

## PROYECTO II

Elaborar una aplicación de tareas pendientes que le permita al usuario gestionar sus tareas de una manera flexible teniendo en cuenta los siguientes requerimientos:

1. La aplicación debe tener un menú con las siguientes secciones:

- 1. Lista de tareas
  - Sub-menu:
    - ◆ Completadas
    - ◆ Pendientes
    - ◆ Atrás
- 2. Filtrar tareas
  - Sub-menu
    - ◆ Filtrar por título
    - ◆ Atrás
- 3. Añadir tarea
- 4. Actualizar tarea
  - Sub-menu
    - ◆ Actualizar título
    - ◆ Actualizar descripción
    - ◆ Actualizar status (completado o pendiente)
    - ◆ Atrás
- 5. Eliminar tarea (eliminar tarea por código único)
- 6. Salir

## 2. Requerimientos:

- Lista de tareas: deberá mostrar un sub-menu con tareas **completadas** y **pendientes** donde el usuario al seleccionar una de ellas deberá mostrar las tareas que posean dicho status seleccionado, por lo que se debe aplicar filtro de datos por status. La opción de **atrás** deberá llevar a la sección principal
- Filtrar tareas: deberá mostrar un sub-menu con **filtrar por título** lo que le permitirá al usuario obtener los posibles títulos que comiencen por el dato que se introduzca en pantalla. Esto quiere decir que, si una tarea tiene como título **Aprender Python** y el usuario escribe **aprender**, deberá mostrar todas las posibles opciones que comiencen por **aprender**, **no debe ser sensible a minúsculas y mayúsculas** por lo que se debe buscar la manera de hacer comparaciones correctas
- Añadir tarea: se le pedirá al usuario que introduzca el **código (identificador único, valor numérico)**, **título**, **descripción**, **status**, y **todos los campos deben ser validados y no pueden estar vacíos**, el **código** debe estar conformado por un valor numérico entero no repetido, el **título** solo puede tener una longitud de veinte (20) caracteres, la **descripción** puede tener cualquier longitud pero no puede estar vacío, el **status** solo debe estar conformado por dos opciones que son **completado** y **pendiente (True y False)**, si el usuario añade otra entrada se debe arrojar un mensaje de error y volver a pedir el campo correspondiente
- Actualizar tarea: **buscar tarea por código único** y mostrar el sub-menu ilustrado en el primer punto, solo se debe actualizar la opción

seleccionada por el usuario y se debe validar de manera correspondiente cada campo

- Eliminar tarea: ***eliminar tarea por código único***
- Salir: permite finalizar el programa

### 3. Consideraciones:

- Leer atentamente los requerimientos antes de empezar a crear la app
- Validar correctamente los campos según los requerimientos
- En lista de tareas y filtrar tareas se debe mostrar todos los datos de cada tarea a través de una lista imprimida en pantalla
- ***El código de cada tarea no se debe repetir***, por lo que si se intenta añadir una tarea con un código existente se debe mostrar un mensaje de error y volver a pedir la entrada al usuario, solo se aceptan números enteros, si se intenta añadir un valor flotante, letra, etc, se deberá volver a pedir entrada al usuario
- Se debe hacer uso de ***map, filter*** o ***comprensión de listas*** de manera obligatoria, ***la función reduce es opcional***
- ***El código debe estar modularizado en funciones***
- Se trabaja con el ciclo ***while***

### 4. Fecha de entrega

Posibilidad de entrega hasta el **11-08-2024 12PM**