

ETEC PROFESSOR HORÁCIO AUGUSTO DA SILVEIRA
ENSINO MÉDIO INEGRADO AO TÉCNICO EM DESENVOLVIMENTO DE
SISTEMAS

LUIZA GONÇALVES VENANCIO RODRIGUES
THABATA VITORIA DANIEL DE SOUZA

DISPUTAS ENTRE PERSONAGENS

São Paulo – SP

2025

ÍNDICE

1. INTRODUÇÃO	2
1.1 Propósito da Aplicação	2
2. REALIZAÇÃO DA IMPLEMENTAÇÃO E TESTES	3
2.1 Detalhamento da Implementação	3
2.2 Passo a passo da testagem da funcionalidade de Disputa	4
2.2.1 Estado Inicial:	4
2.2.2 Batalha entre Gandalf e Radagast:	5
2.2.3 Segunda Batalha:	6
2.2.4 Ranking Atualizado:	7
2.2.5 Batalha Geral:	8
2.2.6 Resultado Batalha Geral:	9
3. CONCLUSÃO	10
3.1 Análise dos resultados	11

1. INTRODUÇÃO

1.1 Propósito da Aplicação

O Aplicativo RPG ETEC é um sistema desenvolvido para a disciplina PAM II que oferece diversas funcionalidades, sendo o foco principal desta etapa a disputa entre os usuários com seus personagens. O trabalho concentra-se especificamente na implementação dos serviços, modelos, ICommand e métodos assíncronos para orquestrar e executar três tipos de embates na API: disputa com arma, disputa com habilidades e disputa geral. O sucesso desta implementação confirma a correta integração e comunicação entre o frontend mobile e o backend da aplicação.

2. REALIZAÇÃO DA IMPLEMENTAÇÃO E TESTES

2.1 Detalhamento da Implementação

Para que as disputas pudessem acontecer no aplicativo, o projeto foi organizado em camadas, garantindo que o código ficasse limpo e fácil de manter. A implementação seguiu os passos abaixo:

- **Preparação dos Dados:**

Criamos estruturas de dados (as classes Habilidade e PersonagemHabilidade) para que o aplicativo pudesse entender e guardar informações sobre as habilidades de cada personagem e como elas se relacionavam entre si.

- **Comunicação com o Servidor (API):**

Foram criadas duas áreas de serviço. Uma delas, o DisputaService, ficou responsável por enviar os pedidos de batalha (com arma, com habilidade e geral) para o sistema que calcula o resultado (a API).

A outra, o PersonagemHabilidadeService, foi criada para buscar, na API, quais habilidades um personagem específico possui.

- **Lógica de Ações e Comandos:**

Dentro da tela de disputas (o ViewModel), foram adicionadas as regras de negócio para que a luta pudesse ocorrer.

Implementamos os comandos de ação (os ICommands), que são o que fazem os botões da tela funcionar. Cada comando (Disputa com Arma, Habilidade e Geral) está ligado a um método que prepara os dados da luta e envia a requisição ao servidor.

Também ajustamos a seleção de personagens: ao escolher um atacante, o sistema agora busca e carrega automaticamente a lista de habilidades que ele pode usar na batalha.

- **Configuração da Tela (View):**

A tela de Disputas (ListagemView) foi atualizada para acomodar as novas funções.

Foram inseridos os botões para cada tipo de disputa (Arma, Habilidade e Geral).

Adicionamos um seletor (Picker) para que o usuário pudesse escolher a habilidade que o atacante irá usar no combate, antes de clicar no botão de Disputa com Habilidade.

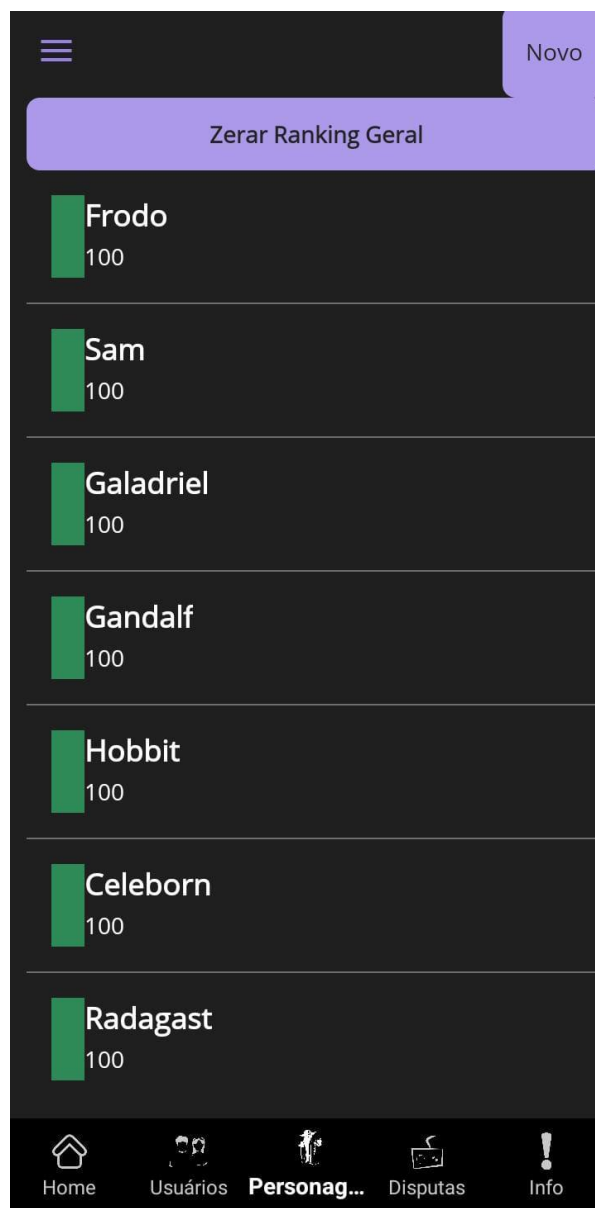
2.2 Passo a passo da testagem da funcionalidade de Disputa

Para validar que os comandos de ação e os serviços de disputa estavam funcionando corretamente, foram realizados testes práticos no aplicativo. Os resultados mostram o impacto imediato das batalhas no ranking dos personagens:

2.2.1 Estado Inicial:

O ranking mostra todos os personagens com o HP no máximo em 100 pontos.

Figura 1 – Ranking Zerado

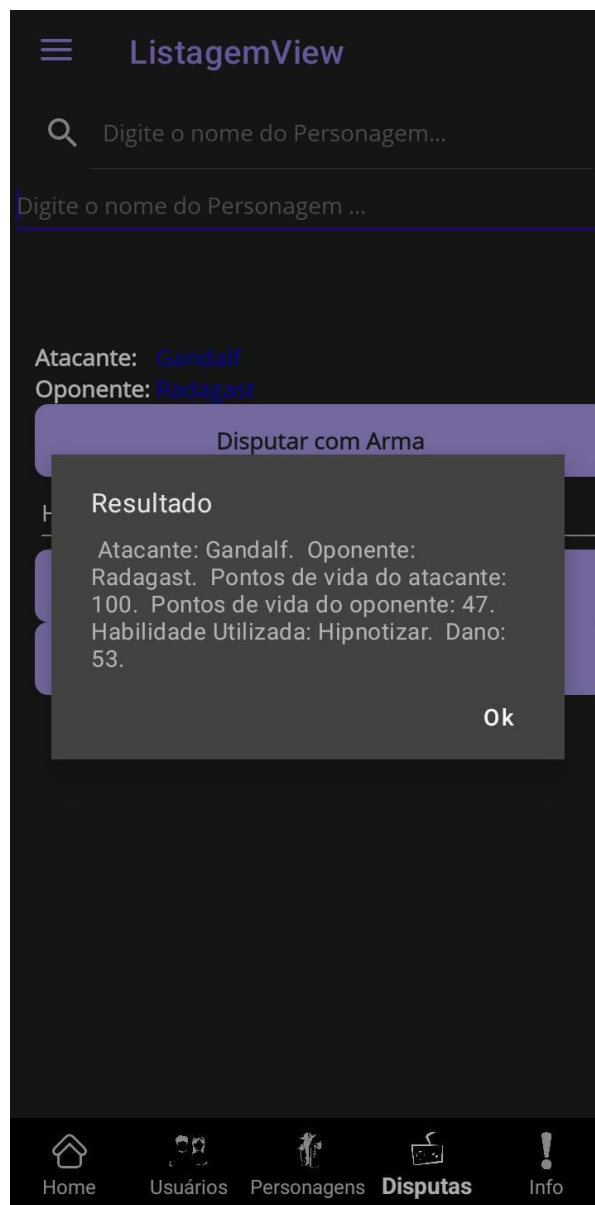


Fonte: Página Personagens

2.2.2 Batalha entre Gandalf e Radagast:

Uma batalha entre Gandalf como atacante e Radagast como oponente foi feita para comprovar se o método DisputaViewModel funcionava e se poderia processar o combate que utiliza uma habilidade.

Figura 2 – Primeira Batalha

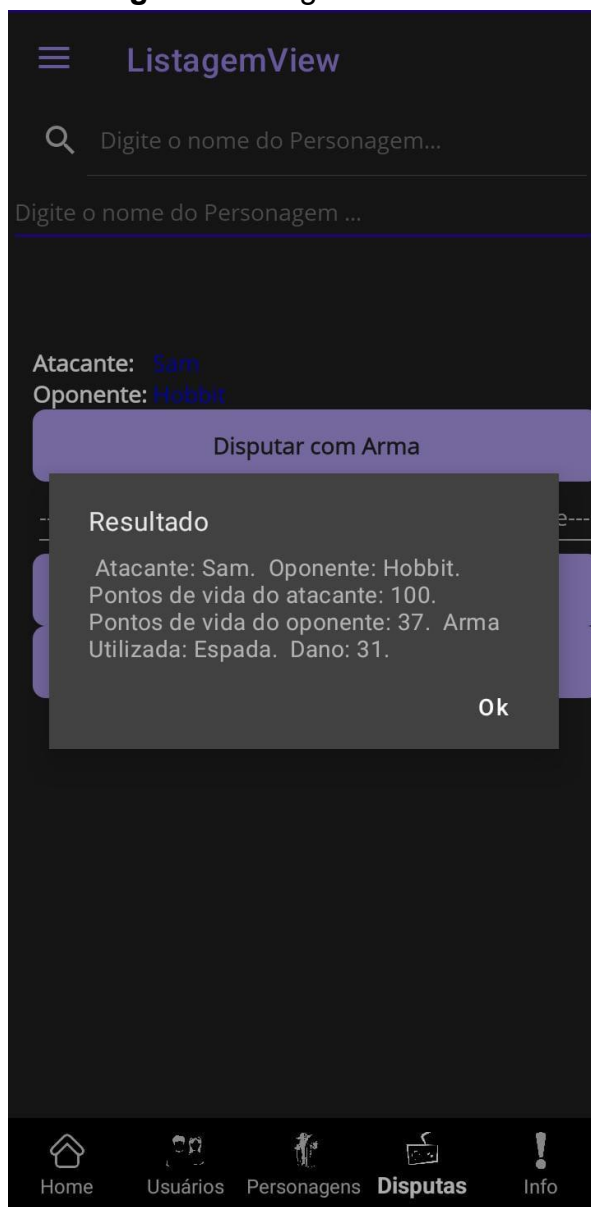


Fonte: Página Disputas

2.2.3 Segunda Batalha:

Outra batalha foi travada entre o atacante Sam e o oponente Hobbit.

Figura 3 – Segunda Batalha

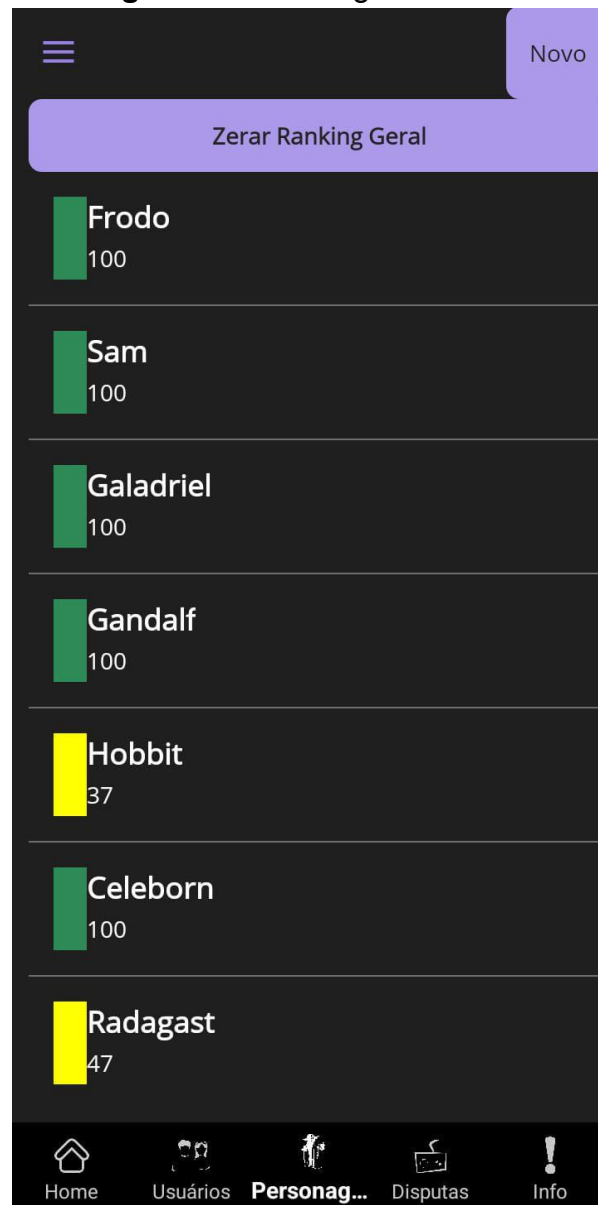


Fonte: Página Disputas

2.2.4 Ranking Atualizado:

O ranking foi atualizado após a segunda batalha entre Sam e Hobbit e isso evidencia que a backend ou seja a API está se conectando corretamente o frontend no mobile.

Figura 4 – Ranking Atualizado

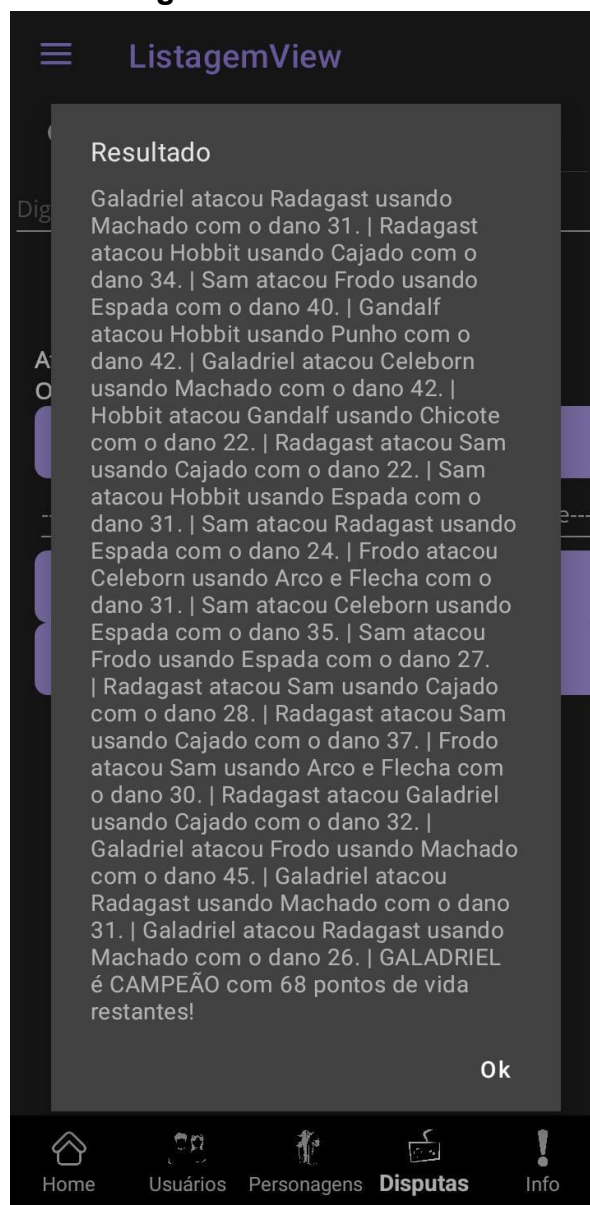


Fonte: Página Personagens

2.2.5 Batalha Geral:

A execução da disputa geral evidência que o ICommand está funcionando corretamente e que a lógica de buscar uma grande quantidade de dados e processá-las funciona corretamente.

Figura 5 – Batalha Geral



Fonte: Página Disputas

2.2.6 Resultado Batalha Geral:

O Ranking ter sido atualizado após a batalha geral mostra novamente que o backend está corretamente conectado ao frontend e que o sistema está atualizando os dados que ocorreram na disputa geral

Figura 6 – Ranking disputa geral

		Novo
Zerar Ranking Geral		
	Frodo	-12
	Sam	-17
	Galadriel	68
	Gandalf	-11
	Hobbit	-7
	Celeborn	-8
	Radagast	-12

Fonte: Página Personagens

3. CONCLUSÃO

A fase de criação e testes das funções de batalha foi um sucesso, atingindo o principal objetivo do projeto.

3.1 Análise dos resultados

Os testes mostraram que o sistema está funcionando perfeitamente. Colocamos os botões de ação (comandos) para as lutas com arma, habilidade e em grupo, e eles rodaram como deveriam. O ponto mais importante é que a comunicação entre o aplicativo e o servidor (API) está ótima. O servidor calculou o dano certinho em todas as brigas. O *ranking* com os pontos de vida dos personagens foi atualizado de forma correta depois de cada batalha, provando que o sistema inteiro está integrado e funcionando bem.

Link do Github:

<https://github.com/Thabata356/AppRpgEtecAzure.git>