Universidade Paulista

DIEGO KAUJON BISCAIA – RA: G50JBB5

Atividade 3 – Disciplina: Projeto de Sistemas Orientados a Objetos

Sistema de Criação de Listas de tarefas

Sorocaba / 2023

1. **Introdução**

O Projeto a seguir se trata de um software que permite ao usuário dele, criar tarefas, edita-las e excluí-las a qualquer momento.

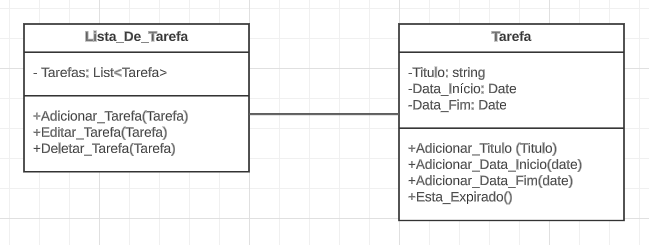
1. **Detalhes técnicos e Funcionamento**

O Software foi feito com a utilização da Tecnologia .NET C# (Csharp) para o Sistema operacional Windows Desktop.

1. **Diagrama de Classes UML**

Na Figura 1 pode-se observar um diagrama de classes UML, que foi feito para mostrar os atributos e métodos das classes existentes no software em questão. Foi utilizada a plataforma LucidChart para a criação do Diagrama da Figura 1

Figura 1



Fonte: Autoria Própria.

* 1. Explicação dos atributos e métodos do diagrama de classes

1. Classe: Lista\_De\_Tarefa: Esta classe representa a lista de tarefas do usuário. Ela contém uma lista de objetos “Tarefa” e fornece métodos para adicionar, editar e excluir tarefas.

* Atributos:
  1. Tarefas: Uma lista de Objetos “Tarefa” para armazenar as tarefas.
* Métodos:
  1. Adicionar\_Tarefa(Tarefa): Adiciona uma tarefa à lista de tarefas.
  2. Editar\_Tarefa(Tarefa): Edita uma tarefa existente na lista.
  3. Deletar\_Tarefa(Tarefa): Exclui uma tarefa da lista.

1. Classe: Tarefa: Esta classe representa uma tarefa individual.

* Atributos:

1. Titulo: Uma string que armazena o título da tarefa.
2. Data\_Inicio: Uma data que representa a data de início da tarefa.
3. Data\_Fim: Uma data que representa a data de término da tarefa.

* Métodos:

1. Adicionar\_Titulo(Titulo): Define o título da tarefa.
2. Adicionar\_Data\_Inicio(date): Define a data de início da tarefa.
3. Adicionar\_Data\_Fim(date): Define a data de término da tarefa.
4. Esta\_Expirado(): Verifica se a tarefa está expirada com base na data atual.
5. **Código fonte do Software**

using System;

using System.Collections.Generic;

// Classe Tarefa

public class Tarefa

{

public string Titulo { get; set; }

public DateTime Data\_Inicio { get; set; }

public DateTime Data\_Fim { get; set; }

public Tarefa(string Titulo, DateTime Data\_Inicio, DateTime Data\_Fim)

{

Titulo = titulo;

Data\_Inicio = data\_inicio;

Data\_Fim = data\_fim;

}

public void Adicionar\_Titulo(string titulo)

{

Titulo = titulo;

}

public void Adicionar\_Data\_Inicio(DateTime data\_inicio)

{

Data\_Inicio = data\_inicio;

}

public void Adicionar\_Data\_Fim(DateTime data\_fim)

{

Data\_Fim = data\_fim;

}

public bool Esta\_Expirado()

{

DateTime currentDate = DateTime.Now.Date;

return currentDate > Data\_Fim;

}

}

// Classe Lista\_De\_Tarefa

public class Lista\_De\_Tarefa

{

private List<Tarefa> Tarefas = new List<Tarefa>();

public void Adicionar\_Tarefa(Tarefa tarefa)

{

tarefas.Add(tarefa);

}

public void Editar\_Tarefa(Tarefa tarefa, string novoTitulo, DateTime newData\_Inicio, DateTime newData\_Fim)

{

// Assume que tarefa é uma instância existente de Tarefa

if (tarefas.Contains(tarefa))

{

// Edita a tarefa

tarefa.Adicionar\_Titulo(newTitulo);

tarefa.Adicionar\_Data\_Inicio(newData\_Inicio);

tarefa.Adicionar\_Data\_Fim(newData\_Fim);

}

}

public void Deletar\_Tarefa(Tarefa tarefa)

{

if (tarefas.Contains(tarefa))

{

tarefas.Remove(tarefa);

}

}

}