Trabalho Final TCP: TO-DO Quest

André Mainardi Klarmann
Juan Marcelo Trabaina
João Gabriel Rau Wendt
Lucas Leal Meregalli

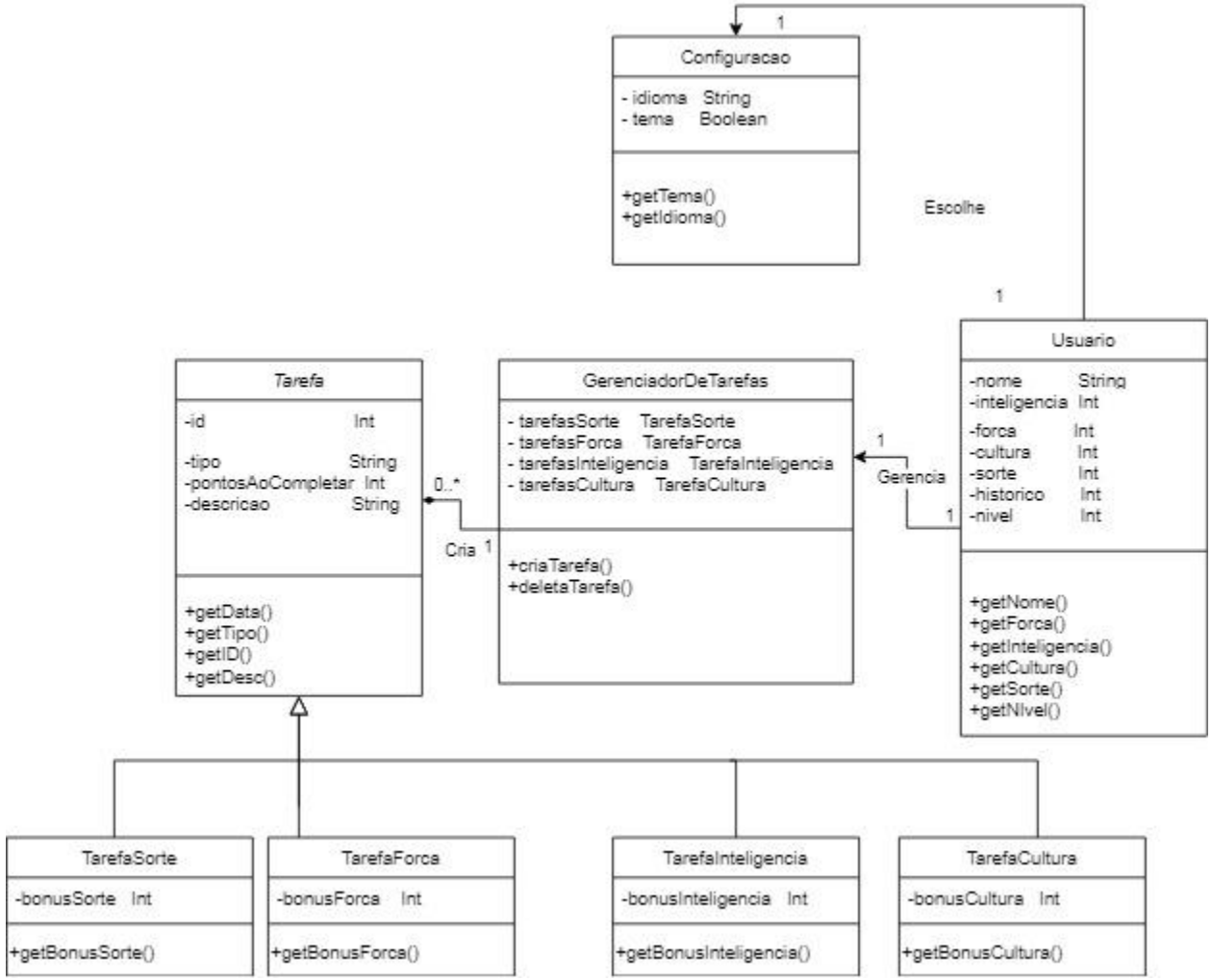
UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

.inf
INSTITUTO DE INFORMÁTICA
UFRGS

Descrição da Aplicação TO-DO Quest

A aplicação TO-DO Quest é um App de Tarefas que organiza as tarefas diárias de seus usuários de forma única, categorizando-as em quatro atributos principais: Inteligência, Força, Cultura e Sorte. Cada tarefa do usuário será associada a um desses quatro pilares, propondo uma gameficação do gerenciamento de atividades diárias, em que o progresso em diferentes áreas é refletido em desempenho e em nivelação do usuário.

Diagrama de Classes



UFRGS
UNIVERSIDADE FEDERAL
DO RIO GRANDE DO SUL

.inf UFRGS

Overview Do Código e Estruturação

Os objetos Usuario e GerenciadorDeTarefas são instanciados na classe HelloApplication (interface com JavaFX).

Algumas funções da classe GerenciadorDeTarefas:

```
public void criaTarefa(String tipo, int pontos, String descricao, int bonus) {
    switch (tipo.toLowerCase()) {
        case "forca":
            tarefasForca.add(new TarefaForca(pontos, descricao, bonus));
            totalForcaTasks++;
            break;
        case "inteligencia":
            tarefasInteligencia.add(new TarefaInteligencia(pontos, descricao, bonus));
            totalInteligenciaTasks++;
            break;
        case "sorte":
            tarefasSorte.add(new TarefaSorte(pontos, descricao, bonus));
            totalSorteTasks++;
            break;
        case "cultura":
            tarefasCultura.add(new TarefaCultura(pontos, descricao, bonus));
            totalCulturaTasks++;
            break;
        default:
            System.out.println("Tipo de tarefa desconhecido: " + tipo);
            break;
    }
}
```

```
public boolean deletaTarefa(String tipo, int indice) {
    switch (tipo.toLowerCase()) {
        case "forca":
            if (indice >= 0 && indice < tarefasForca.size()) {
                tarefasForca.remove(indice);
                return true;
            }
            break;
        case "inteligencia":
            if (indice >= 0 && indice < tarefasInteligencia.size()) {
                tarefasInteligencia.remove(indice);
                return true;
            }
            break;
        case "sorte":
            if (indice >= 0 && indice < tarefasSorte.size()) {
                tarefasSorte.remove(indice);
                return true;
            }
            break;
        case "cultura":
            if (indice >= 0 && indice < tarefasCultura.size()) {
                tarefasCultura.remove(indice);
                return true;
            }
            break;
        default:
            System.out.println("Tipo de tarefa desconhecido: " + tipo);
    }
    return false;
}
```

UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

.inf UFRGS

Overview Do Código e Estruturação

A interface gráfica da aplicação foi implementada com JavaFX, sendo atualizada dinamicamente sempre que uma tarefa é criada, removida ou concluída. Essas atualizações refletem na barra de progresso e nas demais 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.

Overview Dos Testes e Estruturação

GerenciadorTarefasTest

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

```
@Test
    void testDeletaTarefaInvalidIndex() {
        gerenciador.criaTarefa("forca", 50, "Exercício", 5);
        boolean removed = gerenciador.deletaTarefa("forca", 1); // no index 1
        assertFalse(removed);
        assertEquals(1, gerenciador.getTarefasForca().size());
    }
```

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

```
@Test
    void testDeletaTarefaInvalidTipo() {
        gerenciador.criaTarefa("forca", 50, "Exercício", 5);
        boolean removed = gerenciador.deletaTarefa("invalido", 0);
        assertFalse(removed);
        assertEquals(1, gerenciador.getTarefasForca().size());
    }
```

UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

.inf UFRGS

Overview Dos Testes e Estruturação

Tarefa Tests

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

Exemplo com TarefaSorte:

```
@Test
    void testTarefaSorteInitialization() {
        TarefaSorte tarefa = new TarefaSorte(25, "Jogar Loteria", 1);
        assertEquals("Sorte", tarefa.getTipo());
        assertEquals(25, tarefa.getPontos());
        assertEquals("Jogar Loteria", tarefa.getDescricao());
        assertEquals(1, tarefa.getBonus());
    }
```

Overview Dos Testes e Estruturação

UsuarioTest

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

```
@Test
void testGetAndSetNome() {
    assertEquals("TestUser", usuario.getNome());
    usuario.setNome("NewName");
    assertEquals("NewName", usuario.getNome());
}
```

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

```
@Test
void testAttributes() {
    usuario.setForca(10);
    usuario.setInteligencia(20);
    usuario.setCultura(15);
    usuario.setSorte(5);
    usuario.setHistorico(2);
    usuario.setNivel(4);

    assertEquals(10, usuario.getForca());
    assertEquals(20, usuario.getInteligencia());
    assertEquals(15, usuario.getCultura());
    assertEquals(5, usuario.getSorte());
    assertEquals(2, usuario.getHistorico());
    assertEquals(4, usuario.getNivel());
}
```

UFRGS
UNIVERSIDADE FEDERAL
DO RIO GRANDE DO SUL

.inf UFRGS

Demonstração

UFRGS
UNIVERSIDADE FEDERAL
DO RIO GRANDE DO SUL

.inf UFRGS

Instituto de Informática

Obrigado !

UFRGS
UNIVERSIDADE FEDERAL
DO RIO GRANDE DO SUL

.inf
INSTITUTO
DE INFORMÁTICA
UFRGS