**Documentación de la Aplicación Lista de Tareas**

**Tabla de Contenidos**

1. [Introducción](#introducción)
2. [Estructura de Clases](#estructura-de-clases)
   1. [Clase Task](#clase-task)
   2. [Clase Program](#clase-program)
3. [Flujo Principal del Programa](#flujo-principal-del-programa)
4. [Funciones Principales](#funciones-principales)
   1. [AddTask()](#addtask)
   2. [ListTasks()](#listtasks)
   3. [CompleteTask()](#completetask)
   4. [DeleteTask()](#deletetask)
5. [Manejo de Errores](#manejo-de-errores)
6. [Posibles Mejoras](#posibles-mejoras)

**1. Introducción**

Este documento proporciona una visión general de una aplicación simple de Lista de Tareas implementada en C#. La aplicación permite a los usuarios gestionar tareas a través de una interfaz de consola, ofreciendo funcionalidades como agregar tareas, listar tareas, marcar tareas como completadas y eliminar tareas.

**2. Estructura de Clases**

2.1. Clase Task

La clase `Task` representa una tarea individual en la Lista de Tareas.

Propiedades:

- `Descripcion` (string): La descripción de la tarea.

- `DueDate` (DateTime?): La fecha de vencimiento de la tarea (puede ser nula).

- `Completada` (bool): Indica si la tarea está completada.

Métodos:

- `Constructor`: Inicializa una nueva tarea con una descripción y una fecha de vencimiento opcional.

- `MarkAsCompleted()`: Marca la tarea como completada.

- `ToString()`: Proporciona una representación en cadena de la tarea.

2.2. Clase Program

La clase `Program` contiene la lógica principal del programa y la interfaz de usuario.

Campos Estáticos:

- `tasks` (List<Task>): Almacena la lista de tareas.

**3. Flujo Principal del Programa**

La aplicación se ejecuta en un bucle continuo, presentando un menú al usuario con las siguientes opciones:

1. Agregar Tareas

2. Listar tareas registradas

3. Marcar tareas como completadas

4. Eliminar Tareas

5. Salir

El programa procesa la elección del usuario y llama al método apropiado para manejar cada acción.

**4. Funciones Principales**

4.1. AddTask()

Esta función solicita al usuario que ingrese una descripción de la tarea y una fecha de vencimiento opcional. Luego crea un nuevo objeto `Task` y lo agrega a la lista `tasks`.

4.2. ListTasks()

Esta función muestra todas las tareas en la lista `tasks`, mostrando su descripción, fecha de vencimiento y estado de completado.

4.3. CompleteTask()

Esta función permite al usuario marcar una tarea como completada ingresando su número en la lista.

4.4. DeleteTask()

Esta función permite al usuario eliminar una tarea de la lista ingresando su número.

**5. Manejo de Errores**

La aplicación utiliza un bloque try-catch en el bucle principal para capturar y mostrar cualquier excepción que ocurra durante la ejecución. Las comprobaciones de errores específicas incluyen:

- Validación de la entrada del usuario para las opciones del menú

- Verificación de descripciones de tareas nulas o vacías

- Validación del formato de entrada de fecha

- Verificación de números de tarea válidos al completar o eliminar tareas