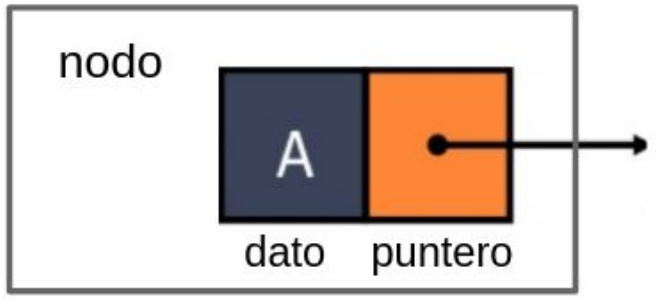
- **Ejemplo:**
  - Supongamos que tenemos una pila de enteros con los valores 1,2 y 3, siendo 3 el elemento del tope de la pila. Necesitamos extraer el elemento del tope:



Implementación de estructura de datos de TDA pila

- La estructura de datos que representa un nodo es la siguiente:

```
typedef struct nodoGenerico
{
  int dato;
  struct nodoGenerico* puntero;
}nodo;
```

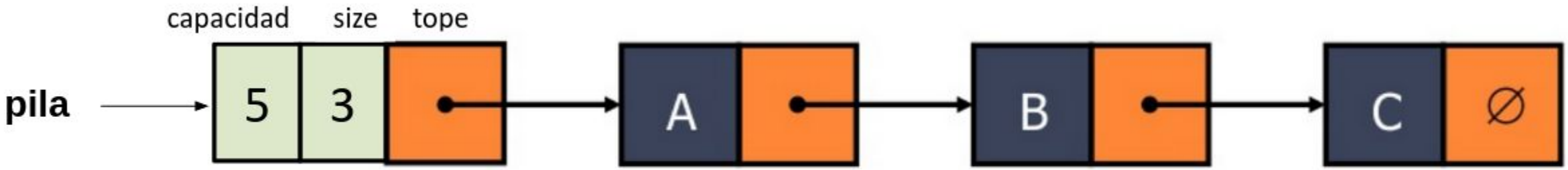


DEPARTAMENTO DE
INGENIERÍA
INFORMÁTICA
UNIVERSIDAD DE SANTIAGO DE CHILE

Implementación de estructura de datos de TDA pila

- La estructura de datos que representa una pila es la siguiente:

```
typedef struct
{
  int capacidad;
  int size;
  nodo* tope;
}TDApila;
```



DEPARTAMENTO DE
INGENIERÍA
INFORMÁTICA
UNIVERSIDAD DE SANTIAGO DE CHILE

Implementación de operaciones de TDA pila

```
TDApila* crearPilaVacia(int capacidad)
{
  TDApila* pila=(TDApila*)malloc(sizeof(TDApila));
  pila->capacidad=capacidad;
  pila->size=0;
  pila->tope=NULL;
  return pila;
}

int esPilaVacia(TDApila* pila)
{
  if (pila->size == 0)
    return 1;
  else
    return 0;
}
```

Actividades de implementación

Actividad 1 - individual

1. Compilar y ejecutar **lab07-pilas.c**
2. Experimentar con las funciones implementadas en **TDApila.h** haciendo llamadas desde función *main()* en **lab07-pilas.c**:
   - crear una pila vacía de capacidad 8
   - verificar si la pila creada está vacía

Actividad 2 - individual

1. Implementar la siguiente función en **TDApila.h**:

```
nodo* tope(TDApila* pila)
```

Actividad 3 - en conjunto

1. En conjunto, implementar la siguiente función en **TDApila.h**:

```
void apilar(TDApila* pila, int dato)
```

2. Evaluar la función creada, generando secuencia de llamadas desde función main() en lab07-pilas.c para crear la pila (tope) 8 5 9 6 2. Después de apilar cada dato, se debe mostrar el tope de la pila.

Actividad 4 - individual

1. Implementar la siguiente función en **TDApila.h**:

```
void desapilar(TDApila* pila)
```

2. Evaluar la función creada, generando secuencia de llamadas desde función main() en lab07-pilas.c para dejar vacía la pila creada en Actividad 3. Después de desapilar cada dato, se debe mostrar el tope de la pila.

Actividad 5 - individual

1. Implementar la siguiente función en **TDApila.h**

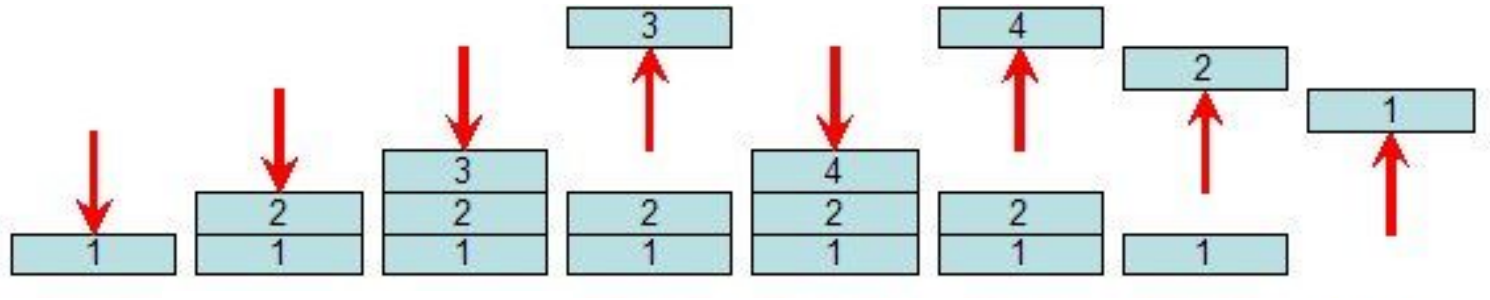
```
int buscarDatoPila(TDApila* pila, int dato)
```

La función *buscarDatoPila* devuelve 1 si el dato ingresado se encuentra en la pila, y 0 en caso contrario.

2. Evaluar la función creada, generando secuencia de llamadas desde función *main()* en lab07-pilas.c

Actividad 6- individual

1. Evaluar todas las funciones creadas en **TDApila.h**, representando el siguiente ejemplo a través de llamadas desde función *main()* de **lab07-pilas.c** y mostrando el dato que se encuentra en el tope de la pila después de cada operación:



Actividad 7 - individual

1. Implementar la siguiente función en **TDApila.h** que ordene ascendentemente los elementos de *pila*, con el menor elemento en el fondo y el mayor elemento en el tope (se pueden ocupar únicamente PILAS auxiliares)

```
void ordenarAscendente(TDApila* pila)
```

2. Evaluar la función creada, generando secuencia de llamadas desde función main() en lab07-pilas.c para ordenar pila creada en Actividad 3.

Entrega de actividad de laboratorio

- Entrega *obligatoria*
- Subir SOLO actividades 2, 3, 4 y 5 de esta sesión en buzón de uVirtual, en único archivo **s7_apellido_nombre.zip**
- Se espera lab07-pilas.c y TDApila.h (ambos modificados para responder a actividades 2, 3, 4 y 5) comprimidos en archivo .zip
- Plazo: **hoy** dentro del horario de laboratorio de cada coordinación

Actividad de cierre

- Ir a menti.com e ingresar código 4535 8151

Próximas fechas...

- Resumen de la semana:
  - TDA pila
  - TDA cola

  ~~cátedra~~ – ~~refuerzo~~ – ~~laboratorio~~

- Próxima semana:
  - TDA grafo

U2 - S7