PILA

- **DEFINICIÓN**

Una pila es una secuencia de elementos en la que todas las operaciones se realizan por un extremo de la misma. Dicho extremo recibe el nombre de tope o cima.

- **ESPECIFICACIÓN**
  - Pila()
    Post: Crea una pila vacía
  - Bool vacia()const
    Post: Devuelve true si la pila está vacía
  - Const tElemento& tope() const
    Pre: La pila no está vacía
    Post: Devuelve el elemento del tope de la pila
  - Void pop()
    Pre: La pila no está vacía
    Post: Elimina el elemento del tope de la pila y el siguiente se convierte en el nuevo tope
  - Void push(const tElemento& x)
    Post: Inserta el element x en el tope de la pila y el antiguo tope pasa a ser el siguiente.

- **IMPLEMENTACIÓN VECTORIAL ESTÁTICA**

Exige fijar el tamaño máximo de la pila en tiempo de compilación, por lo que el tamaño tendrá que ser una constante y el mismo para todas las pilas.

- **IMPLEMENTACIÓN VECTORIAL PSEUDOESTÁTICA**

Cuando queremos tener pilas de distintos tamaños máximos en un programa. Es importante señalar que este tamaño será constante durante el tiempo de vida de cada pila.

Almacena dos campos del vector y la posición del tope, requiere un campo para almacenar el tamaño máximo(LMAX)

- **IMPLEMENTACIÓN VECTORIAL MEDIANTE ESTRUCTURAS ENLAZADAS(DINÁMICA)**

El tamaño es variable y siempre ocupará el espacio justo para almacenar los elementos que contenga la pila en cada momento.

Consistirá en una secuencia de registros enlazados con punteros, en la que cada registro almacenará un elemento y la pila será un puntero al primer elemento de la secuencia(tope), por donde se realizan las inserciones y borrados. Se hace por referencia. El espacio de memoria ocupado es menor a la estática o pseudoestática.

- **PREGUNTAS**
  - En el TAD Pila, puedo guardar un elemento menos debido a que en el nodo cabecera no se guarda nada.
    FALSO

  - El TAD Pila no incorpora una operación concreta para acceder a un elemento cualquiera de la misma, pero sí podemos acceder a cualquier elemento de la Pila, siempre y cuando la implementación del TAD sea utilizando una representación vectorial.
    FALSO

  - En la estructura vectorial del TAD Pila, no es posible aprovechar todos los elementos del vector ya que no nos permite distinguir entre pila llena y pila vacía.
    FALSO

  - No tiene sentido implementar una pila mediante un estructura doblemente enlazada porque, aunque el doble puntero es útil, resulta muy costoso en términos de espacio.
    FALSO

- Si he de escoger entre una estructura simplemente enlaza y doblemente enlazada a la hora de implementar una pila, la decisión dependerá de si alguna operación en concreto se va a realizar con mucha frecuencia o no.
  FALSO

- El TAD Pila es útil para resolver, entre otros, problemas en los que es necesario procesar los elementos en orden inverso al que se proporcionan.
  VERDADERO

- Aunque desconozcamos el tamaño máximo del problema, es posible, aunque ineficiente, usar una estructura vectorial pseudoestática.
  FALSO

- Si conozco a priori el tamaño máximo del problema, desde el punto de vista de la eficiencia espacial, no debo utilizar nunca una estructura enlazada, siempre mejor una vectorial.
  VERDADERO

- La representación vectorial circular del TAD Pila deja un hueco libre para distinguir cuando está llena o vacía.
  FALSO

- En la representación dinámica del TAD Pila, no es necesario añadir por un nodo cabecera, pero si lo ponemos, nos facilita el acceso al elemento que está en el tope.
  FALSO

- El TAD Pila no incorpora una operación concreta para acceder a un elemento cualquiera de la misma, pero sí podemos hacer el cambio elemento de la Pila, implementando la operación a partir de las operaciones del TAD.
  VERDADERO