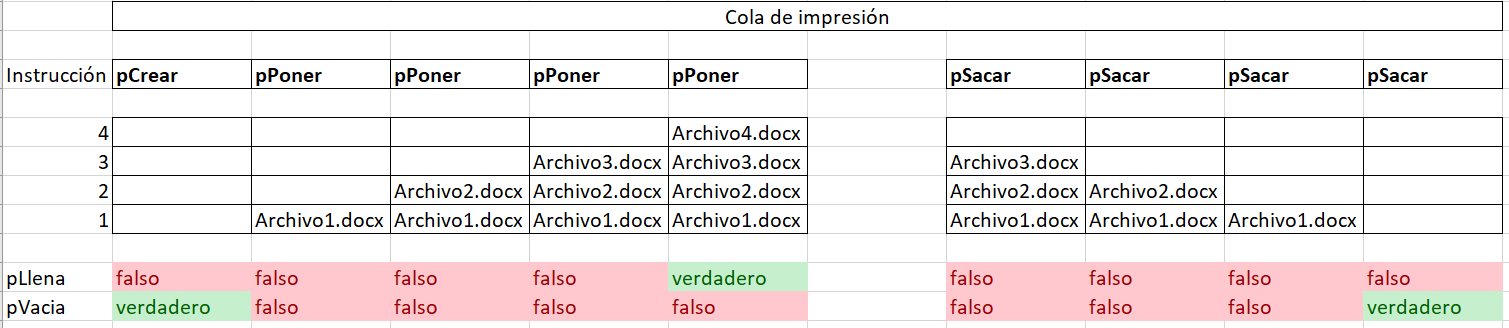
1) Consulte la bibliografía y conteste las siguientes preguntas :

**Qué es una pila?**

Es una estructura ordenada de datos que se caracteriza principalmente por el orden en el que sus elementos son tratados. Las pilas tratan primero al último elemento que haya ingresado en ella.

**Arme un ejemplo, ponga y saque datos de una pila.**



**Piense varios ejemplos de la vida real en que el comportamiento sea el de una pila.**

Meter ropa en un cesto, luego, sacar la ropa del cesto para plancharla

Un estacionamiento en única fila

Una cola de impresión

Varias personas suben a un taxi por una sola puerta, y bajan por la misma puerta

**Qué operaciones de acceso se usan para almacenar y acceder a elementos individuales de datos?**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tipo** | **Operador** | **Firma** | **Resultado** | **Precondciones** |
| Procedimiento | pPoner | (ref P: tPila; valor x: tdato) |  | No(pLlena) |
| Procedimiento | pSacar | (ref P: tPila; ref x: tdato) |  | No(pVacia) |

Qué significa LIFO?

Del inglés Last In First Out, hace referencia al orden en que se tratan los elementos al trabajar con una cola.

Defina el tipo de datos abstracto (ADT ó TDA) "Pila".

tPila

Abstracciones de datos

tPila no implementa tipos de datos abstractos.

Abstracciones procedimentales

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Tipo** | **Operador** | **Firma** | **Resultado** | **Precondciones** |
| Funcion | pVacía | (ref p: tPila) | booleano |  |
| Funcion | pLlena | (ref p: tPila) | booleano |  |
| Procedimiento | pCrear | (ref p: tPila) |  |  |
| Procedimiento | pPoner | (ref P: tPila; valor x: tdato) |  | No(pLlena) |
| Procedimiento | pSacar | (ref P: tPila; ref x: tdato) |  |  |

Qué significa "ocultamiento de la información" (information hiding)? Qué ventajas tiene? Cómo se relaciona con el tema que estamos viendo?

Se refiere a la capacidad de ocultar los detalles de implementación de una parte o módulo del software; esto tiene muchas ventajas, como la simplificación de las rutinas que dependen de dicho módulo, y como consecuencia, la facilidad para verificar el funcionamiento de dichas rutinas, y, en mi opinión personal, el bajísimo acoplamiento que se logra al escribir rutinas que dependen de abstracciones (Principio conocido como inversión de control).

En nuestro pseudocódigo, usamos abstracciones, las cuales ocultan detalles de implementación, todo el tiempo, de forma que podamos concentrarnos en la resolución de los problemas dados. Si tuviésemos que escribir las rutinas de tPila por cada programa que escribimos, no aprobaríamos nunca la materia.