

CENTRO UNIVERSITÁRIO UNIRUY WYDEN
CIÊNCIA DA COMPUTAÇÃO

LUAN MASAO HIRAHATA

PROJETO JAVA
ARMASTATS

SALVADOR
2025

LUAN MASAO HIRAHATA

**PROJETO JAVA
ARMASTATS**

Trabalho apresentado ao curso de Ciência da Computação, Uniruy Wyden, para obtenção de nota do projeto Web Java.

Orientador: Prof. Heleno Cardoso

**SALVADOR
2025**

RESUMO

O seguinte trabalho é um projeto web escrito em Java, envolvendo estatísticas do jogo ArmA 3. Utilizando os frameworks Spring Boot e Tailwind, o projeto se resume em retirar informações do servidor do game, para que sejam exibidas na aplicação. Assim sendo, usuários do servidor terão acesso as suas estatísticas em partidas jogadas, como: eliminações, mortes e tipos de eliminação. Para armazenar os dados, o banco de dados MySQL. O trabalho visa resolver um problema real, envolvendo uma comunidade online do simulador militar.

SUMÁRIO

1 INTRODUÇÃO	5
2 OBJETIVOS DA APLICAÇÃO	5
3 FUNÇÕES/LISTA DE EVENTOS (FUNCIONALIDADES) – RF / RNF	5
3.1 REQUISITOS FUNCIONAIS (RF)	6
3.2 REQUISITOS NÃO FUNCIONAIS (RNF).....	6
4 ESPECIFICAÇÃO DE PROGRAMAS.....	6
4.1 LAYOUT DA TELA.....	6
4.2 REGRAS DE NEGÓCIO	7
4.3 ENTIDADES ENVOLVIDAS	7
4.4 TABELAS	7
5 DER OU DIAGRAMA DE CLASSE.....	9
5.1 ARTEFATO GRÁFICO.....	9
5.2 DICIONÁRIO DE DADOS	9
6 APLICAÇÃO WEB	12
6.1 DASHBOARD/MENU	12
6.2 TELAS FUNCIONAIS.....	12
6.3 TELAS DE DIÁLOGO.....	13
6.4 ESTATÍSTICAS	13

1 INTRODUÇÃO

O projeto trata-se de uma aplicação web desenvolvida para extrair informações de servidores dedicados do jogo ArmA 3. Tais informações são logs feitos pelo servidor, salvando dados dos jogadores, como: eliminações, mortes e tempo jogado. Após extrair os dados, a aplicação será responsável por exibi-los em uma dashboard pública, reunindo rankings e informações individuais de cada jogador. Para administrar a banco de dados de jogadores e partidas, a aplicação possuí uma dashboard customizada para o administrador. Esta aplicação visa ajudar as comunidades envolvidas neste jogo, mantendo um histórico de partidas e estimulando a participação de seus membros em diferentes eventos.

Utilizando o Spring Boot para o backend, a aplicação foi construída a com o Thymeleaf, incluso no framework. Além disso, o Tailwind foi escolhido como o estilizador CSS da aplicação. Tais escolhas foram justificadas pela simplicidade do projeto. O banco de dados escolhido foi o MySQL, rodando em um ambiente cloud (Railway) assim como a aplicação em si. Não menos importante, a versão Java da aplicação trata-se do SDK 24.

2 OBJETIVOS DA APLICAÇÃO

- Ler os arquivos “server.log” gerados pelo servidor dedicado.
- Exibir as informações coletadas do arquivo numa dashboard.
- Ranking de jogadores baseado em eliminações.
- Login de administrador para o upload de arquivos.
- Possibilidade de deletar dados de partidas ou jogadores.

3 FUNÇÕES/LISTA DE EVENTOS (FUNCIONALIDADES) – RF / RNF

Os Requisitos Funcionais e Não Funcionais da aplicação também foram discutidos durante o seu desenvolvimento.

3.1 REQUISITOS FUNCIONAIS (RF)

- **RF1:** O sistema irá ler os arquivos “server.log” e fazer o parsing dos dados.
- **RF2:** O sistema exibirá os dados numa dashboard.
- **RF3:** O sistema fará um ranking de jogadores baseado em suas eliminações.

3.2 REQUISITOS NÃO FUNCIONAIS (RNF)

- **RNF1:** A aplicação deverá consumir o mínimo possível de bandwidth afim de evitar custos adicionais na nuvem.

4 ESPECIFICAÇÃO DE PROGRAMAS

A seguir, serão introduzidos os layouts das telas da aplicação, assim como as regras de negócio e entidades envolvidas.

4.1 LAYOUT DA TELA

A principal tela da aplicação, que é acessada pela maioria dos usuários, é dividida em elementos retangulares que contém rankings de cada estatística do jogador (Eliminações, mortes etc.). Todas as telas possuem uma barra de navegação na parte superior do site, possibilitando o acesso a outras páginas.

O painel do administrador inclui suas funções e informações em elementos retangulares, assim como a página principal, porém exibidos de forma horizontal para se adequar aos gráficos e botões.

Missões e jogadores são listados em tabelas sem linhas verticais, apenas com linhas horizontais. Botões para detalhes e delete são destacados em azul e vermelho.

A tela de login do administrador é acessada na barra de navegação, com um design tradicional: uma página inteira para o acesso, com campos de nome usuário e senha inseridos numa caixa retangular.

4.2 REGRAS DE NEGÓCIO

A seguir, serão descritas as regras de negócio da aplicação, consideradas essenciais para o bom funcionamento dela, considerando as especificações e peculiaridades do negócio.

Considerando que um jogador pode trocar seu nome no game, é necessário que a aplicação não crie duas entradas de nome para o mesmo Steam ID (Identificador único de cada jogador dentro do game). Quando um jogador logar com um novo nome, o antigo será substituído pelo mais atual.

Ao ler os dados de um arquivo server.log, um jogador pode se conectar e desconectar diversas vezes da partida. Deve ser considerado somente o tempo que ele esteve presente no servidor.

Para checar se a presença do jogador foi válida numa missão, ele deverá jogar 80% do tempo total de uma partida.

O primeiro horário reportado no arquivo log, assim como o último, não devem ser considerados como final e início de uma missão ou partida. Utilizar as informações “Game started” e “Game finished” para o cálculo.

4.3 ENTIDADES ENVOLVIDAS

- **Player:** Representa as informações do jogador que participou de uma missão, possuindo os atributos: id, steamId e name.
- **GameSession:** Representa uma única missão ou partida jogada, com os atributos: id, missionName, island, startTime, endTime e durationSeconds.
- **PlayerSessionStats:** id, player, gameSession, killsTotal, deaths, killsInfantry, killsArmor, killsAir, killsPlayers, killsSoft e accumulatedSecondsPlayed.

4.4 TABELAS

Abaixo estão as tabelas do banco de dados correspondentes às entidades JPA, com seus campos, tipos de dados SQL inferidos e restrições:

Tabela: player (Mapeada da entidade Player)

- id: BIGINT (Chave Primária, Auto-incremento)
- steam_id: VARCHAR(255) (Único, Não Nulo)
- name: VARCHAR(255) (Não Nulo)

Tabela: game_session (Mapeada da entidade GameSession)

- id: BIGINT (Chave Primária, Auto-incremento)
- mission_name: VARCHAR(255)
- island: VARCHAR(255)
- game_type: VARCHAR(255)
- start_time: DATETIME (ou TIMESTAMP)
- end_time: DATETIME (ou TIMESTAMP)
- duration_seconds: DOUBLE

Tabela: player_session_stats (Mapeada da entidade PlayerSessionStats)

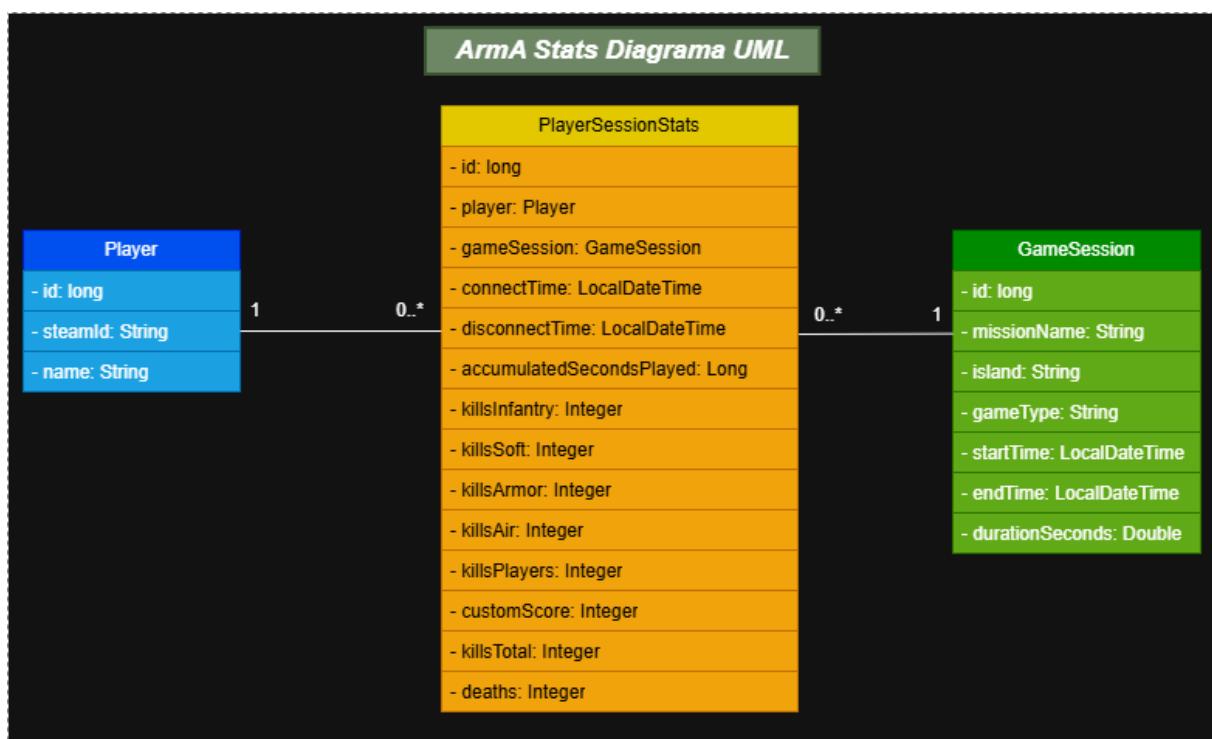
- id: BIGINT (Chave Primária, Auto-incremento)
- player_id: BIGINT (Chave Estrangeira para player.id, Não Nulo)
- game_session_id: BIGINT (Chave Estrangeira para game_session.id, Não Nulo)
- connect_time: DATETIME (ou TIMESTAMP)
- disconnect_time: DATETIME (ou TIMESTAMP)
- accumulated_seconds_played: BIGINT
- kills_infantry: INT
- kills_soft: INT
- kills_armor: INT
- kills_air: INT
- kills_players: INT
- custom_score: INT
- kills_total: INT
- deaths: INT

5 DER OU DIAGRAMA DE CLASSE

A seguir, serão apresentados o Diagrama de Classe UML do projeto e o Dicionário de Dados (DD).

5.1 ARTEFATO GRÁFICO

Figura 1 - Diagrama UML



5.2 DICIONÁRIO DE DADOS

A aplicação possuí três tabelas em seu Banco de Dados SQL: *player*, *game_session* e *player_session_stats*.

Tabela 1 - Tabela player

Campo	Tipo de Dado	PK/FK	Nulo? (S/N)	Descrição
id	BIGINT	PK	N	Identificador numérico único para cada jogador.
steam_id	VARCHAR(255)		N	Identificador único da Steam

name	VARCHAR(255)	N	(SteamID64) do jogador. Usado para login e rastreamento. O nome (nickname) do jogador.
------	--------------	---	---

Tabela 2 - Tabela game_session

Campo	Tipo de dado	PK/FK	Nulo? (S/N)	Descrição
id	BIGINT	PK	N	Identificador numérico único para cada sessão de jogo. O nome do arquivo da missão (ex: "MyMission.Altis").
mission_name	VARCHAR(255)		S	O mapa/ilha onde a missão ocorreu (ex: "Altis", "Tanoa").
island	VARCHAR(255)		S	O tipo de jogo (ex: "PvE", "PvP", "Coop").
game_type	VARCHAR(255)		S	Data e hora exatas de quando a sessão de jogo (missão) começou no servidor.
start_time	DATETIME		S	Data e hora exatas de quando a sessão de jogo (missão) terminou.
end_time	DATETIME		S	Duração total da missão em segundos (calculado como endTime - startTime).
duration_seconds	DOUBLE		S	

Tabela 3 - player_session_stats

Campo	Tipo de Dado	PK/FK	Nulo?	Descrição
id	BIGINT	PK	N	Identificador numérico único para este registro de estatística.
Player_id	BIGINT	FK	N	Referência à chave primária (id) da tabela

Game_session_Id	BIGINT	FK	N	player. Indica qual jogador. Referência à chave primária (id) da tabela game_session. Indica em qual sessão.
Connect_time	DATETIME		S	Data e hora em que o jogador conectou nesta sessão.
Disconnect_time	DATETIME		S	Data e hora em que o jogador desconectou desta sessão.
Accumulated_seconds_played	BIGINT		S	Tempo total em segundos que o jogador esteve presente na sessão.
Kills_infantry	INT		S	Número de abates contra unidades de infantaria (IA).
Kills_soft	INT		S	Número de abates contra veículos leves (ex: carros, caminhões).
Kills_armor	INT		S	Número de abates contra veículos blindados (ex: tanques, APCs).
Kills_air	INT		S	Número de abates contra veículos aéreos (ex: helicópteros, jatos).
Kills_players	INT		S	Número de abates contra outros jogadores (humanos).
Custom_score	INT		S	Pontuação personalizada (se aplicável, para objetivos, etc.).
Kills_total	INT		S	Soma total de todos os tipos de abates.
deaths	INT		S	Número total de vezes que

o jogador
morreu.

6 APLICAÇÃO WEB

A seguir, serão apresentadas as principais funcionalidades da aplicação Arma Stats. Construída com Tailwind e Spring Boot, ela visa ser rápida e de fácil entendimento pelo usuário, evitando excessos em seu design.

6.1 DASHBOARD/MENU

A tela principal da aplicação é o dashboard de rankings. Nele serão exibidos os rankings de eliminações, eliminações específicas (Infantaria, veículos terrestres ou aéreos), missões jogadas no total, proporção de eliminações/mortes e eliminações de jogadores contra jogadores. O dashboard de rankings é dividido em listas retangulares, com número de colocação, nome do jogador e kills, para cada tipo de ranking.

6.2 TELAS FUNCIONAIS

Todas as telas da aplicação possuem uma barra de navegação na parte superior, contendo as principais páginas: Rankings, Players, Admin Login e Admin Panel (Admin Logout para desconectar como Administrador).

A tela players lista os jogadores, com as colunas Player Name e Steam ID. Ao lado de cada jogador listado, há uma opção para ver mais detalhes sobre o jogador ou deletá-lo do banco de dados. Ao acessar mais detalhes sobre jogador, aparecerá uma lista com as missões jogadas por ele, assim como número de eliminações, eliminações específicas e mortes.

Para acessar o dashboard de administrador, existe um elemento retangular centralizado, contendo o nome do usuário acima de senha para acesso, assim como o botão de login abaixo de ambos.

No painel do administrador, os elementos são divididos de forma semelhante ao painel de rankings, com elementos retangulares para cada função. Existe um botão para escolher e outro para enviar um arquivo log. Também existe um botão para visualizar as missões jogadas. Abaixo destas opções, existem gráficos que ilustram o

número de jogadores, eliminações e mortes por missão. Serão comparadas somente as últimas 15 missões em cada gráfico.

Ao clicar no botão para ver a lista de missões, o administrador verá uma tela semelhante a lista de players, com o nome da missão, data e hora de registro e ações para ver presença ou deletar missão. Ao clicar para ver a presença na missão, aparecerá uma lista de jogadores, com a porcentagem de presença e uma coluna para indicar presença ou falta de cada jogador.

6.3 TELAS DE DIÁLOGO

A aplicação utiliza telas de diálogo nativas do navegador para alertar o administrador sobre o *delete* de um usuário ou missão, utilizando JavaScript sem HTML ou CSS dedicados. Quando não há qualquer estatística carregada no servidor, a mensagem “No stats available”.

Para tratar de erros no upload de arquivos *.log*, as mensagens de erro serão exibidas acima do título “Admin Panel” na página do administrador, assim como mensagens de êxito.

6.4 ESTATÍSTICAS

Figura 2 - Estatísticas Administrador

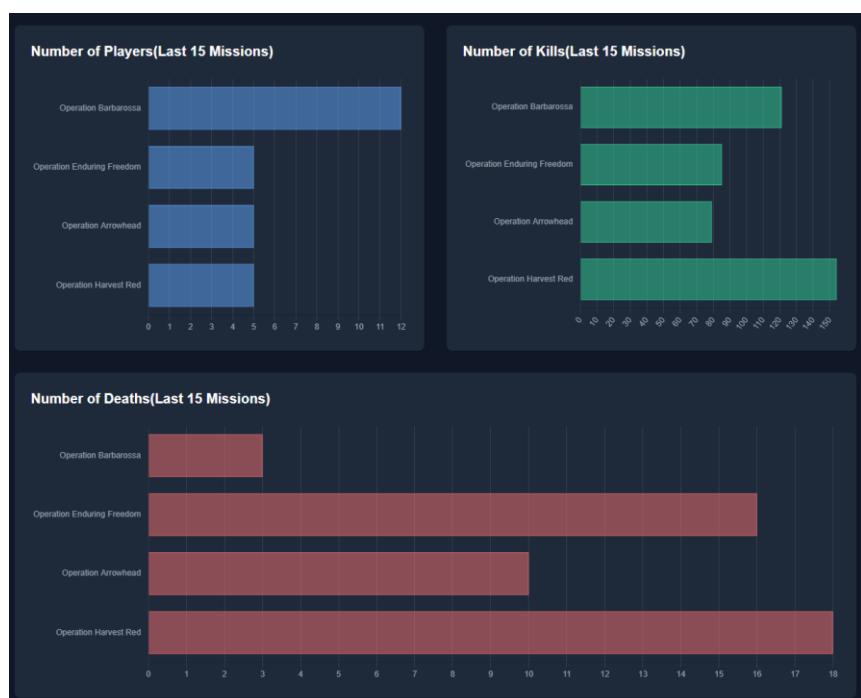


Figura 3 - Dashboard de Rankings

Best K/D Ratio		
#	PLAYER	K/D
1	Taurax	29,00
2	Aikan	14,00
3	YourDoctor	10,50
4	Hunter	0,00
5	Luanmh	0,00