

Mudanças em Relação à Etapa Anterior

- A **classe Tarefa** foi alterada para ser **abstrata**, o que implica que não pode mais ser instanciada diretamente, servindo como base para outras classes de tarefas específicas.
- A **classe Calendário** foi removida do projeto.
- O **GerenciadorDeTarefas** agora possui **4 ArrayLists** de tarefas, sendo um para cada tipo de tarefa, ao invés de um único ArrayList como especificado anteriormente.
- A **classe BotaoCompleta** foi **removida** do projeto, não sendo mais necessária para a implementação atual.

Implementação:

Na classe HelloApplication, é realizada a interface com JavaFX. Nela, são instanciados o objeto Usuario e o GerenciadorDeTarefas.

- O objeto Usuario armazena informações relacionadas ao perfil do usuário, incluindo os seguintes atributos:
 - `private String nome;`
 - `private int inteligencia;`
 - `private int forca;`
 - `private int cultura;`
 - `private int sorte;`
 - `private int historico;`
 - `private int nivel;`
- O GerenciadorDeTarefas é responsável pela gestão de diferentes tipos de tarefas. Ele organiza essas tarefas utilizando uma ArrayList para cada tipo, sendo que as tarefas são derivadas de uma classe abstrata chamada Tarefa. A partir dessa classe base, são criadas subclasses específicas para lidar com cada tipo de tarefa.

O GerenciadorDeTarefas oferece funcionalidades para adicionar e remover tarefas, além de métodos *getter* que permitem obter o total de tarefas cadastradas e o total de tarefas concluídas e as tarefas em si.

A interface gráfica do programa é implementada com JavaFX, sendo atualizada dinamicamente sempre que uma tarefa é criada ou concluída. Essas atualizações refletem no progresso da barra de progresso e nas informações exibidas na tela.

Ao iniciar o programa, uma janela inicial solicita o nome do usuário, que será utilizado na interface ao longo do uso, proporcionando uma experiência mais personalizada.

Casos de Teste Realizados:

GerenciadorTarefasTest

testDeletaTarefaInvalidIndex: Verifica a exclusão de uma tarefa com índice inválido.

testDeletaTarefaInvalidTipo: Testa a exclusão de uma tarefa com tipo inválido.

Tarefa Tests

TarefaCulturaTest, TarefaForcaTest, TarefaInteligenciaTest, TarefaSorteTest: Validam a inicialização correta das tarefas, garantindo que os atributos (tipo, pontos, descrição, bônus) sejam configurados corretamente.

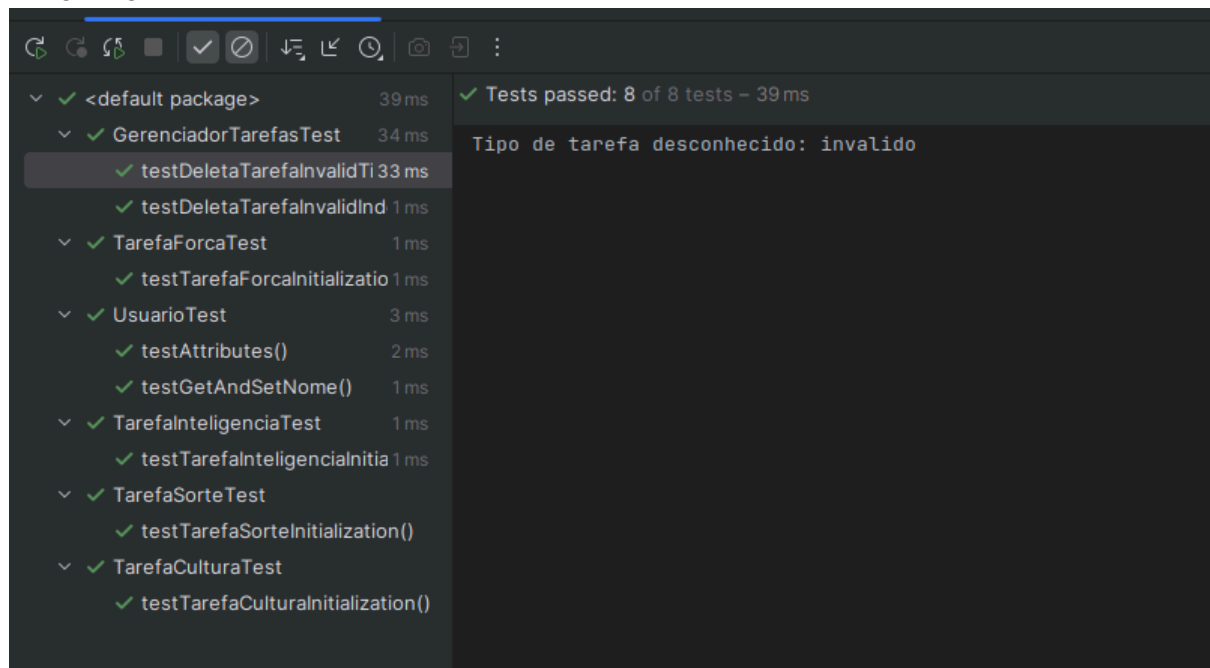
UsuarioTest

testGetAndSetName: Testa a atualização e recuperação do nome do usuário.

testAttributes: Verifica a configuração e acesso aos atributos principais do usuário (força, inteligência, cultura, etc.).

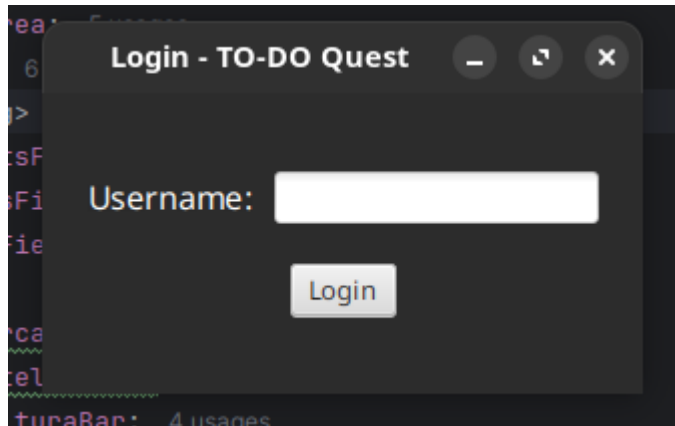
Experiência com Testes Unitários

Os testes unitários facilitaram a detecção de erros e deram maior confiança na qualidade do código. No início, o grupo encontrou dificuldades na estruturação dos testes, mas com a prática, o processo tornou-se mais eficiente. A abordagem ajudou a organizar melhor o código e garantiu maior robustez ao sistema.



Executável:

Ao abrir o programa, aparece uma tela perguntando o nome:



A interface inicial:



Tarefa criada:

TO-DO Quest

juan

Nível 0

Força

Inteligência

Cultura

Sorte

Força: 0/0 tarefas completadas

Inteligência: 0/1 tarefas completadas

Cultura: 0/0 tarefas completadas

Sorte: 0/0 tarefas completadas

Pontos

Bonus

Descrição

Add Task

+10

+10

Estudo

Data/Hora

Remover

Tarefa concluída:

TO-DO Quest

juan

Nível 0

Força

Inteligência

Cultura

Sorte

Força: 0/0 tarefas completadas

Inteligência: 1/1 tarefas completadas

Cultura: 0/0 tarefas completadas

Sorte: 0/0 tarefas completadas

Pontos

Bonus

Descrição

Add Task

Para rodar o programa:

1. No terminal digite: 'Glt clone <https://github.com/Andre-klarmann/TrabTCP.git>'.
2. Abra no inteliJ.
3. E clique em 'run program' que o inteliJ ira baixar as dependencias - JavaFX - e compilar e rodar o programa.