

Relatório do Projeto "TaskBuddy"

1. Introdução

O projeto "TaskBuddy" foi concebido como um aplicativo de gerenciamento de tarefas para auxiliar os usuários na organização de suas atividades diárias. Este relatório detalha as escolhas de design feitas durante a criação do Diagrama de Classes para o "TaskBuddy" e destaca como essas escolhas estão alinhadas com os requisitos do projeto.

2. Escolhas de Classes

2.1 Task

A classe Task foi escolhida para representar uma tarefa no aplicativo. Esta escolha é fundamental para atender ao requisito de permitir que os usuários criem, editem, e marquem tarefas como concluídas.

2.2 User

A classe User representa um usuário do aplicativo, incluindo atributos para o nome de usuário, senha e uma lista de tarefas associadas. Isso é essencial para identificar usuários e vincular tarefas aos seus proprietários.

2.3 TaskManager

A classe TaskManager foi introduzida para gerenciar operações relacionadas a tarefas, como criar, editar e excluir tarefas. Essa escolha permite uma organização eficiente das operações de gerenciamento.

3. Relacionamentos

O Diagrama de Classes reflete os relacionamentos cruciais entre as classes.

3.1 Relacionamento User - Task

Existe uma relação de composição entre User e Task, indicando que um usuário pode ter várias tarefas associadas. Isso atende à necessidade de os usuários poderem criar várias tarefas.

3.2 Relacionamento TaskManager - Task

A classe TaskManager possui uma associação com Task, permitindo que gerencie várias tarefas. Essa relação é crucial para as operações de gerenciamento de tarefas.

4. Atributos e Métodos

4.1 Task

Atributos:

```
description: String
dueDate: Date
isCompleted: boolean
Métodos:
```

```
markAsCompleted(): void
```

4.2 User

Atributos:

```
username: String
password: String
tasks: List<Task>
Métodos:
```

```
addTask(task: Task): void
```

4.3 TaskManager

Atributos:

```
tasks: List<Task>
```

Métodos:

```
createTask(description: String, dueDate: Date): void
```

```
editTask(task: Task, newDescription: String, newDueDate: Date): void
```

```
deleteTask(task: Task): void
```

5. Padrões de Projeto

Foi aplicado o padrão Observer para notificar os usuários sobre mudanças nas tarefas. Esta escolha contribui para a flexibilidade e extensibilidade do sistema.

6. Modelo Estrutural

O Diagrama de Classes reflete uma organização lógica das classes e suas interações, proporcionando uma visão clara da estrutura do "TaskBuddy".

7. Conclusão

Este relatório detalhou as decisões de design para o Diagrama de Classes do projeto "TaskBuddy". A escolha cuidadosa de classes, relacionamentos e padrões de projeto contribui para a eficácia e manutenibilidade do sistema.

8. Participantes do Grupo

[Lucas Henrique Santos Figueiredo]