# Pila

1. Estructura TPila.

TPila consiste en 4 campos:

* *Void \*\*pila:* en este campo se guardan las direcciones de los elementos de la pila.
* *Int maxDatos:* como el nombre lo indica, este campo almacena la cantidad máxima de datos de la pila.
* *Int tamDatos:* como el nombre lo indica, este campo almacena el tamaño de los datos almacenados en la pila.
* *Int tos:* Top Of Stack. Este campo contiene el índice de último elemento de la pila.

1. TPila \*nuevaPila (int tamDato,int maxDatos)

* *Función:* Crea una nueva pila con memoria asignada y datos inicializados.
* *Pasos:*

1. Crea un nuevo apuntador de tipo TPila llamado pila.
2. Le asigna al apuntador una dirección de memoria a través de la función malloc con el tamaño de la estructura TPila.
3. Le asigna al campo pila de pila una dirección de memoria a través de la función malloc con el tamaño de maxDatos apuntadores de tipo void.
4. Le asigna al campo maxDatos de la pila el entero maxDatos recibido.
5. Le asigna al campo tamDato de la pila el entero tamDato recibido.
6. Se le asigna al campo tos de la pila el valor -1.
7. Se regresa pila.
8. Int vacia (TPila \*pila)

* *Función:* Verifica si la pila se encuentra vacía.
* *Pasos:*

1. Entra una condición. Si el campo tos de la pila recibida es -1, la función regresa uno.
2. Regresa cero.
3. Int llena (TPila \*pila)

* *Función:* Verifica si la pila se encuentra llena.
* *Pasos:*

1. Regresa si el índice del tope del stack es mayor o igual al máximo de datos menos uno.
2. Void pop (TPila \*pila, void \*dato)

* *Función:* Saca el último elemento de la pila, eliminándolo de la pila y regresando el dato eliminado.
* *Pasos:*

1. Entra una condición que verifica si la pila está vacía.
2. Si la pila no está vacía se copia el dato del tope de la pila a la dirección dato pasada como parámetro a través de memcpy con el tamaño almacenado en el campo tamDato de pila (Sintáxis: a donde se va a copiar, de donde se va a copiar, tamaño de lo que se va a copiar).
3. Libera la dirección de memoria del tope del stack a través de la función free.
4. Le resta una unidad al valor del campo tos de la pila.
5. Regresa a la función principal.
6. Si no entra la condición, imprime el mensaje “Error: Stack underflow” y regresa a la función principal.
7. Void push (TPila \*pila, void \*valor)

* *Función:* Agrega el elemento dado como tope de la pila.
* *Pasos:*
  1. Entra una condición que verifica si la pila está llena.
  2. Si la pila no está llena, se aumenta en una unidad la cantidad almacenada en el campo tos de la pila.
  3. Le asigna al tope de la pila una dirección a través de la función malloc con el tamaño almacenado en el campo tamDato de la pila.
  4. Copia el dato almacenado en el apuntador valor obtenido como parámetro a la dirección en el tope de la pila con el tamaño del dato.
  5. Regresa a la función principal.
  6. Si no entra a la condición, solo regresa a la función principal.

1. Void top (TPila \*pila, void \*tope)

* *Función:* Copia en la dirección recibida el dato actual del tope de la pila.
* *Pasos:*
  1. Entra una condición que verifica si la pila está vacía.
  2. Si la pila no está vacía, se copia a través de memcpy la información almacenada en el tope de la pila a la dirección tope recibida como parámetro, con el tamaño del campo tamDato de la pila.
  3. Regresa a la función principal.
  4. Si no se cumple la condición, el tope se iguala a null y se regresa a la función principal.

1. Void liberaPila (TPila \*pila)

* *Función:* Libera toda dirección almacenada en la pila y luego libera la dirección de la pila. Elimina la pila.
* *Pasos:*
  1. Entra un ciclo while que se ejecuta mientras el índice del tope de la pila, es decir, el campo tos de la pila es mayor o igual a cero.
  2. Dentro del ciclo, libera la dirección almacenada en el tope de la pila a través de la función free, después se le resta una unidad al campo tos de la pila.
  3. Con el ciclo finalizado, libera la dirección de la pila a través de la función free.
  4. Regresa a la función principal.

1. Void imprimePila (TPila \*pila, void (\*función) (void \*info))

* *Función:* Con ayuda de una función pasada como parámetro, imprime cada elemento de la pila.
* *Pasos:*
  1. Crea un entero k inicializado en cero.
  2. Imprime el texto “La pila es:”.
  3. Entra un ciclo while que se ejecuta mientras el entero k sea menor o igual al campo tos de la pila.
  4. Dentro del ciclo, imprime k, llama a la función pasada como parámetro y pasa el elemento de la pila de índice k como parámetro, se añade una unidad a k.
  5. Regresa a la función principal.