**ESTRUCTURA DE PILA**

Nava Soler Raúl

Jeronimo Rojano Kevin Marzul

Antonio Ramírez Juan Elias

**Introducción**

En el estudio de las estructuras de datos, una de las más fundamentales es la pila. Una pila es una estructura de datos lineal que sigue el principio LIFO (Last In, First Out), lo que significa que el último elemento en ser insertado es el primero en ser eliminado.

En esta práctica, nos enfocaremos en la implementación de una pila utilizando arreglos en el lenguaje de programación C. El uso de arreglos nos permite entender el funcionamiento de esta estructura a un nivel más bajo, ya que nos obliga a gestionar manualmente los límites de la estructura.

El objetivo principal de esta práctica es comprender cómo se pueden manipular los elementos de una pila mediante operaciones básicas como push y pop, utilizando únicamente un arreglo para almacenar los datos.

**Métodos y materiales**

La implementación se realiza en un archivo de código fuente en C. La estructura de datos utilizada es un arreglo estático de tamaño fijo. Se definen varias funciones para manipular la pila, incluyendo la de push, pop, peek y verificar.

* **Herramientas Utilizadas Lenguaje de Programación:** C
* **Compilador:** gcc.exe (tdm64-1) 10.3.0 Copyright (C) 2020 Free Software Foundation, Inc. Esto es software libre; vea el codigo para las condiciones de copia. NO hay garantia; ni siquiera para MERCANTIBILIDAD o IDONEIDAD PARA UN PROPOSITO EN PARTICULAR
* **Entorno de Desarrollo:** Visual Studio Code
* **Versión de C:** C23 o ISO/IEC 9899:2024.

**Descripción General del Código**

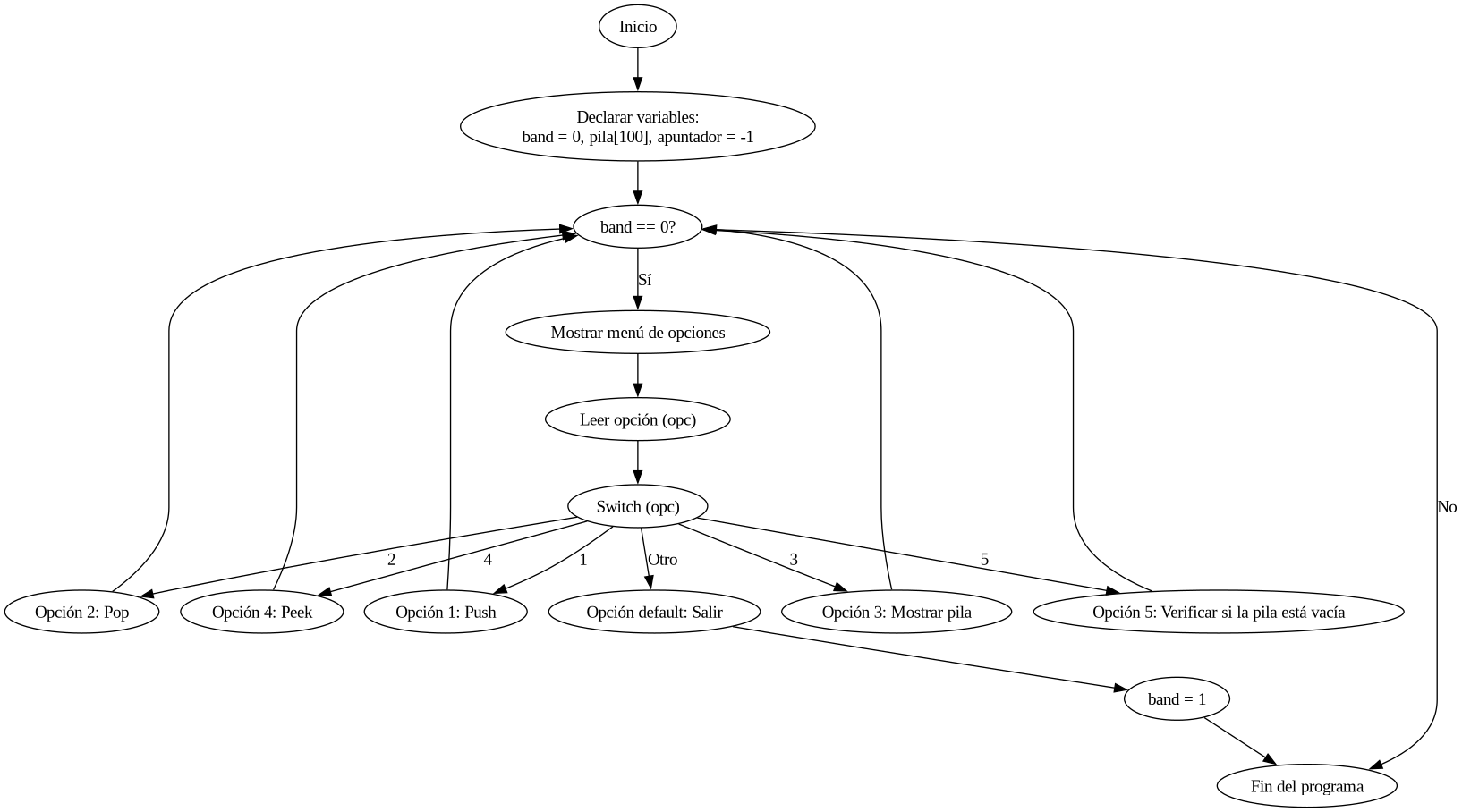
El código está compuesto por una estructura de datos y varias funciones que operan sobre esta estructura. A continuación, se presenta un diagrama de flujo que ilustra el funcionamiento general del programa:

**Desarrollo**

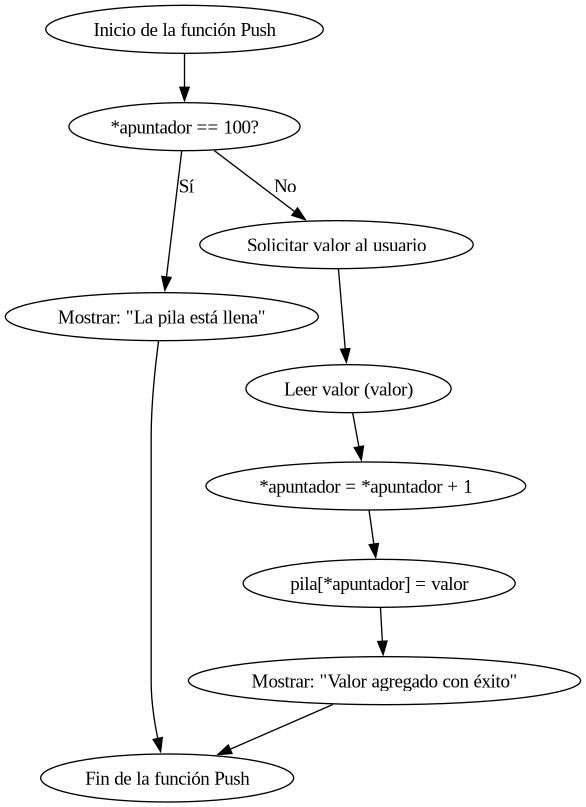
Se implementaron las siguientes funciones para manipular la lista:

1. **push**: Añade un elemento al final de la pila.
2. **pop**: Elimina el último elemento de pila.
3. **mostrar**: Muestra los elementos de la pila.
4. **peek**: Nos muestra el último elemento de la pila.

Cada función se explica a detalle a continuación:

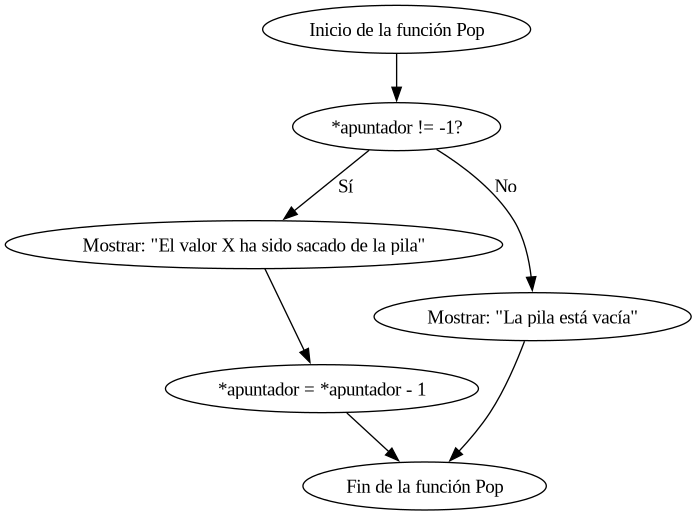


**Función** push



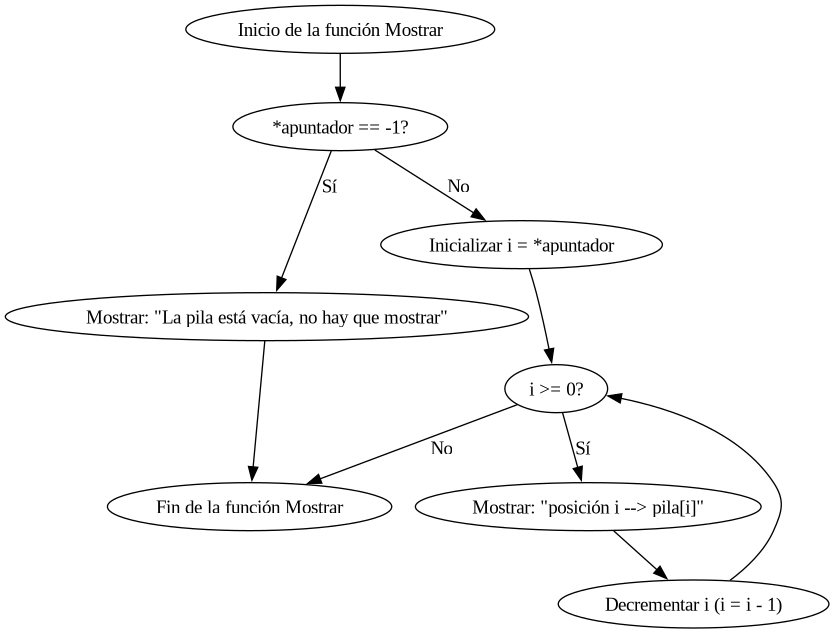
Esta función agrega un nuevo valor al final de la pila. Si la pila está llena, muestra un mensaje de error. En caso contrario, agrega el nuevo valor al final y actualiza el índice.

**Función**  pop

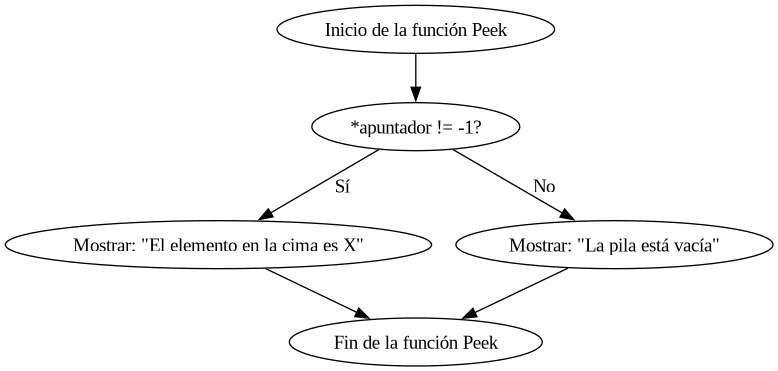


Esta función muestra y elimina el último elemento en ser agregado de la pila. Si la pila está vacía, muestra un mensaje de error. En caso contrario, muestra y elimina el ultimo elemento en ser agregado o el primero de la pila y actualiza el índice.

**Función** mostrar



Esta función muestra todos los elementos de la pila. En caso de estar vacía se lo notificará al usuario, de otro modo mostrará todos los elementos de la pilá del ultimo al primero.



**Función** peek

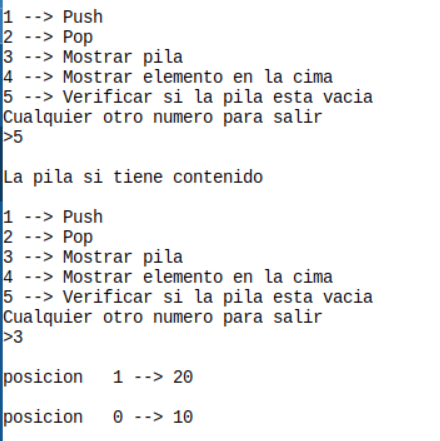
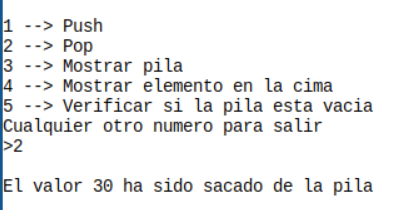
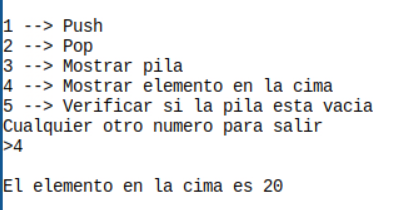
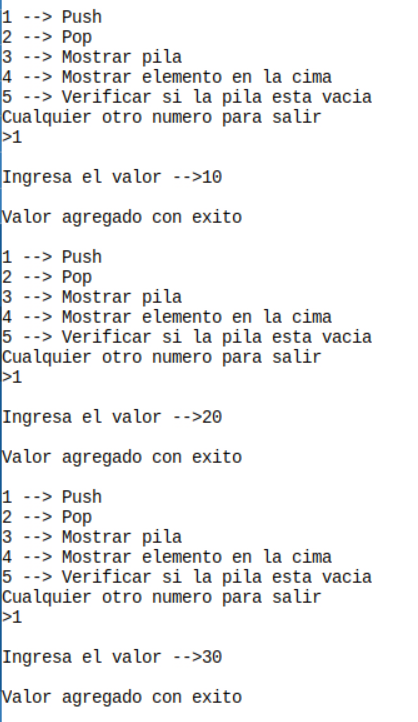
Esta función mostrará el último elemento de la pila en ser agregado.

**Conclusión**

El programa cumple con la funcionalidad básica de una lista utilizando arreglos en C. Permite insertar elementos, eliminar el último elemento en ser agregado, mostrar el elemento que este arriba de la pila y mostrar la pila completa.

La implementación utiliza un arreglo estático de tamaño fijo, lo cual limita la capacidad de la pila a 100 elementos. Esta limitación se debe a la decisión de no utilizar memoria dinámica.

Los resultados obtenidos en nuestro programa son los siguientes:



**Limitaciones**

**Capacidad Fija**: La pila tiene una capacidad fija de 100 elementos, lo cual puede ser insuficiente para algunas aplicaciones.

**Flexibilidad**: Al no utilizar memoria dinámica, la implementación carece de flexibilidad para adaptarse a diferentes tamaños de pilas.

**Áreas de Mejora**

**Memoria Dinámica**: Implementar a la pila memoria dinámica permitiría una capacidad variable y mejoraría la flexibilidad del programa.

**Errores y Validaciones**: Mejorar la gestión de errores y validaciones para proporcionar una experiencia de usuario más robusta.

El programa funciona correctamente dentro de sus limitaciones. La pila permite realizar todas las operaciones básicas requeridas. Sin embargo, la capacidad limitada es un área que podrían mejorarse en futuras versiones.