

CURSO DE CIÊNCIA DA COMPUTAÇÃO

Gabriel Kirst da Rocha

**DEFINIÇÃO DE MÉTRICAS PARA ANÁLISE DE DADOS  
EM JOGOS DO TIPO MMORPG**

Santa Cruz do Sul

2016

Gabriel Kirst da Rocha

**DEFINIÇÃO DE MÉTRICAS PARA ANÁLISE DE DADOS  
EM JOGOS DO TIPO MMORPG**

Trabalho de Conclusão apresentado ao curso de Ciência da Computação da Universidade de Santa Cruz do Sul para obtenção do título de Bacharel em Ciência da Computação.

Orientador Prof. Msc. Eduardo Kroth

Santa Cruz do Sul  
2016

*À minha noiva e aos meus pais,  
que sempre me apoiaram.*

## **AGRADECIMENTOS**

Gostaria de agradecer primeiramente aos meus pais, Nelson e Síría que sempre me apoiaram em minha vida acadêmica desde minha adolescência, e que me transmitiram todos os ensinamentos e valores que levo.

Agradeço à minha noiva Letícia por ter sempre me apoiado e incentivado nesta trajetória. Por ter me ajudado em momentos ruins e sempre me dado forças, indicando o melhor caminho a ser seguido. Muito obrigado.

Deixo também um agradecimento ao meu orientador, Prof. Msc. Eduardo Kroth, pelo apoio tanto na vida acadêmica quanto na profissional, e por me indicar o norte em momentos de dúvidas no desenvolvimento do trabalho.

Por fim, agradeço aos professores que participaram comigo desta trajetória.

## RESUMO

O mercado de jogos computacionais está em um patamar nunca antes visto, com uma estimativa de movimentar US\$99,6 bilhões em 2016. Dentre as diversas categorias de jogos, os MMORPG (Massive Multiplayer Online Role-Playing Game) atraem centenas de milhões de jogadores, que interagem através de seus personagens simultaneamente em um mesmo universo. Como consequência ao grande número de jogadores, uma massa inimaginável de dados é gerada, contendo as ações de cada um dos jogadores dentro do jogo. Devido ao tamanho destes dados e da quantidade de diferentes atributos gravados, extrair informações úteis pode ser um desafio. Estas características dos logs destes jogos tornam a mineração de dados como uma opção viável para extração de informações, pois esta já está preparada para lidar com grandes volumes de dados, e possui algoritmos próprios para descoberta de conhecimentos sobre os dados. O trabalho de conclusão proposto tem por objetivo a identificação de quais características presentes em logs de jogos do tipo MMORPG podem ser utilizadas para análise e classificação de perfis de jogadores. Para tanto, será feito o uso de um algoritmo de mineração de dados, verificando quais características possuem resultados mais satisfatórios para categorização de jogadores. Também busca identificar características em comum entre logs de diferentes jogos, visando manter o resultado genérico.

## **ABSTRACT**

*The market for computer games is in a level never seen before, estimating to move US\$99.6 billion in 2016. Among the various categories of games, MMORPGs (Massive Multiplayer Online Role-Playing Game) attract hundreds of millions of players, who interact through their characters simultaneously in the same universe. As a result of the large number of players, an unimaginable mass of data is generated, containing the actions of each player in the game. Due to the size of the data, and the number of distinct attributes recorded, extract useful information can be challenging. These characteristics of the logs of those games make data mining as a viable option to extract information, as this is already prepared to handle large volumes of data, and has its own algorithms for knowledge discovery in datasets. The proposed final paper aims to identify which characteristics in MMORPG's game logs can be used for analysis and classification of player profiles. For this, a data mining algorithm will be used, checking which characteristics bring better results to players' categorization. The paper also try to identify characteristics in common among different game logs, looking for maintain generic results.*

## LISTA DE FIGURAS

Figura 1 – Processo de Knowledge Discovery in Database.....	17
Figura 2 – Fases do pré-processamento de dados. ....	18
Figura 3 – Utilizando o algoritmo <i>K-means</i> para encontrar três figuras nos dados de exemplo. ....	23
Figura 4 – O PDP-I e o jogo Spacewar!. ....	24
Figura 5 – Pong e VCS, respectivamente, lançados pela Atari. ....	25
Figura 6 – Nintendo NES e seu jogo Super Mario Bros. ....	26
Figura 7 – World of Warcraft, lançado em 2004 pela Blizzard.....	27
Figura 8 – Relação entre categorias e o ambiente do jogo. ....	31
Figura 9 – Interação entre jogador, jogo e desenvolvedores. ....	33
Figura 10 – Etapas no processo de Data Mining utilizado. ....	34
Figura 11 – Árvore de decisão final gerada pelo trabalho. ....	35
Figura 12 – Exemplo de tabela extraída, com experiência x métrica. ....	37
Figura 13 – Estatística dos dados utilizados. ....	38
Figura 14 – Detecção de bots considerando diferentes estilos de jogo. ....	42
Figura 15 – Etapas do processo, da captura dos logs à análise dos resultados. ....	47
Figura 16 – Estrutura das informações de jogador no MMORPG Tibia. ....	49
Figura 17 – Representação da extração das informações de jogador no Talend. ....	51
Figura 18 – Estrutura do arquivo com informações do jogado unificadas. ....	53
Figura 19 – Tratamento dos arquivos de “fotografia” da base de dados. ....	53
Figura 20 – Arquivo de saída pronto para ser aplicado ao <i>k-means</i> . ....	54
Figura 21 – Configuração do <i>k-means</i> dentro do WEKA 3.6.....	55
Figura 22 – <i>Clusters</i> criados pelo algoritmo <i>k-means</i> . ....	56
Figura 23 – Gráficos de (1) variação de ganho de experiência por jogador e (2) variação de ganho de dinheiro por jogador. ....	58
Figura 24 – Gráfico de variação de nível (y) por jogador. ....	59
Figura 25 – Média de experiência (x) por variação de nível (y), com nível do jogador na cor. ....	60
Figura 26 – Outros testes realizados.....	61
Figura 27 – <i>Clusters</i> obtidos com nova estrutura de dados. ....	62

Figura 28 – Gráfico de jogador (x) por nível (y).....	63
Figura 29 – Base de dados gerada pelo Zereal Agent-based Simulator. ....	65
Figura 30 – Etapas de (1) agrupamento de ações por jogador e tipo de ação e (2) agrupamento de ações por jogador.....	67
Figura 31 – Arquivo pronto para ser aplicado ao algoritmo <i>k-means</i> . ....	68
Figura 32 – Configuração do <i>k-means</i> dentro do WEKA 3.6.....	69
Figura 33 – <i>Clusters</i> criados em teste sobre base de dados dinâmicos. ....	70
Figura 34 – Atribuição de <i>clusters</i> a tipos de jogador.....	71
Figura 35 – <i>Clusters</i> resultantes de análise desconsiderando ação <i>leaveworld</i> . ....	71
Figura 36 – Ações de movimento (x) por ações de ataque (y). ....	72
Figura 37 – Atribuição de <i>clusters</i> a tipos de jogador após ajuste na análise. ....	73
Figura 38 – Funcionamento básico do Talend. ....	74
Figura 39 – Imagem da tela inicial do Talend.....	75
Figura 40 – Imagem da tela inicial do WEKA, em sua função <i>Explorer</i> . ....	76

## LISTA DE TABELAS

Tabela 1 – Algoritmo <i>K-means</i> básico.....	22
Tabela 2 – Exemplo de registro de ações dos jogadores.....	39
Tabela 3 – Comparativo entre trabalhos relacionados. ....	43
Tabela 4 – Regra para aplicação de coeficiente sobre experiência ganha. ....	62

## LISTA DE ABREVIATURAS

CSV	–	Comma-separated values
ETL	–	Extraction, Transformation, Loading
IA	–	Inteligência Artificial
KDD	–	Knowlegde Discovery in Database
MIT	–	Massachusetts Institute of Technology
MMORPG	–	Massive Multiplayer Online Role-Playing Game
MUD	–	Multi-User Dungeon
NES	–	Nintendo Entertainment System
NPC	–	Non-player character
PK	–	Player-killer
RPG	–	Role-Playing Game
RUnEA	–	Robust Unique Effect Analysis
SQL	–	Structured Query Language
UV	–	Unidade de Valor
VCS	–	Video Computer System

## SUMÁRIO

<b>1.</b>	<b>INTRODUÇÃO .....</b>	<b>12</b>
1.1.	Objetivos do trabalho .....	13
1.2.	Organização do texto .....	14
<b>2.</b>	<b>DATA MINING .....</b>	<b>15</b>
2.1.	Knowledge Discovery in Database .....	16
2.2.	Extração e transformação de dados .....	17
2.2.1.	Considerações do autor .....	19
2.3.	Tarefas em mineração de dados: .....	20
2.3.1.	Tarefas preditivas .....	20
2.3.2.	Tarefas descritivas .....	21
2.4.	Análise de agrupamento k-means.....	21
<b>3.</b>	<b>JOGOS COMPUTACIONAIS .....</b>	<b>24</b>
3.1.	Conceitos em MMORPG.....	26
3.2.	Estruturas de informações nos logs.....	28
3.3.	Categorização de Perfis de Jogadores .....	30
3.4.	Considerações do autor .....	32
<b>4.</b>	<b>TRABALHOS RELACIONADOS .....</b>	<b>33</b>
4.1.	Using Data Mining to Model Player Experience .....	33
4.2.	Detecting Player Goals from Game Log Files .....	34
4.3.	Mining Rules from Player Experience and Activity Data .....	36
4.4.	Data Mining and Machine Learning with Computer Game Logs .....	37
4.5.	A Data Mining Approach to Strategy Prediction .....	39
4.6.	Game Bot Detection Approach Based on Behavior Analysis and Consideration of Various Play Styles .....	41
4.7.	Conclusão sobre os trabalhos relacionados .....	42

<b>5.</b>	<b>SOLUÇÃO DESENVOLVIDA.....</b>	<b>46</b>
<b>5.1.</b>	<b>Análise sobre base de dados estática .....</b>	<b>47</b>
<b>5.1.1.</b>	<b>Extração e transformação das informações .....</b>	<b>50</b>
<b>5.1.2.</b>	<b>Aplicação do algoritmo k-means.....</b>	<b>54</b>
<b>5.1.3.</b>	<b>Avaliação dos resultados.....</b>	<b>64</b>
<b>5.2.</b>	<b>Análise sobre base de dados dinâmica.....</b>	<b>65</b>
<b>5.2.1.</b>	<b>Extração e transformação das informações .....</b>	<b>67</b>
<b>5.2.2.</b>	<b>Aplicação do algoritmo k-means.....</b>	<b>69</b>
<b>5.2.3.</b>	<b>Avaliação dos resultados.....</b>	<b>73</b>
<b>5.3.</b>	<b>Ferramentas utilizadas .....</b>	<b>74</b>
<b>5.3.1.</b>	<b>Talend Open Studio for Data Integration .....</b>	<b>74</b>
<b>5.3.2.</b>	<b>WEKA 3.....</b>	<b>75</b>
<b>6.</b>	<b>CONCLUSÕES .....</b>	<b>77</b>
	<b>REFERÊNCIAS .....</b>	<b>79</b>

## 1. INTRODUÇÃO

Com a constante evolução da tecnologia e o surgimento de sistemas que vem em auxílio às mais variadas áreas de negócio, as empresas passaram a ter seus dados registrados em gigantescas bases de dados. A extração de informações úteis, tais como padrões nos dados, possibilidade de otimização em processos ou prospecção de novos clientes, podem ser desafiadoras, já que algoritmos tradicionais de extrações de dados não são eficazes sobre este tipo de informação.

Amorim (2006) comenta que atualmente as organizações mostram que estão avançadas nesta obtenção diária de informações, porém a grande maioria não utiliza os meios corretos para extrair informação dessas imensas bases de dados. Amorim complementa dizendo que essa forma inadequada de extrair informação desses dados prejudica suas atividades, sendo que com grandes investimentos nessa área de armazenamento de dados faz-se necessário o estudo da forma correta da busca de informação.

Visando a solução destes problemas, surgiu na década de 80 a Mineração de Dados, com o principal objetivo de extrair informações úteis destas grandes massas de dados (na ordem de gigabytes e milhões de registros), através da aplicação de algoritmos tradicionais combinados com métodos capazes de processar grandes quantidades de dados, além de metodologias de análise das informações.

Porém, a aplicação de mineração de dados não foca apenas no âmbito de negócios tradicionais, podendo ser utilizada nas mais diversas áreas de conhecimento, tais como pesquisas científicas, medicina, campanhas eleitorais, segurança, dentre outros.

Com a evolução e popularização de jogos computacionais e com dados destes jogos passando a ser salvos (especialmente em jogos que utilizam a Internet), a utilização de técnicas de mineração de dados para interpretar e extrair informações destes dados surgiu de forma natural. Dentre os objetivos possíveis destas análises, tendo como base os trabalhos de Weber (2011), DiCerbo (2012), Colton (2012), Wender (2007) e Chung (2013), é possível citar:

- Análise comportamental do jogador para auxílio no desenvolvimento de inteligências artificiais;
- Identificação de padrões dentro dos jogos;

- Análise da relação entre as características do jogo e o jogador;
- Análise comportamental do jogador;
- Detecção de condutas antiéticas de jogadores dentro do jogo, tais como uso de ferramentas auxiliares para trapacear;
- Extração de informações úteis para evoluções nos jogos.

A realização do trabalho aqui apresentado visa classificar perfis de jogadores dentro de jogos da categoria MMORPG (*Massive Multiplayer Online Role-Playing Game*) através de técnicas de identificação de padrões de ações utilizando técnicas de mineração de dados, também com o propósito de verificar o comportamento destes algoritmos frente a essas informações.

### **1.1. Objetivos do trabalho**

O objetivo geral deste trabalho de conclusão será a aplicação de mineração de dados sobre registros de eventos de jogos do tipo MMORPG, a fim de categorizar perfis de jogadores tendo como base seus padrões de comportamento.

Os objetivos específicos deste trabalho são:

- Realizar o pré-processamento dos logs dos jogos;
- Estudar e aplicar o algoritmo de mineração de dados *k-means*, utilizando a ferramenta WEKA;
- Verificar a capacidade do algoritmo de agrupar as informações nos perfis propostos;
- Identificar características em comum entre os logs e quais atributos devem ser considerados na análise.

No Trabalho de Conclusão I, a lista de objetivos específicos continha o seguinte tópico, porém, (1) devido à complexidade em obter as bases de dados, (2) devido ao tempo dedicado no tratamento de dados das duas bases de dados obtidos e (3) devido à satisfação nos resultados obtidos; não se teve tempo suficiente para aplicar as bases de dados a um segundo algoritmo.

*“Comparar o desempenho dos algoritmos de mineração de dados utilizados, quanto a qualidade de resultados e tempo de execução.”*

## **1.2. Organização do texto**

O texto do trabalho será dividido da seguinte forma: o capítulo dois aborda os conceitos e processos envolvidos na mineração de dados, assim como conceitua a técnica que foi utilizada no trabalho. O capítulo três traz a história de jogos computacionais, introduz os MMORPGs e fornece conceitos gerais sobre estes. Também fundamenta conceitos presentes dentro desta categoria de jogos. O capítulo quatro aborda os trabalhos relacionados pesquisados, trazendo também um comparativo entre eles. O capítulo cinco traz sobre o desenvolvimento da solução. Por fim, o capítulo seis explanará sobre considerações finais do trabalho.

## 2. DATA MINING

Com o avanço tecnológico na computação, existem hoje *hardwares* capazes de armazenar grandes quantidades de informações. Junto disto, com o crescimento do uso de sistemas computacionais por parte de organizações, foi natural o acúmulo de uma vasta quantidade de dados contendo a vida destas. Além do âmbito empresarial, cabe citar como exemplo pesquisas científicas, cujos resultados são cadeias imensas de dados, como, por exemplo, o projeto Genoma, que armazena milhares de *bytes* para cada uma das bilhões de bases genéticas.

Porém, todo este acúmulo de dados nos trouxe problema em relação à extração de informações úteis destas imensas massas de dados. As técnicas usuais de coletas de dados tornaram-se obsoletas, pois, além de não serem capazes de trabalhar com bases de dados destas magnitudes, não conseguem dar todas as informações desejadas sobre os dados gravados.

Na década de 80 surgiu a mineração de dados, do inglês *Data Mining*, como uma solução para estas questões, através da combinação de técnicas de análise de dados tradicionais com algoritmos desenvolvidos especificamente para terem capacidade de processar grandes volumes de dados, em um tempo hábil e obtendo informações satisfatórias. Segundo Fayyad (et. al., 1996), *Data Mining* é o processo não trivial de identificar, em dados, padrões válidos, novos, potencialmente úteis e ultimamente compreensíveis. Já Tan (2009) conceitua mineração de dados como um processo de descoberta automática de informações úteis em grandes depósitos de dados.

Segundo Fayyad (et. al., 1996), o modelo tradicional para transformação dos dados em informação (conhecimento) consiste em um processamento manual de todas essas informações por especialistas que, então, produzem relatórios que deverão ser analisados. Tratando-se de bases de informações de *giga*, ou até mesmo *terabytes*, o modelo tradicional torna-se inviável. Estas novas técnicas possibilitam a identificação, em meio de uma grande base de dados, de padrões que podem passar despercebidos. Além disto permitem, tendo como base os dados registrados, fazer análises preditivas de cenários futuros. Em síntese, a mineração de dados permite a extração de informações úteis, e não apenas de dados.

De acordo com Dantas (2008), a grande importância da utilização da mineração de dados se deve às grandes quantias de informações guardadas, onde esses arquivos possuem informações úteis, porém de difícil interpretação, auxiliando em previsões futuras, tornando então dados sem utilidades e desorganizados em grandes fontes de informação. Witten (et. al., 2005) Olson (et al., 2008) e Bramer (2007) apresentam algumas das áreas nas quais a Mineração de Dados pode ser aplicada, cabendo citar:

- Retenção de clientes: identificação de perfis para determinados produtos, venda cruzada;
- Bancos: identificar padrões para auxiliar no gerenciamento de relacionamento com o cliente;
- Cartão de Crédito: identificar segmentos de mercado, identificar padrões de rotatividade;
- Cobrança: detecção de fraudes;
- Telemarketing: acesso facilitado aos dados do cliente;
- Eleitoral: identificação de um perfil para possíveis votantes;
- Medicina: indicação de diagnósticos mais precisos;
- Tomada de Decisão: filtrar as informações relevantes, fornecer indicadores de probabilidade.

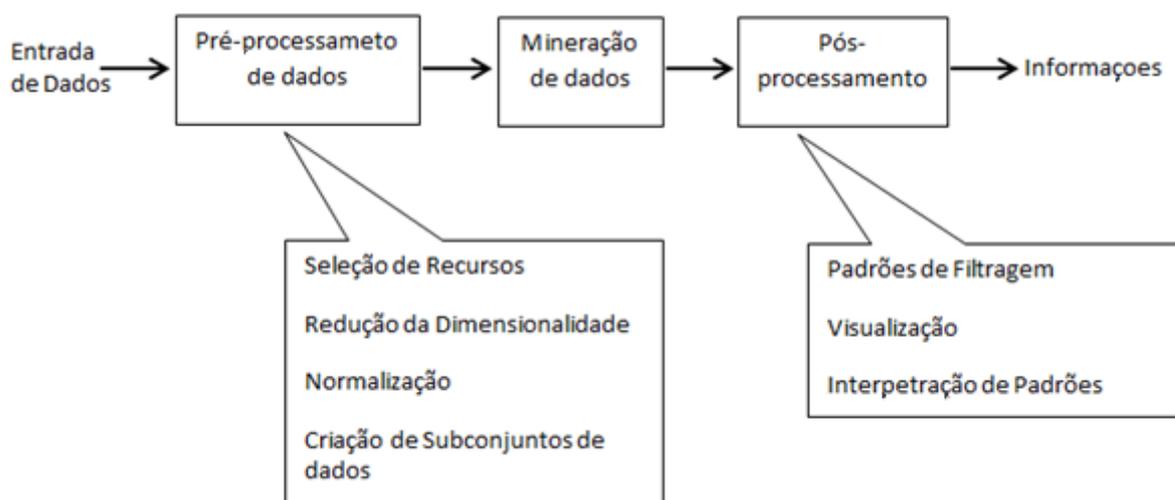
Segundo Tan (2009), quanto às tarefas incumbidas à mineração de dados, estas podem ser divididas em dois grupos principais: tarefas de previsão - tendo como objetivo prever o valor de um determinado atributo baseado em outro atributo - e tarefas descritivas - com o objetivo de identificar padrões (correlações, tendências, grupos, anomalias) que resumam os relacionamentos entre dados.

## **2.1. Knowledge Discovery in Database**

A definição do processo de *Knowledge Discovery in Database* ainda não é consenso. Para alguns autores, como por exemplo Han (2006), seu conceito é o mesmo de mineração de dados, enquanto para Tan (2009) e Fayyad (1996), a mineração de dados é apenas um dos processos que compõem o KDD. Porém, independente disto, é consenso que o processo deve ser dividido em etapas.

A Descoberta de Conhecimentos em Bases de Dados, segundo Tan (2009) é conceituada como o processo de extração de dados úteis sobre massas de dados.

**Figura 1 – Processo de Knowledge Discovery in Database.**



Fonte: TAN, 2009.

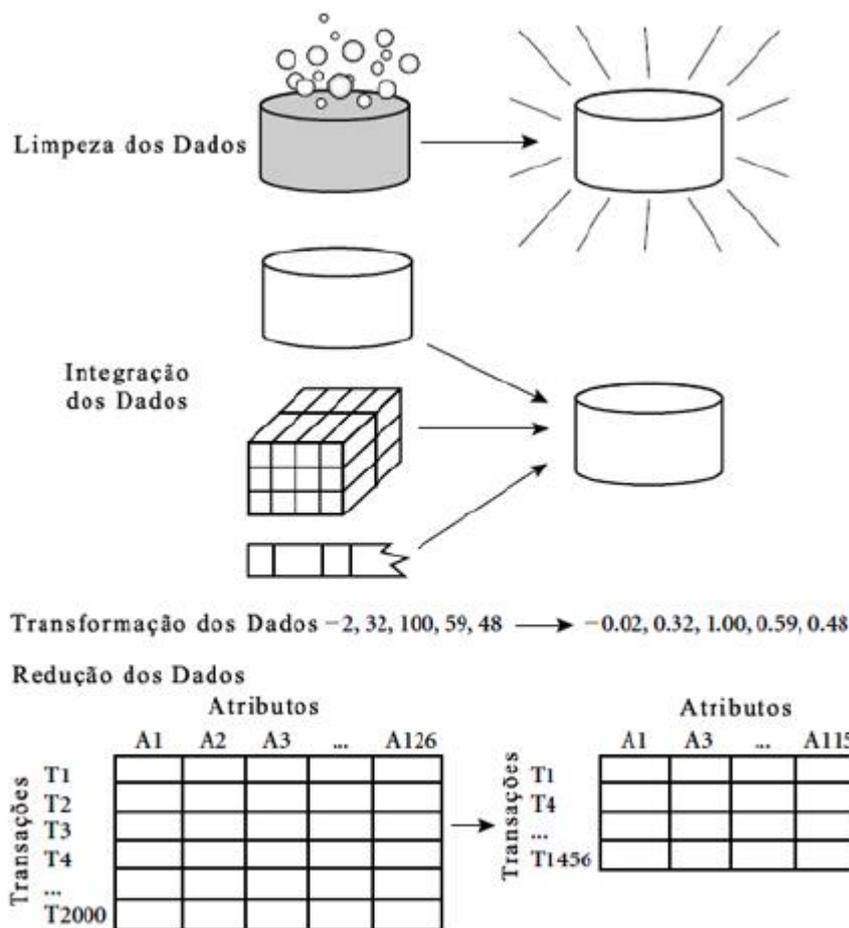
Existem diversas padronizações das fases do processo de KDD, porém todas possuem em comum as etapas apresentadas na figura 1. Devido a possibilidade dos dados serem de diferentes fontes (arquivos, bancos de dados ou diretórios), ou também podendo eles estarem mal formatados, é necessária uma etapa de pré-processamento de dados, a fim de padronizar os dados de forma que possam ser analisados por algoritmos de mineração. Além disto, após a mineração, é necessária a etapa de pós-processamento, buscando garantir que apenas os resultados válidos e úteis sejam levados em conta.

## 2.2. Extração e transformação de dados

De forma geral, antes de se aplicar qualquer algoritmo de mineração de dados, é necessário uma etapa de preparação dos dados. Ao explorá-los, além de obter mais conhecimento sobre os dados, é possível encontrar com problemas que podem comprometer a qualidade da análise, tais como: valores nulos, registros em branco, valores viciados ou duplicados, dentre outros.

A medida que os dados vão sendo entendidos e os problemas encontrados, ocorre a fase de preparação ou pré-processamento dos dados. Segundo Han (et. al., 2006), este processo consiste principalmente nas etapas mostradas na figura 2:

Figura 2 – Fases do pré-processamento de dados.



Fonte: HAN, et. al., 2006.

- **Limpeza de dados:** na grande maioria dos casos, os dados não são perfeitos. Eles contêm inconsistências, tais como registros incompletos ou nulos, registros em branco ou valores errados. A etapa de limpeza dos dados tem como objetivo tratar estes erros de forma que eles não influenciem no resultado dos algoritmos, através de técnicas para remoção dos registros com erros (podendo estes ser ignorados ou terem algum valor padrão atribuído).
- **Integração dos dados:** como citado anteriormente, os dados a serem minerados podem ser provenientes de diversas fontes, como bancos de dados, arquivos textos, páginas na *web*, planilhas, dentre outros. Esta etapa

busca integrar estes dados em apenas um repositório. É necessária uma análise profunda dos dados, identificando redundâncias, conflitos e dependências entre as fontes, a fim de termos um repositório consistente como resultado.

- **Transformação dos dados:** A etapa de transformação dos dados visa colocar os dados em padrões que possam ser compreendidos pelos algoritmos. Para isto, faz-se uso de algumas técnicas, tais como: suavização (remoção de valores errados), generalização (conversão de valores específicos em genéricos), normalização (colocação das variáveis em um mesmo nível), criação de novos atributos (utilizar regras para gerar atributos a partir de outros já existentes) e agrupamento (sumarização de valores).
- **Redução de dados:** Como algumas vezes o volume de dados utilizado na mineração tende a ser alto a ponto do processo de análise de dados ser inviável, técnicas de redução de dados tornam-se necessárias. Estas visam converter a massa de dados original em um volume menor, porém sem perder a representatividade original destes dados, permitindo assim que os algoritmos de mineração sejam aplicados com mais eficiência e com a mesma qualidade de resultados. Dentre as estratégias adotadas, cabe citar: criação de cubos de dados, redução da dimensionalidade e seleção de subconjuntos de dados.

### **2.2.1. Considerações do autor**

Esta é uma etapa chave deste trabalho. Os dados a serem minerados (no caso, os registros de eventos de jogos computacionais) serão de jogos diferentes, e conseqüentemente, terão estruturas diferentes. Deste modo, deverá ser feito um trabalho de pré-processamento sobre estes dados, a fim de estrutura-los de uma forma que os algoritmos de mineração de dados sejam capazes de analisa-los.

Dentre os processos citados, é possível prever a utilização de três dentre eles: limpeza de dados, integração dos dados e transformação dos dados. O processo de limpeza será utilizado para melhor entendimento dos dados (crucial, já que existirão fontes de dados diferentes). O processo de integração poderá ser utilizado se os dados estiverem fragmentados em diferentes arquivos e/ou tabelas

em um banco de dados. E o processo de transformação será vital, já que será necessária a padronização das diferentes fontes de dados existentes.

Será analisada a necessidade de aplicar o processo de redução de dados, dependendo este do comportamento dos algoritmos frente ao tamanho da massa de dados aplicada.

### **2.3. Tarefas em mineração de dados:**

Em mineração de dados não existe um algoritmo ou função padrão que solucione todos os problemas existentes (CORNELIUS JR., 2015). Existem diversas técnicas, cada uma com seus objetivos, e a escolha da que melhor se encaixa no problema proposto faz parte do *Knowledge Discovery in Databases*.

Em alto nível, os dois principais objetivos de uma análise de mineração de dados, na prática, tendem a ser tarefas preditivas e tarefas descritivas (FAYYAD, PIATETSKY-SHAPIRO, SMYTH, 1996).

#### **2.3.1. Tarefas preditivas**

A predição envolve o uso de algumas variáveis ou campos da base de dados para prever valores desconhecidos ou futuros, ou outras variáveis que sejam de interesse (FAYYAD, PIATETSKY-SHAPIRO, SMYTH, 1996).

No aprendizado preditivo, também conhecido como supervisionado, os métodos são providos com um conjunto de dados que possuem uma variável alvo já pré-definida, sendo os registros categorizados em relação a ela (CAMILO, SILVA, 2009).

Classificação é um exemplo de tarefa de predição, sendo, segundo Frank, Hall e Witten (2009), aprender uma função que classifica algum item de dado em uma dentre determinadas classes pré definidas. Nesta tarefa, o modelo analisa o conjunto de registros fornecidos, com cada registro já contendo a indicação à qual classe ele pertence, com o objetivo de “aprender” como classificar um novo registro. (CAMILO, SILVA, 2009).

Outro exemplo de predição são técnicas de regressão, sendo este similar a classificação, porém sendo utilizado quando um registro é identificado por um valor

numérico, assim podendo-se estimar o valor de uma determinada variável analisando os valores das demais (CAMILO, SILVA, 2009). Regressão também abrange a identificação de distribuição de tendências de acordo com os dados disponíveis (HAN, KAMBER, PEI, 2011).

### 2.3.2. Tarefas descritivas

Também chamadas de não supervisionadas, as tarefas descritivas focam em encontrar padrões interpretáveis nos dados (FAYYAD, PIATETSKY-SHAPIRO, SMYTH, 1996). Uma de suas principais diferenças em relação às tarefas preditivas é o fato que estes métodos não precisam de uma pré-categorização para os registros (não é necessário um atributo alvo). Os métodos utilizam alguma medida de similaridade entre os atributos (CAMILO, SILVA, 2009).

As regras de associação estão dentre as tarefas descritivas mais utilizadas, sendo que esta tarefa busca identificar associações interessantes (sejam relações ou dependências) em um grande volume de atributos numa base de dados. Uma das análises clássicas de associação é a Análise de Cesta de Mercado, que visa identificar padrões de compra através da associação entre itens que o comprador coloca em sua cesta de mercado. (CIOS, et. al., 2007).

Outro grande grupo de tarefas descritiva são as tarefas de agrupamento, sendo que elas visam identificar um número finito de categorias (ou grupos) para descrever os dados (FAYYAD, PIATETSKY-SHAPIRO, SMYTH, 1996). Esta tarefa se difere de classificação pois não necessita que os registros sejam previamente classificados (CAMILO, SILVA, 2009).

### 2.4. Análise de agrupamento k-means

O algoritmo *k-means* (MacQueen, 1967) é um método de agrupamento largamente utilizado, sendo uma heurística de agrupamentos não hierárquico e divisível que busca minimizar a distância dos elementos a um conjunto de  $k$  centros dado por  $\chi = \{x_1, x_2, \dots, x_k\}$  de forma iterativa.

De acordo com Wu (et. al., 2007), o algoritmo funciona com um conjunto  $d$  de vetores dimensionais,  $D = \{x_i \mid i = 1, \dots, N\}$ , onde  $x_i \in \mathfrak{R}^d$  denota o  $i$ -ésimo ponto de

dados. O algoritmo é inicializado escolhendo  $k$  pontos em  $\mathfrak{R}^d$  como os  $k$  representantes iniciais de agrupamento ou “centroides”. Neste caso,  $k$  indica o número de grupos desejados. Segundo Tan (et. al., 2009), cada ponto é atribuído a seguir ao centroide mais próximo, e cada coleção de pontos atribuídos a um centroide é um grupo. O centroide de cada grupo então é atualizado baseado nos pontos atribuídos ao grupo. Este passo é repetido até que nenhum ponto mude de grupo ou, equivalentemente, até que os centroides continuem os mesmos.

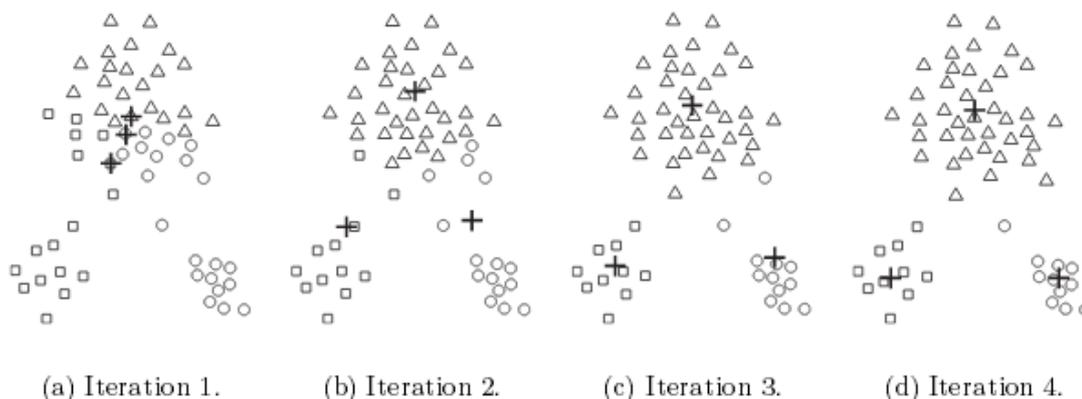
**Tabela 1 – Algoritmo *K-means* básico.**

Passo	Operação
1	Selecione $k$ pontos como centroides iniciais.
2	<b>Repita</b>
3	Forme $k$ grupos atribuindo cada ponto ao seu centroide mais próximo.
4	Recalcule o centroide de cada grupo.
5	<b>Até que</b> os centroides não mudem.

Fonte: TAN, 2009

De acordo com Tan (2009), no primeiro passo, conforme mostra a figura 3, os pontos são atribuídos aos centroides iniciais, simbolizados pelo símbolo de “+”. Após a atribuição, o centroide é atualizado. No segundo passo, os pontos são atribuídos aos centroides atualizados, e estes são atualizados novamente. Os passos 2, 3 e 4 são mostrados respectivamente nas subfiguras (b), (c) e (d) da figura 3, com dois dos centroides se movendo para dois grupos menores na parte inferior das figuras. Quando o algoritmo *k-means* termina, na figura 3(d), porque não há mais mudanças, os centroides identificaram os agrupamentos naturais dos pontos.

Figura 3 – Utilizando o algoritmo *K-means* para encontrar três figuras nos dados de exemplo.



Fonte: TAN, 2009.

Ainda de acordo com Tan (2009), para atribuir um valor ao seu centroide mais próximo, é necessária uma medida que nos indique a noção de “mais perto” para o dado específico em consideração. Para isto, algumas funções para cálculo de distância podem ser utilizadas, tais como, a função Euclidiana de distância, que geralmente é utilizada para *data points* em um espaço Euclidiano, sendo que esta métrica minimiza a distância média entre pontos e os centroides.

Um ponto que pode afetar a qualidade de uma análise com o algoritmo *k-means* é a escolha do número *k* de conjuntos. Caso o número for muito pequeno, o algoritmo pode causar a junção de dois conjuntos naturais, enquanto que se o número de conjuntos for muito grande, um conjunto natural pode ser quebrado em mais conjuntos. (Linden, 2009).

De acordo com MacQueen (1967), talvez a aplicação mais óbvia para o algoritmo *k-means* seja processamento de problemas de “agrupamento por similaridade”. O ponto de vista do algoritmo não é de achar um agrupamento único e definitivo, mas sim simplesmente ajudar na obtenção de uma compreensão qualitativa e quantitativa de uma grande quantidade de dados N-dimensionais, fornecendo grupos de similaridade razoavelmente bons. Ainda segundo MacQueen, este método deve ser utilizado em uma estreita interação com a teoria e a intuição de quem irá analisar os resultados.

### 3. JOGOS COMPUTACIONAIS

O surgimento do primeiro jogo iterativo de computador data da década de 60. No verão de 1961, a *Digital Equipment* doou seu mais novo computador para o MIT, o PDP-I (*Programmable Data Processor-I*). Para testar o poder desta nova aquisição, Steve Russel, estudante do MIT, levou quase seis meses e 200 horas para completar a primeira versão de seu jogo: um simples duelo entre naves espaciais. Utilizando chaves comutadoras adaptadas ao PDP-I, jogadores controlavam a velocidade e direção de ambas as naves, atirando torpedos dentre eles. Russel chamou seu jogo de Spacewar!. (KENT, 2001)

Figura 4 – O PDP-I e o jogo Spacewar!.



Fonte: OVERMARS, 2012.

Alguns anos depois, no final dos anos 60, Nolan Bushnell, aluno da Universidade de Utah e que tinha Spacewar! como jogo favorito, também começou a desenvolver jogos. Seu primeiro desenvolvimento foi um *Tic Tac Toe* computadorizado. Mas não foi este fato que o deixou famoso. Bushnell desenvolveu sua própria versão do Spacewar!, que funcionava com moedas, e projetou um dispositivo com o propósito apenas de suportar seu jogo. Assim surgiu o Computer Space. Encontrando a Nutting Associates como parceira, foram produzidos 1.500 máquinas deste jogo. A iniciativa não deu certo, sendo que nem todas as máquinas produzidas foram vendidas.

Após a falha com Computer Space, Bushnell decidiu iniciar sua própria empresa, surgindo assim a Atari, e seu primeiro grande sucesso, um jogo de padel chamado Pong. Em 1976, a Atari foi vendida para a Warner Communications por

US\$28 milhões. Em outubro de 1977, a Atari lançou o Video Computer System (VCS), com nove cartuchos, sendo um console que poderia ser jogado em casa, o primeiro com múltiplos games.

**Figura 5 – Pong e VCS, respectivamente, lançados pela Atari.**



Fonte: OVERMARS, 2012.

Os próximos anos, entre 79 e 83, foram a era de ouro para os vídeo games, cabendo citar o lançamento de Pac-Man, pela Namco, em 1980, o arcade game mais popular da história, com cerca de 300 mil unidades vendidas, assim como o lançamento de Donkey Kong pela Nintendo, em 1981, e também o lançamento pela Midway de Ms. Pac-Man, o maior arcade na história da América.

Seguindo a linha de tempo da história do vídeo game, outros lançamentos de consoles marcantes foram o Nintendo Entertainment System (NES) e o Sega Master System nos Estados Unidos, em 1986, respectivamente pela Nintendo e pela Sega. A Sega ainda lançou o console de 19-bit Mega Drive em 1989, enquanto a Nintendo lançou seu console portátil GameBoy. Jogos como Sonic, Legend of Zelda, Super Mário 3, Final Fantasy e Tétris foram lançados neste período, e tem seus lugares na história. Ainda é necessário citar o lançamento do Super NES (Nintendo) e Genesis (Sega) nos próximos anos.

Figura 6 – Nintendo NES e seu jogo Super Mario Bros.



Fonte: OVERMARS, 2012.

A próxima grande geração de consoles, agora com gráficos em três dimensões, surgiu no Japão, em 1994, tendo como grande destaque o lançamento do Playstation, pela Sony. Em 1996, a Nintendo entrou para esta geração, com o lançamento do Nintendo 64. Jogos de ambos marcaram esta geração, cabendo citar Crash Bandicoot, Final Fantasy XII, Goldeneye 007 e The Legend of Zelda: Ocarina of Time.

Paralelamente aos consoles, jogos de computadores também se tornaram populares, especialmente na segunda metade da década de 90, com a expansão da Internet e com a chegada de sistemas operacionais mais amigáveis aos usuários. Um dos primeiros grandes sucessos para computadores foi o jogo DOOM, de 1993, que simulava gráficos 3D. Algumas categorias de jogos se tornaram quase que exclusivas para computadores, como jogos de estratégias, como Age of Empires e Command & Conquer, assim como o jogo The Sims, que se tornou em 2002 o jogo de PC mais vendido de todos os tempos, com 16 milhões de cópias.

### 3.1. Conceitos em MMORPG

Com a expansão e popularização da Internet, uma categoria de games que ganhou força nos anos 2000 foi a de MMORPG (*Massive Multiplayer Online Role-Playing Game* / Multiplayer Massivo Online de Jogos de Interpretação de Personagens). Esta categoria surgiu tendo como base o RPG (*Role-Playing Game* / Jogo de Interpretação de Personagens), que consiste em um grupo de jogadores reunidos, com cada um interpretando seu personagem e ditando suas ações e

comportamentos durante a partida. Com a expansão e popularização da internet, foi possível levar o universo dos RPGs para o mundo virtual, integrando e permitindo a interação de milhares de jogadores de todo o mundo em um mesmo universo simultaneamente.

A representatividade dos MMORPGs no mercado de games pode ser exposta com números: World of Warcraft, um dos maiores da categoria, atingiu no final de 2014 a marca de 10 milhões de contas de jogadores ativas, segundo a Blizzard, fabricante do jogo.

**Figura 7 – World of Warcraft, lançado em 2004 pela Blizzard.**



Fonte: WORLD OF WARCRAFT, 2016.

Os MMORPGs desenvolveram naturalmente, ao longo dos anos, toda uma coleção de termos, grande parte propostos pelos próprios jogadores, e que acabaram se estabelecendo como conceitos dentro do jogo. Para melhor entendimento do conteúdo apresentado, alguns destes conceitos devem ser estabelecidos, para ambientação com os termos utilizados dentro do MMORPGs. Os conceitos abaixo apresentados são comuns em todos os jogos desta categoria:

- *Character*: personagem controlado por um jogador dentro do jogo.
- NPC: *Non-player Character*, personagem controlado por inteligência artificial, normalmente utilizado para compra/venda de itens ou como parte de uma *quest*.
- Pontos de experiência: sendo adquirido através de *quests* ou matando criaturas, irá ditar a evolução do personagem, e o incremento de seu nível.

- *Criatura*: também tratado por monstro, personagem “não humano” do jogo, sendo normalmente hostil. O jogador, ao matar uma criatura, será recompensado com pontos de experiência e/ou itens.
- *Quest*: aventura, em tradução literal. Normalmente, dentro do jogo é caracterizada por uma lista de objetivos dados por um NPC, sendo que o jogador que cumpri-los será recompensado com pontos de experiência e/ou itens.
- *Player-killer*: também tratados pela sigla PK ou apenas por *Killers*, são jogadores que tem como objetivo atacar e eliminar outros jogadores.
- *Farmer*: jogadores que permanecem um grande tempo em um mesmo local do jogo, extraindo todos os recursos que este local pode fornecer e acumulando pontos de experiência.

### 3.2. Estruturas de informações nos logs

Um *game log* em um MMORPG é um registro de todos os eventos realizados pelo jogador dentro de uma sessão, ou então “fotografias” dos *status* do jogador contendo suas informações naquele momento. Estas informações são cruciais para as empresas desenvolvedoras dos jogos, já que, em posse delas, podem monitorar o que está ocorrendo em seus servidores, identificando ações que podem ser consideradas antiéticas dentro do jogo e que acabam prejudicando a experiência de jogo de uma maioria.

Estes *logs* fazem registro de cada uma das ações tomadas pelos jogadores, como, por exemplo, cada ordem para atacar uma criatura, ou ações relacionadas à venda ou compra de itens. Também é registrado o horário em que essas ações ocorrem, tendo assim todo o histórico de jogo do jogador registrado.

Com milhares de jogadores conectados simultaneamente e gerando a todo instante conteúdo para estes registros de eventos, é natural que o tamanho destas informações atinjam níveis alarmantes. Mesmo assim, informações estatísticas e individuais, como, por exemplo, ganho de experiência por período ou número de horas jogadas, são de fácil extração.

Porém, devido ao tamanho dessa massa de dados, a extração de informações realmente úteis pode ser um desafio. Como solução para este problema

desponta a mineração de dados, com seus algoritmos capazes de processar esse volume de informações e extrair dali o que for interessante para o negócio.

Para melhor trabalho na análise e aplicação dos *logs* de jogos em algoritmos de mineração de dados, se faz necessário um estudo profundo das informações existentes nestes, categorizando-as. Tendo como base o tipo de informações que será analisada nos logs, serão estabelecidas duas categorias: ações e objetos.

Por ação será definido o que o jogador executou dentro do jogo. Adicionar um item ao inventário, coletar a recompensa de alguma *quest*, atacar alguma criatura são exemplos que podem ser citados. As ações existentes dentro dos *logs* sempre possuem um objeto como alvo, e podem resumidamente serem categorizadas da seguinte forma:

- Ataque: ação de atacar, seja uma criatura ou outro jogador;
- Compra: ação de comprar um item, podendo ser de um NPC ou de um jogador;
- Venda: ação de vender um item, podendo ser para um NPC ou para um jogador;
- Troca: ação de trocar um item com outro jogador.
- Conversa: ação de troca de mensagens com outros jogadores.
- Coleta: ação de coletar algum item, podendo ser recompensa de uma *quest* ou de alguma criatura.

Sempre que uma ação é executada, ela estará relacionada a um objeto. Sendo assim, objeto por sua vez é tudo o que pode ser alvo de uma ação do jogador. Na ação de atacar uma criatura, o objeto será esta criatura atacada. Dentre as categorias de objetos que podem existir, é cabível citar:

- Item: qualquer item dentro do jogo, seja este consumível ou não. Por exemplo, uma poção de cura (consumível), ou uma espada (não consumível).
- Dinheiro: moeda fictícia utilizada para transações comerciais dentro do jogo.
- Depósito: local em que o jogador armazena seus itens e/ou dinheiro.
- Monstro: tratado também por criatura, personagem não humano, predominantemente hostil, alvo de ações de ataque.
- Character: personagem humano, alvo de ações de ataque ou troca.

### 3.3. Categorização de Perfis de Jogadores

Não existe um padrão de jogo em MMORPGs, uma forma comum como todos os jogadores jogam. Como nesta forma de jogo é dada grande liberdade ao jogador, e como é este quem dita a ação que o personagem vai ter dentro do jogo, o comportamento de cada um dentro tende a variar. Porém, mesmo com essas diferenças, alguns padrões de jogo são comuns entre jogadores.

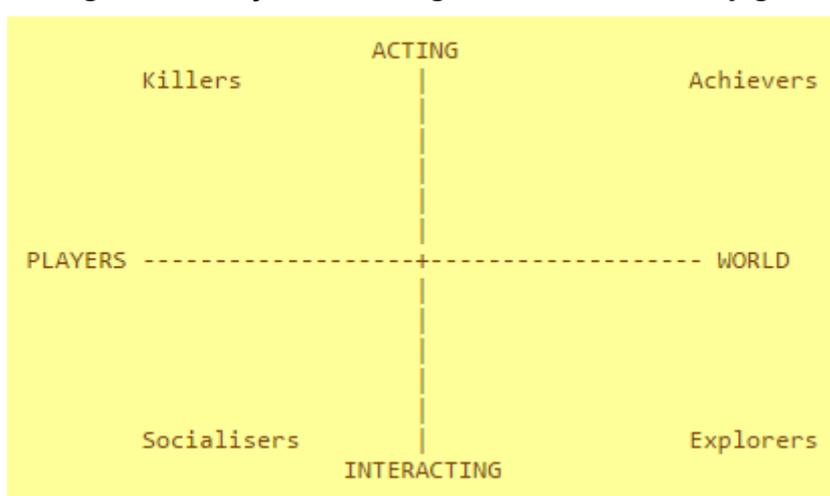
Richard A. Bartle, escritor e pesquisador sobre jogos que participou ativamente no desenvolvimento do MUD(Multi-User Dungeon), escreveu um artigo sobre a personalidade de jogadores em MMORPGs, no qual categoriza os jogadores em quatro perfis e pesquisa quais atividades dentro dos jogos os jogadores mais apreciam, tendo como base seu estilo de jogo. Além disto, ele define como cada categoria de jogador interage com as demais.

Segundo Bartle (1996), os quatro tipos de jogadores que podem ser rotulados são: *achievers* (conquistadores), *explorers* (exploradores), *socializers* (sociais) e *killers* (assassinos). Bartle propõe a seguinte analogia quanto a caracterização: *achievers* são diamantes, sempre em busca de tesouros; *explorers* são pás, eles “cavam” o jogo por informação; *socializers* são corações, eles simpatizam com outros jogadores; e *killers* são clavas, eles batem em outros jogadores com elas. Segue a seguir um detalhamento destas quatro categorias:

- *Achievers*: estes jogadores consideram o ganho de experiência e aumento de nível como seus principais objetivos no jogo. Para estes, exploração só é necessária para achar melhores locais para extração de pontos de experiência. Somente socializam para descobrir com outros jogadores fontes melhores de experiência, e atacam outros jogadores apenas para eliminar alguém que está no caminho, ou caso esta ação proporcione pontos.
- *Explorers*: para estes jogadores, ações de exploração são o foco no jogo. Eles buscam descobrir sobre novos mapas, aprender sobre locais secretos dentro do jogo, e descobrir como as coisas funcionam dentro do jogo. Buscam ganhar experiência apenas se precisarem para alcançar algum local no jogo. Utilizam a socialização apenas como fonte de informações, mas não se prendem muito a estas, pois para estes a diversão está na ação de explorar.

- *Socializers*: estes jogadores estão interessados em relacionamentos, sendo o jogo apenas um plano de fundo para que eles interajam com outras pessoas. Para estes, relações inter-jogador são importantes, tais como desenvolver empatia com pessoas, brincar, se divertir, escutar, ou até mesmo somente observar seus comportamentos. Estes jogadores podem explorar um pouco, a fim de se inteirar sobre um assunto falado no jogo. Buscam ponto de experiência somente em casos que isto melhore seu papel social (adquirir mais nível para melhorar seu status ante a comunidade). Ações de atacar outros personagens geralmente são dispensadas por estes jogadores.
- *Killers*: o objetivo destes jogadores é se impor sobre o outros. Para eles, atacar personagens não humanos não tem utilidade, suas ações são contra personagens controlados por pessoas. Eles atacam outros jogadores apenas com o objetivo de matar seus personagens, sem ser necessária nenhuma justificativa. Podem buscar pontos de experiência apenas para se tornarem fortes o suficiente para batalhar com outros jogadores. A exploração é necessária para aprender sobre os lugares e como eles podem ser utilizados para ajudar a batalhar contra outros jogadores. Utilizam socialização apenas para provocar suas vítimas ou outros jogadores, e discutir táticas de ataque.

Figura 8 – Relação entre categorias e o ambiente do jogo.



Fonte: BARTLE, 1996.

### **3.4. Considerações do autor**

A categorização das informações contidas nos logs de jogos que serão analisados no trabalho será uma etapa crítica. Como o trabalho pretende abordar uma análise sobre registros de jogos diferentes, com estruturas distintas, será necessária uma etapa de tratamento de dados, traduzindo o que está no log para a estrutura desejada.

Para tanto, é necessário conhecimento sobre conceitos utilizados em MMORPGs, assim como quais ações podem ser tomadas dentro do jogo, e sobre quais objetos, e o que estas ações podem significar.

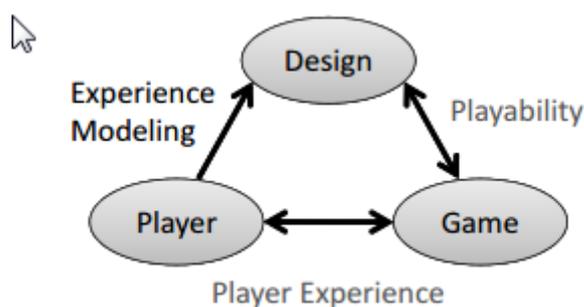
## 4. TRABALHOS RELACIONADOS

### 4.1. Using Data Mining to Model Player Experience

Em seu trabalho, Weber, Mateas e Jhala (2011) utilizam modelos de regressão a fim de modelar com medidas quantitativas a experiência de jogo do usuário, nos jogos Madden NFL 11 e Infinite Mario. O principal objetivo do trabalho proposto é ajudar através destas informações o desenvolvimento de novos jogos, identificando quais elementos do *gameplay* que mais impactam nestas medidas.

Os autores representam a retenção do jogo como uma medida de sua experiência, se referindo há quanto tempo um jogador continua jogando o jogo através de toda a história do jogo. Mesmo a retenção não sendo uma medida direta da experiência do usuário, ela é capaz de fornecer uma medida quantitativa de engajamento do usuário ao jogo.

Figura 9 – Interação entre jogador, jogo e desenvolvedores.

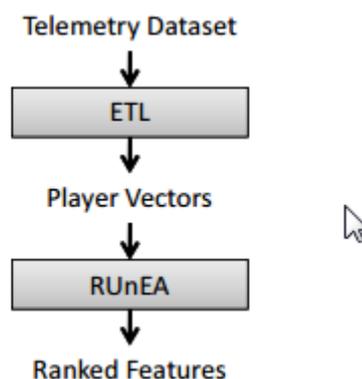


Fonte: WEBER, 2011.

Para esta análise, foi utilizada a técnica *Robust Unique Effect Analysis* (RUnEA), que visa construir e analisar modelos de experiência de jogos. Ela faz uso de algoritmos de regressão para construir modelos para predição de retenção de jogadores, e então analisa o impacto de cada característica do *gameplay* na retenção do jogador. Como entrada, o RUnEA utiliza um conjunto de vetores de características, com os dados de cada jogador gravados em uma posição deste. Para tanto, se faz necessário o uso ETL para o tratamento dos registros dos jogadores, convertendo-os neste conjunto de vetores. Como saída, o RUnEA

apresenta uma lista de características, ordenadas de acordo com a influência que possuem na retenção.

**Figura 10 – Etapas no processo de Data Mining utilizado.**



Fonte: WEBER, 2011.

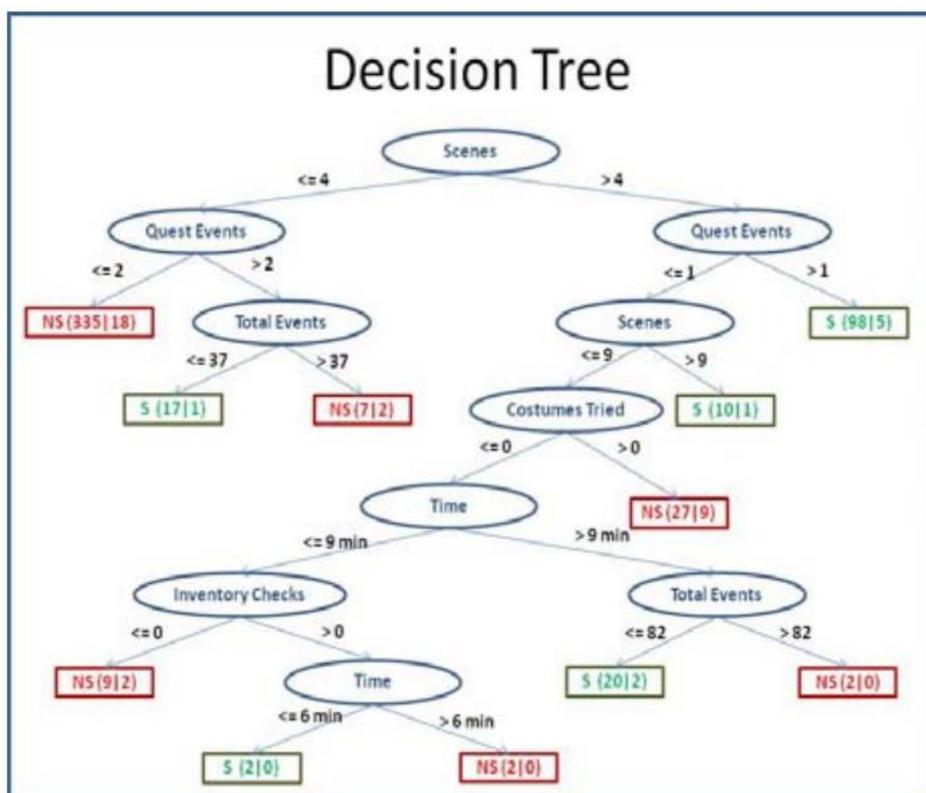
Como resultados do trabalho, os autores comprovaram que o RUnEA pode ser utilizado para melhorar a retenção do jogador ao jogo, fornecendo informações importantes aos desenvolvedores das próximas gerações destes jogos.

#### **4.2. Detecting Player Goals from Game Log Files**

Em seu trabalho, DiCerbo e Kidwai (2012) buscam criar uma forma de detectar se um jogador está perseguindo seriamente os objetivos (*quests*) do jogo. Para isto, são utilizadas informações do jogo Poptropica®, que captura informações de eventos de cada jogador em períodos de tempo.

Para a solução do problema foram utilizadas árvores de classificação e regressão, utilizando o algoritmo J48 (como é chamado o C4.5 dentro do *software* WEKA). Dentre os elementos contidos no arquivo de log que são relevantes em relação ao resultado esperado estão: número total de eventos concluídos na ilha; tempo total gasto na ilha; número total de eventos relacionados com a finalização de *quests*; número de locais visitados na ilha; número de modificação de aparência do personagem; e número de verificações de inventário. Os dois últimos foram selecionados para serem relacionados negativamente a finalização de missões.

Figura 11 – Árvore de decisão final gerada pelo trabalho.



Fonte: DICERBO, 2012.

Segundo os autores, o resultado do trabalho foi satisfatório, sendo que o detector criado consegue identificar de forma confiável se um jogador está perseguindo de forma séria os objetivos para completar *quests* dentro do jogo. Além disto, eles sugerem quatro passos básicos para que trabalhos semelhantes possam ser realizados em outros jogos. São eles:

1. Identificar os objetivos potenciais do jogo.
2. Identificar quais métricas são aplicáveis para esses objetivos.
3. Tratamento de um pequeno conjunto de dados, dizendo se essas métricas são favoráveis ou não aos objetivos.
4. Aplicação da árvore de classificação e análise de regressão.

Por fim, os autores concluem que a identificação dos objetivos do jogador pode ajudar os administradores dos jogos a entenderem as ações que os jogadores tomam dentro do jogo, e aumentar a capacidade de inferência nas características dos jogadores.

### 4.3. Mining Rules from Player Experience and Activity Data

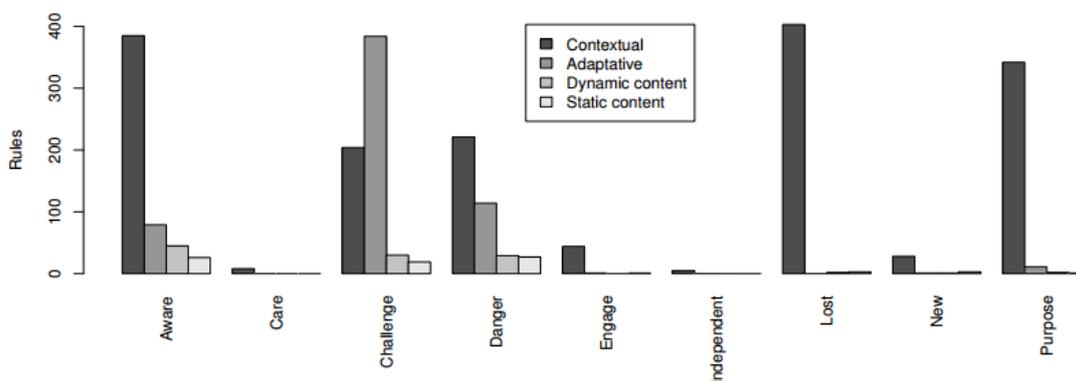
O trabalho apresentado por Colton (et. al., 2012) tem por objetivo gerar conhecimento relevante ao design, através de informações de logs de jogos integradas a dados de *feedback*. O estudo foi feito com 24 jogadores do jogo de tiro em terceira pessoa *Rogue Trooper* (da Eidos Interactive), onde experiências e dados de atividades foram detalhadamente capturada, visando gerar regras de experiência (condições sob as quais experiências específicas são reportadas) significativas e relevantes ao design.

Para o trabalho, foram definidas quatro tipos de regras de experiência, sendo elas: Regras contextuais (descrevem o contexto de uma experiência: estas experiências observadas ao mesmo tempo em que suas consequências); Regras adaptativas (captura a situação diretamente observável associada com uma experiência); Regras de conteúdo dinâmico (associa recursos controláveis da atividade do episódio com uma experiência específica); Regras de conteúdo estático (descrevem como as condições iniciais de um episódio podem impactar na experiência do jogador).

Os registros de dados e testes foram feitos em um ambiente controlado. Os 24 jogadores jogaram do início do primeiro nível por pelo menos 20 minutos (podendo continuar jogando pelo tempo que desejavam, até o final do nível 3). A cada período entre 5 e 10 minutos era aplicado um questionário de experiência. As experiências questionadas eram quanto: a consciência, o cuidado, o desafio, o perigo, o envolvimento, a independência, o propósito, o quão perdido o jogador estava e o quão novo/repetitivo estava sendo a experiência.

Fora isso, foram extraídas do jogo as experiências de combate do jogador, computando 118 características para cada um dos 633 episódios de combate. Um conjunto de características nominais foi então definido para cada combate, por discretização de recursos numéricos como alta, média ou baixa. Para minerar regras de associação entre as características nominais, foi utilizado o algoritmo *FP-Growth*, com o software WEKA, e também utilizadas 4 métricas: confiança, evolução, convicção e influência.

Figura 12 – Exemplo de tabela extraída, com experiência x métrica.



Fonte: COLTON, 2012.

Este estudo mostrou que algumas experiências são melhores modeladas do que outras. Desafio e Perigo possuem uma boa seleção de tipos de regras descrevendo níveis altos e baixos de experiência. Porém outras experiências não são capturadas com tanto sucesso, talvez pela falta de correlação entre experiências ou devido a falta de características relevantes nos dados coletados.

Trabalhos futuros podem abordar como a experiência do jogador pode ser gravada a fim de que as avaliações sejam menos distorcidas, e não dominadas por respostas simples. Os resultados também podem ser melhorados através de seleção de melhores características e técnicas de discretização, e aplicação de regras de mineração focadas em valores de experiência específicos.

#### 4.4. Data Mining and Machine Learning with Computer Game Logs

Em seu trabalho, Wender (2007) tem como principal objetivo descobrir padrões em um conjunto de dados, através da aplicação de algoritmos de aprendizado de máquina, visando utilizar estes padrões com dois principais objetivos: melhorar a experiência de jogo de jogadores; e auxiliar na criação de jogadores artificiais, possibilitando que estes tenham comportamentos mais humanos.

O trabalho tem como base o jogo Team Fortress Classic, um jogo *multiplayer online* de tiro em primeira pessoa, baseado em times de jogadores que competem em diferentes cenários contra outros times. Neste caso, foi somente considerado o modo de jogo *capture-the-flag*, onde o objetivo do time é capturar a bandeira do time

adversário. O jogo conta com 9 diferentes classes que podem ser escolhidas pelo jogador, cada uma com seus pontos fortes e fracos. Os *logs* de jogos foram extraídos através do site *The Blarghalizer*, que fornece um serviço para análise dos arquivos de *logs* das partidas de Team Fortress Classic.

**Figura 13 – Estatística dos dados utilizados.**

Number of recorded games:	6099
Number of recorded players:	90524
Average players per team:	7.36
Number of recorded kills:	4,352,279
Average kills (including own team) per player per game:	48.01
Number of recorded class choices:	157,767
Average number of different classes per player:	1.74
Number of recorded flag captures:	84,469
Included Coast-To-Coast-Captures:	15,087 (17,86% of all Captures)
Average number of captures per game:	13.73
Number of Touches:	597,821
Average number of Touches per Capture:	7.08
Average recorded game time:	29 Minutes 17 Seconds

Fonte: WENDER, 2007.

Para a análise, foi feito uso do algoritmos J48 (C4.5 *Decision Tree*), o modelo Naive Bayes, o algoritmo k-nn (*Nearest Neighbor*) e também Redes Neurais de 2 camadas que usa *Backpropagation* para treinamento. São realizadas duas abordagens para a busca de padrões aplicando algoritmos de aprendizagem de máquina.

A primeira abordagem visa prever resultados de jogos através da composição dos times. Neste caso, é necessária uma análise de desempenho de classes em diferentes cenários de jogos, e como são formados os times nestes cenários. A segunda abordagem visa a previsão de resultados baseado na performance de classes. Neste caso, as principais atividades consideradas são *kills* (ação de matar um oponente) e *flag activities* (atividades em relação a captura de bandeira). Neste ponto, uma dificuldade encontrada foi a simplificação do significado de performance, já que as classes possuem pontos fortes e fracos distintos, e conseqüentemente, funções diferentes dentro do time.

Os resultados obtidos não foram satisfatórios, podendo ser por não existirem padrões identificáveis nos dados utilizados, ou então por estes padrões não serem

tão óbvios, necessitando de mais variáveis na análise. Porém, o autor salienta que mesmo assim, os resultados podem ser considerados promissores. Para trabalhos futuros, ele pretende fazer uma abordagem com uma massa de dados maior para avaliação, e também com refinamento nas métricas utilizadas nas análises.

#### 4.5. A Data Mining Approach to Strategy Prediction

Através de abordagens de mineração de dados em uma grande quantidade de *logs* de jogos, Weber e Mateas (2009) buscam em seu trabalho modelar oponentes para jogos de estratégia. Segundo os autores, um dos maiores desafios para designers de games é a criação de Inteligências Artificiais (IAs) adaptativas em oponentes, para reagir as ações do jogador.

No caso de um jogo de estratégia em tempo real, um oponente inteligente deveria ser capaz de responder a uma estratégia de um jogador com uma contra estratégia. Para tanto, a IA do oponente deve reconhecer a estratégia que o jogador está usando. Desta forma, o artigo explora uma abordagem com *Data Mining* para reconhecer estratégias em jogos de tempo real com informações parciais. Para este trabalho, os autores utilizaram informações do jogo de estratégia em tempo real StarCraft, desenvolvido pela Blizzard EntertainmentTM.

**Tabela 2 – Exemplo de registro de ações dos jogadores.**

Player	Game Time (minutes)	Action
Player 2	0:00	Train Drone
Player 1	0:00	Train SCV
Player 2	1:18	Train Overlord
Player 1	1:22	Build Supply Depot
Player 1	2:04	Build Barracks
Player 2	2:25	Build Hatchery
Player 1	2:50	Build Barracks

Player 2	2:54	Build Spawning Pool
Player 1	3:18	Train Marine
Player 2	4:10	Train Zergling

Fonte: WEBER, 2009.

Neste trabalho foi utilizada uma forte camada de ETL, para extração das informações de sites de estatísticas do jogo e transformações destas informações em estruturas compatíveis às técnicas de mineração de dados utilizadas. Após isto, foram feitas duas análises diferentes sobre os dados: predição estratégica e predição de tempo.

Na predição estratégica, foram utilizados os seguintes algoritmos: J48 (*C4.5 decision tree*), k-NN (*Nearest neighbor*), NNge (*Non-nested generalized exemplars*) e LogitBoost (*Additive logistic regression*), todos aplicados utilizando o software WEKA. Os algoritmos mostraram variação do comportamento e precisão nos resultados quando aplicados em diferentes momentos do jogo, e como resultado se mostraram capazes de prever a estratégia a ser executada pelo jogador com certa precisão.

Quanto a predição de tempo, este foi representado como um problema de regressão, sendo os seguintes algoritmos aplicados: ZeroR (*Estimates the mean*), LinearRegression (*Akaike linear regression*), M5' (*Inducing model trees*) e AdditiveRegression (*Stochastic gradient boosting*). Para esta análise os resultados mostraram que o M5' fez a predição de tempo com o menor erro médio dentre as técnicas aplicadas. Porém, os resultados mostraram que a regressão obteve bons resultados apenas com um subconjunto das características.

Também foram feitas análises comparativas de tempo de resposta de todas as técnicas analisadas, chegando a conclusão que todos os algoritmos poderiam ser utilizados uma vez por segundo com mínimo impacto no sistema.

Como trabalhos futuros, os autores citam duas direções principais. A primeira seria focar na melhora da precisão das análises de previsão de estratégia e de tempo. A segunda seria incorporar os resultados das análises resultantes da mineração de dados em um agente de jogo em um jogo completo.

#### 4.6. Game Bot Detection Approach Based on Behavior Analysis and Consideration of Various Play Styles

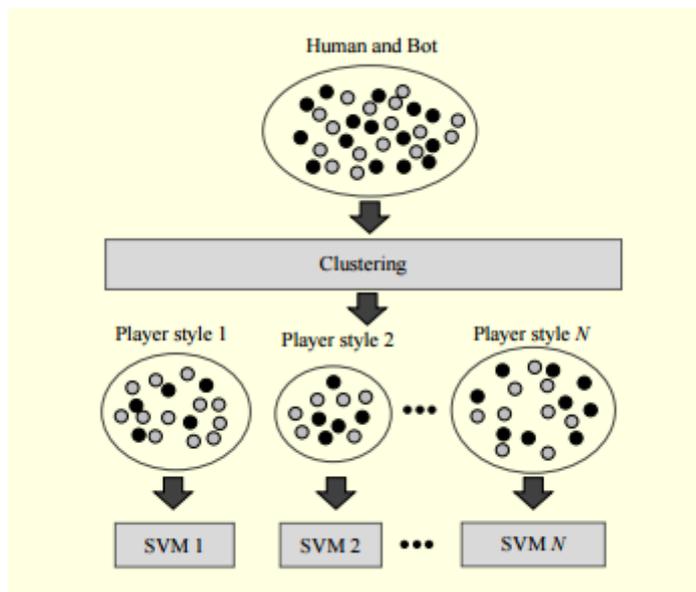
De acordo com Chung, Y (et. al., 2013), *game bots* são programas automatizados de forma a realizar ações repetitivas no lugar de jogadores humanos, utilizados para obter uma vantagem injusta em relação a jogadores honestos, já que podem realizar ações tediosas sem pausas. Sendo assim, são uma potencial ameaça para o negócio de games, sendo a detecção de uso de importância para os administradores de servidores de MMORPGs.

Em seu trabalho, os autores propõem uma metodologia de detecção de *game bots* baseada em análise de comportamento, agrupando jogadores pelas suas similaridades de comportamentos, detectando *bots* em cada grupo e utilizando um conjunto de regras de detecção customizadas para cada grupo de jogadores. Como fonte de dados, foram utilizados *logs* do MMORPG coreano Yulgang Online.

Quanto as características que podem ser encontradas dentro do jogo, elas foram classificadas em três categorias, todas comuns em MMORPGs, da seguinte forma: Batalha, contendo caça, ataque, acerto (de ataque), defesa, esquiva e cura; Coleta, contendo Item, Inventário e Largar (um item); e Movimento, contendo X, Y e uso de portal.

A metodologia proposta é dividida em duas fases. A primeira busca dividir os jogadores de acordo com suas similaridades de comportamento. A segunda, aplicar um modelo para detecção de *game bots*, com regras distintas para cada grupo de jogadores. Para agrupar os jogadores de acordo com suas similaridades, é utilizado o algoritmo de análise de agrupamento *k-means*, obtendo um número *k* de grupos de jogadores, cada um com jogadores com padrões de jogos semelhantes. Para cada grupo de similaridade, é gerado uma Máquina de vetores de suporte (*Support Vector Machine*, SVM), uma técnica de aprendizado de máquina utilizada em classificação binária.

Figura 14 – Detecção de bots considerando diferentes estilos de jogo.



Fonte: CHUNG, 2013.

Através dos experimentos, analisando os resultados, foi verificado que a proposta apresentada foi efetiva, e a abordagem de aplicação de regras distintas para cada grupo de jogadores produziu resultados mais acurados e estáveis, para todos os estilos de jogos.

#### 4.7. Conclusão sobre os trabalhos relacionados

Conforme será descrito no capítulo 5, a solução desenvolvida teve como base a aplicação de técnicas de mineração de dados em registros de eventos de jogos MMORPG, para classificação de perfis de jogadores. Para auxiliar na decisão de quais técnicas melhor se aplicam, foram analisados os trabalhos descritos acima. Para uma análise mais completa, foram selecionados trabalhos com certa distinção entre si, a fim de manter as possibilidades de escolha de técnicas o mais aberta possível.

Tabela 3 – Comparativo entre trabalhos relacionados.

Trabalho	Objetivo	Técnicas utilizadas	Resultados
4.1: WEBER, B. et. al.	<p>Mensurar em medidas quantitativas a experiência do jogador.</p> <p>Medir o nível de retenção do jogo.</p>	<p>ETL, para conversão de registros de jogo em conjunto de vetores.</p> <p>Uso de algoritmos de regressão com a ferramenta Weka.</p> <p>Aplicação da técnica <i>Robust Unique Effect Analysis</i> (RUnEA).</p>	Positivos, sendo o RUnEA capaz de apontar características que retém o jogador.
4.2: DiCerbo, E., et. al.	Detectar se o jogador está perseguindo os objetivos do jogo de forma séria.	Aplicação do algoritmo de classificação e árvore de regressão J48 (C4.5).	Satisfatórios, o detector criado identifica com sucesso as intenções do jogador.
4.3: Colton, S., et. al.	Gerar regras de experiência relevantes para o design.	<p>Ambiente controlado, com jogo sendo monitorado e testes escritos aplicados.</p> <p>ETL para juntar as informações de log com testes.</p> <p>Algoritmo FP-Growth para mineração, com a ferramenta Weka.</p>	Não satisfatórios, não sendo possível detectar todas as regras de experiência.
4.4: Wender, S.	Identificar padrões nos logs de dados, com objetivo de melhorar a experiência do jogador e auxiliar na criação de jogadores	<p>ETL para extração dos logs.</p> <p>Aplicação das técnicas J48 (C4.5 <i>Decision Tree</i>), modelo Naive Bayes, k-nn(<i>Nearest</i></p>	<p>Não satisfatórios, ou pela não existência de padrões, ou por padrões não tão óbvios para análise.</p> <p>Trabalhos futuros</p>

	artificiais.	<i>Neighbor</i> ) e Redes Neurais.  Uso da ferramenta Weka.	com métricas melhor definidas e massa de dados maior.
4.5: Weber, B., Mateas, M.	Reconhecer estratégias em jogos de tempo real com informações parciais.  Obter informações úteis para auxílio na criação de IAs.	ETL, para extração e transformação das informações em estruturas compatíveis com os algoritmos.  Na predição de estratégia: Algoritmos J48 (C4.5 <i>Decision Tree</i> ), k-NN, NNge e LogiBoost, aplicados no software Weka.  Na predição de tempo: ZerpR, LinearRegression, M5' e AdditiveRegression.	Para predição de estratégia: resultados capazes de prever próximas ações do jogador.  Para predição de tempo: bons resultados apenas com subconjuntos das características.  Análise de tempo mostrou que todos os algoritmos podem ser utilizados em tempo real sem grande impacto ao jogo.
4.6: Chung, Y., et. al.	Detecção de <i>game bots</i> baseado na análise de comportamento, agrupando jogadores de acordo com sua similaridade comportamental.	Algoritmo <i>k-means</i> , para agrupamento de jogadores em grupos de similaridades.  Para cada grupo gerado, geração de Máquina de vetores de suporte (SVM).	Satisfatórios, com a aplicação de regras distintas para cada grupo de jogadores sendo capaz de gerar resultados acurados e estáveis.
Trabalho proposto	Categorização de perfis de jogadores segundo padrões em logs.  Identificação de características que auxiliam na	ETL para tratamento das informações dos logs.  Algoritmos C4.5 ( <i>Decision Tree</i> ), <i>k-nn Nearest Neighbor</i> e <i>k-</i>	

	categorização.  Análise comparativa entre desempenho de algoritmos.	<i>means</i> .  Utilização da ferramenta Weka para aplicação das técnicas de <i>data mining</i> .	
--	---	---	--

Fonte: O autor.

Conforme a tabela 4, foi possível concluir que dentre todas as técnicas aplicadas nos trabalhos, a que se mostrou mais efetiva para classificação de jogadores foi o algoritmo de análise de agrupamento *k-means*. Além deste, o algoritmos C4.5 (*Classification and Regression Tree*) também parece promissor para a realização deste tipo de análise sobre os registros de eventos. Além disto, dos seis trabalhos descritos, cinco utilizam em algum momento de sua análise o *software* WEKA, uma ferramenta em Java que possui uma lista de algoritmos de aprendizado de máquina para aplicação de mineração de dados.

## 5. SOLUÇÃO DESENVOLVIDA

Como já abordado nos capítulos anteriores, técnicas de *Data Mining* tendem a ser a melhor opção no tratamento de registros de eventos de servidores de MMORPGs, devido a sua capacidade de trabalhar com grandes massas de dados, características destas estruturas, e extrair delas informações úteis.

A solução aqui apresentada verificou a capacidade do algoritmo *k-means* de agrupar os jogadores de acordo com os perfis propostos por Bartle (abordado no capítulo 3), através da análise de fontes de dados de dois jogos diferentes, identificando os atributos que melhor se aplicam na análise, quais métricas devem ser consideradas e se o algoritmo consegue retornar resultados coerentes. Porém, para a definição de perfis de personagens, foram feitas pequenas alterações nos perfis que Bartle propôs em seu trabalho, as quais são citadas logo a seguir. As alterações são sutis, apenas para contextualizar os perfis propostos com os objetivos do trabalho.

Para os **Achievers**, pouco se alterou, sendo apenas dado maior destaque as ações que buscam a obtenção de pontos de experiência através de caça (ataque sucessivo a criaturas no jogo) e a coleta de itens. As definições de perfil de **Killers** se mantém as mesmas apresentadas, sendo jogadores que buscam atacar outros personagens humanos.

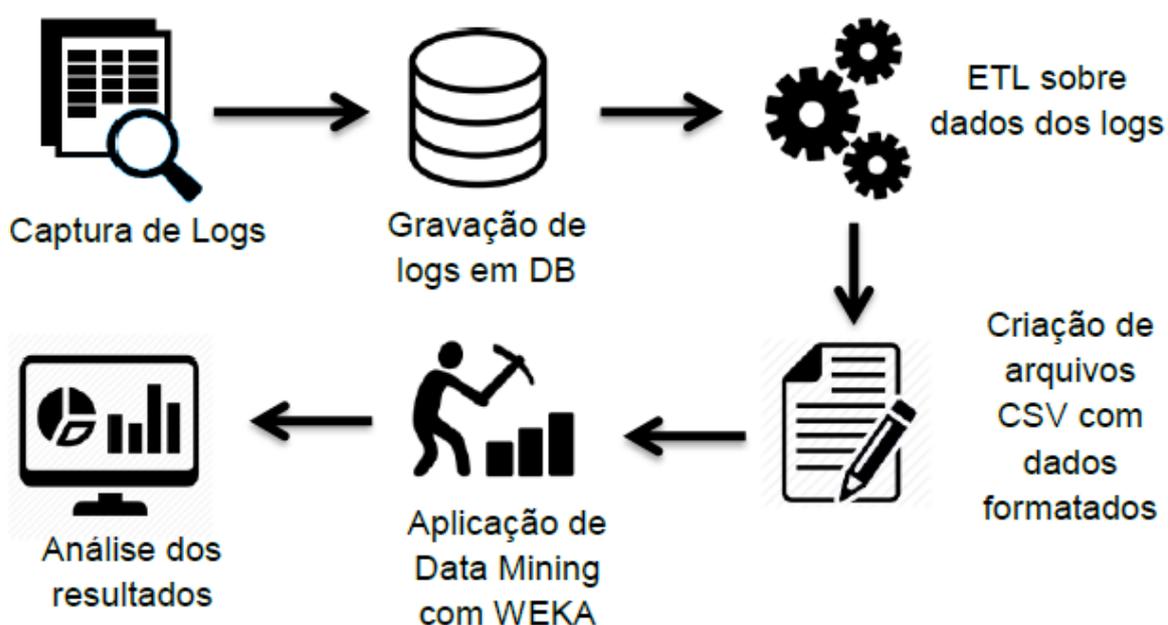
As alterações mais significativas são nos perfis de **Socializer** e **Explorers**. Para o primeiro, foi dado também destaque as ações de comércio e inventário realizadas por estes jogadores, além das demais iterações sociais. Quanto aos **Explorers**, observou-se especialmente a grande quantidade de ações de movimento, e a pouca atividade de caça (tendo como consequência menor ganho de pontos de experiência) e de obtenção de itens.

Foram analisadas duas estruturas de registros de dados, sendo a primeira do jogo **Tibia**, e a segunda gerada pelo software **Zereal Agent-based Simulator**, ambas com modelagens bastante distintas. No jogo **Tibia**, a base de dados é estática. Esta base não grava cada ação executada pelo jogador, apenas grava “fotografias” a cada vez que o jogador é desconectado do jogo, contendo seus status naquele momento (nível, pontos de experiência, itens, quantidade de dinheiro, amigos, etc). A base gerada pelo **Zereal** é dinâmica, ou seja, ela grava cada ação

realizada pelo jogador dentro do jogo. Cada movimento realizado, cada ataque e cada aquisição de item é gravado dentro de um registro de eventos desse jogador.

A figura 15 mostra todas as etapas envolvidas nas análises realizadas. As etapas de captura de *logs* e gravação de *logs* em DB ocorrem no próprio jogo, onde as informações dos jogadores são capturadas e gravadas. A partir destas bases de dados é realizado o ETL sobre os *logs* dos jogos, gerando um arquivo CSV delimitado por vírgulas como saída. Após isto, o arquivo gerado é aplicado ao *k-means* através da ferramenta WEKA, e por fim os resultados obtidos são analisados.

Figura 15 – Etapas do processo, da captura dos logs à análise dos resultados.



Fonte: O autor.

### 5.1. Análise sobre base de dados estática

Conforme já citado, uma base de registros estática em um MMORPG irá gravar em um banco de dados apenas “fotografias” contendo o status que o jogador se encontra no momento em que ele se desconecta do jogo. Informações como nível, pontos de experiência, itens possuídos, quantidade de dinheiro e lista de amigos são gravadas nesta base.

Um problema deste tipo de base de dados é que apenas a última posição dos status do jogador é mantida gravada, sendo todo o histórico perdido. Desta forma,

uma análise sobre perfis de jogadores são necessárias bases de mais de um dia, a fim de validar as informações do jogador.

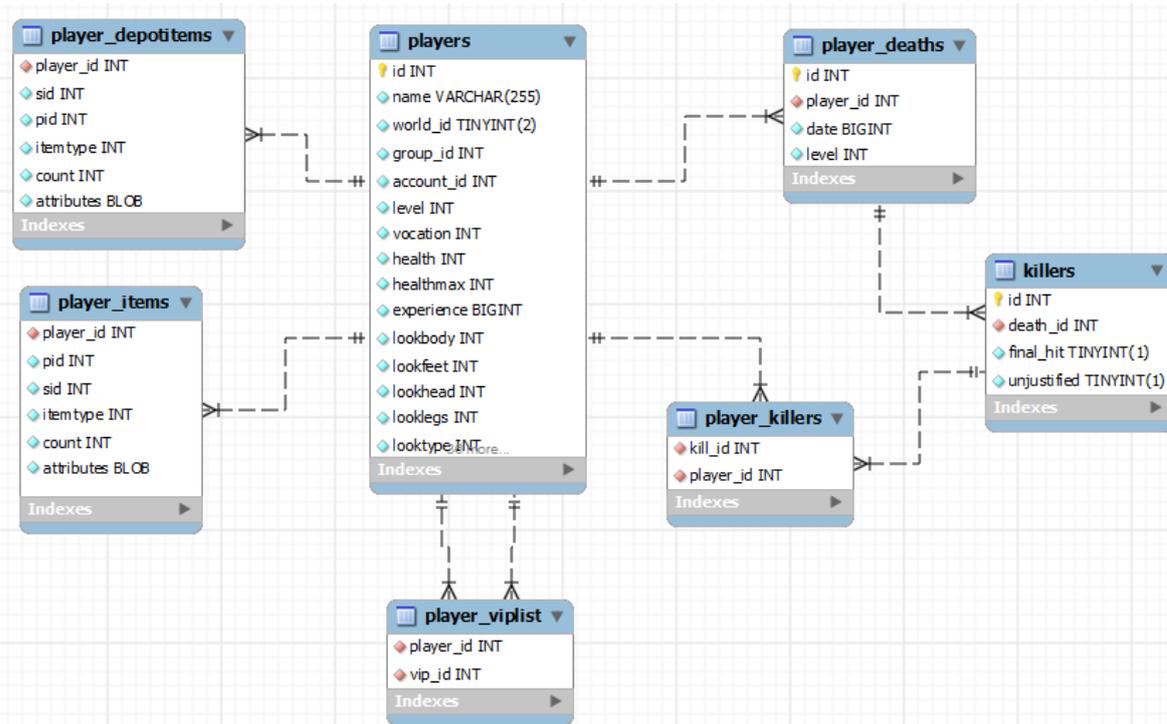
Para a análise sobre dados de uma base estática foram utilizadas informações do jogo Tibia, da alemã CipSoft. Lançado em 1997, é um dos MMORPGs mais antigos ainda em atividade. Uma particularidade sobre o Tibia é que este permite servidores alternativos serem criados, sendo estes não vinculados à CipSoft. Os dados obtidos são de um servidor alternativo, sendo que foram capturados registros de três dias consecutivos para a análise. As bases de dados consistem em três arquivos de SQLite (.s3db), respectivamente dos dias 28, 29 e 30 de maio de 2016.

Verificando a estrutura da base de dados, foram descartadas informações referentes a contas de jogador e as configurações do servidor. Dentre das informações referentes ao jogador, foram consideradas válidas para análise as seguintes informações:

- id do jogador
- nível do jogador
- pontos de experiência
- quantidade de itens
- dinheiro
- amigos
- quantos outros personagens o jogador matou.

A estrutura de dados contendo as tabelas e os relacionamentos de onde são obtidas estas informações é apresentada na figura 16, e descritas logo em seguida.

Figura 16 – Estrutura das informações de jogador no MMORPG Tibia.



Fonte: MySQL Workbench. Adaptado pelo autor.

- Tabela *players*: retorna todas as informações únicas relacionadas ao jogador, contendo seus principais status no momento de seu último *logoff* no jogo. Dentre elas, as mais importantes para a análise são as colunas *level* (nível que o jogador se encontra no jogo no momento), *experience* (quantidade de pontos de experiência do jogador) e *sex* (sexo do personagem no jogo). A coluna *sex* será útil apenas para excluir da análise jogadores do tipo *God*, sendo estes administradores do jogo.
- Tabela *player\_items*: contém todos os itens que o jogador possui consigo no momento, sejam equipamentos, poções, itens de comida ou moedas (dinheiro do jogo). Serão extraídos desta tabela as colunas *player\_id* (ID do jogador) e *count* (quantidade do item). Para as análises serão desconsiderados os itens de *comida* e *poções*, já que normalmente os jogadores possuem grande quantidade destes itens, e eles atrapalhariam a análise dos dados. Quanto às moedas, existem três tipos que são consideradas: (1) *gold coins*, que vale uma Unidade de Valor (UV), (2) *platinum coins*, que vale 100 UV e (3) *crystal coins*, que vale 10.000 UV. Toda a análise como dinheiro utiliza os valores convertidos para a unidade de *gold coins*.

- Tabela *player\_depotitems*: contém todos os itens que pertencem ao jogador e estão em seu depósito dentro do jogo. Para análise, extrai as mesmas informações e utiliza as mesmas regras da tabela *player\_items*.
- Tabela *player\_viplist*: contém gravado a lista *vip* de cada jogador, sendo considerada a lista de amigos que este jogador possui dentro do jogo. Serão extraídos o *player\_id* (ID do jogador) e uma contagem da coluna *vip\_id* (ID do “amigo”).
- Tabelas *player\_deaths*, *killers* e *player\_killers*: retorna todas as informações sobre mortes de jogadores dentro do jogo. As informações extraídas sobre mortes de jogadores são da tabela *player\_killers*, sendo a coluna *player\_id* (ID do jogador) e uma contagem sobre a coluna *kill\_id* (chave com a tabela *killers*). Além disso, através de *joins* com a tabela *killers*, também será extraída a coluna *date* da tabela *player\_deaths*, para saber a data em que a morte ocorreu.

Para melhor analisar o padrão de comportamento dos jogadores, foram consideradas para análise apenas as informações referentes a jogadores que possuem informações diferentes nas três “fotografias” de dados, garantindo assim mais precisão no momento de determinar o padrão de comportamento do jogador. A análise sobre as informações estáticas é composta por três etapas: (1) extração e transformação das informações, (2) aplicação do algoritmo *k-means* e (3) avaliação dos resultados.

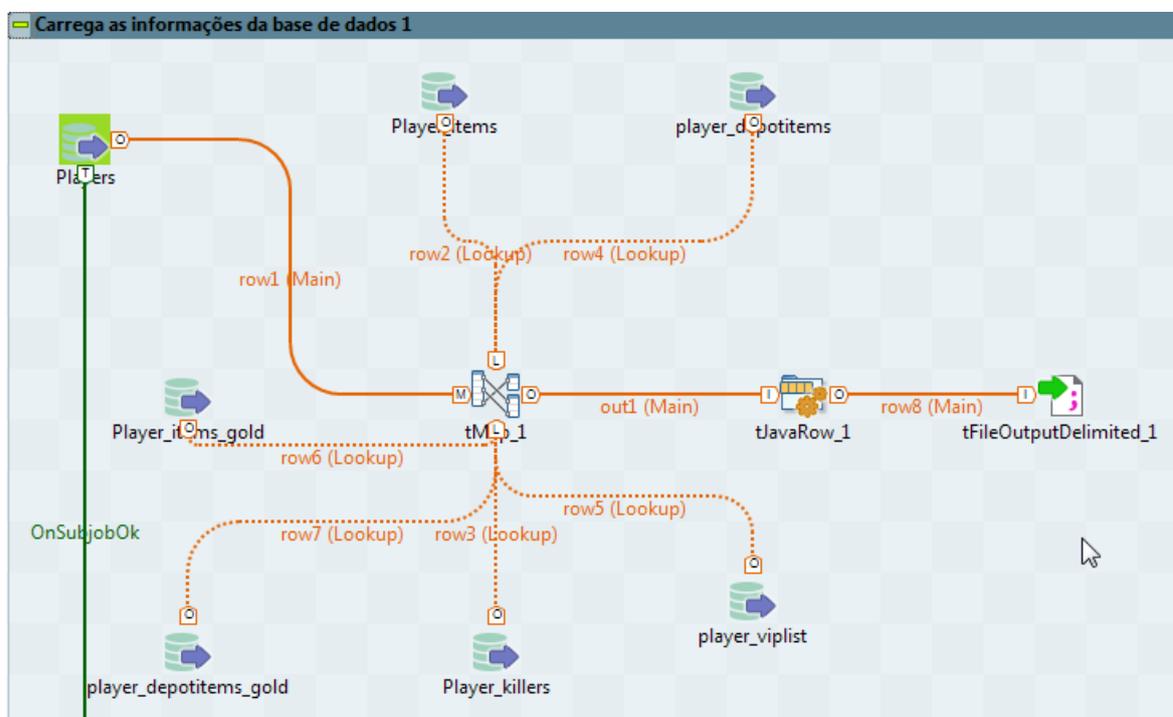
### 5.1.1. Extração e transformação das informações

A etapa da extração e transformação das informações foi realizada utilizando a ferramenta **Talend Opens Studio**. Foram efetuadas duas etapas, extração das informações de cada uma das “fotografias” de dados; e tratamento das informações para serem lidas pelo algoritmo.

A extração das informações da base de dados consiste em leitura das tabelas que gravam informações relevantes do jogador e escrita destas informações em arquivos intermediários, sendo que estes arquivos irão conter um resumo com as informações abertas por jogador referentes à base de dados extraída. Esta etapa é

executada uma vez para cada arquivo de base de dados, sendo gerado um arquivo diferente para cada um deles.

Figura 17 – Representação da extração das informações de jogador no Talend.



Fonte: Talend. Adaptado pelo autor.

Cada tabela é lida individualmente, sendo as informações extraídas de cada uma delas descrita abaixo:

- *players*: são extraídas as informações de *id*, *sexo*, *experiência* e *nível* do jogador, considerando apenas as informações dos jogadores que se conectaram ao jogo na data da “fotografia” dos dados. Na figura 17, é representado pelo objeto *Players*.
- *player\_items*: contém as informações de *itens* que o jogador possui junto a si. São feitas duas extrações diferentes sobre esta tabela. A primeira busca os *itens* de forma geral do jogador, somando a quantidade de *itens* por jogador e ignorando quando o *item* for moedas, sendo representado na figura 17 pelo objeto *Player\_items*. A segunda busca apenas as moedas que o jogador possui, contando assim a quantidade de *dinheiro* que ele possui no jogo, ignorando todos os outros itens. É representado pelo objeto *Player\_items\_gold*.

- *player\_depotitems*: consulta semelhante à *player\_items*, porém busca os *itens* que o jogador possui em seu depósito no jogo. Também faz duas busca, uma delas considerando apenas *itens* (objeto *player\_depotitems*) e a outra considerando apenas as *moedas* que o jogador possui (objeto *player\_depotitems\_gold*).
- *player\_viplist*: a consulta nesta tabela retorna a quantidade de *amigos* que o jogador possui dentro do jogo. É realizada uma contagem de quantos *amigos* cada jogador possui. Esta consulta é representada na imagem pelo objeto *player\_viplist*.
- *player\_killers*: a consulta nesta tabela traz o *id do player* que executou a morte, e o *id da morte* de outro jogador. É efetuado um *join* com a tabela *player\_deaths*, através da tabela *killers*, para buscar a data das mortes contidas na tabela. Sendo assim, basta apenas contar em quantas mortes diferentes o jogador teve participação na data da gravação da base de dados para obter a quantidade de jogadores que ele matou. O objeto que representa esta consulta é o *Player\_killers*.

Após a leitura das tabelas, é realizado um mapeamento (representado pelo objeto *tMap*), no qual todas as tabelas de leitura de dados são ligadas, sendo a tabela *players* a mandante no mapeamento. A ligação com as outras tabelas é feita através do *id do jogador*. Como estrutura de saída, é gerado um arquivo CSV separado por ponto e vírgula (um para cada “fotografia” da base), contendo um resumo das informações do jogador na imagem da base de dados lida. A estrutura deste arquivo pode ser observada na figura 18.

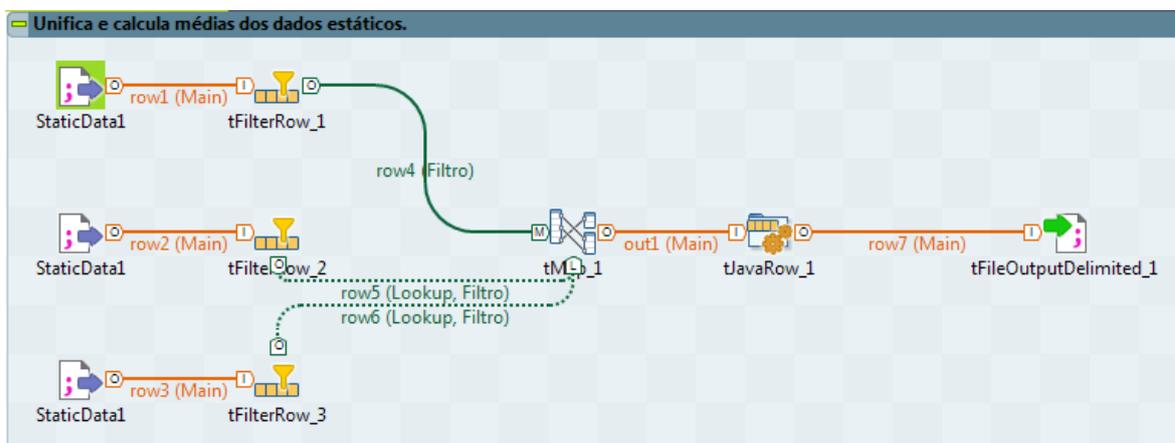
**Figura 18 – Estrutura do arquivo com informações do jogador unificadas.**

Column	Key	Type	✓	N..	Date Pa...	Len...	Pre...	D...	Co...
id	<input type="checkbox"/>	Stri...	<input type="checkbox"/>			0	0		
experience	<input type="checkbox"/>	Stri...	<input checked="" type="checkbox"/>			0	0		
level	<input type="checkbox"/>	Stri...	<input checked="" type="checkbox"/>			0	0		
sex	<input type="checkbox"/>	Stri...	<input checked="" type="checkbox"/>			0	0		
num_items	<input type="checkbox"/>	Int...	<input checked="" type="checkbox"/>			0	0	0	
num_gold	<input type="checkbox"/>	Int...	<input checked="" type="checkbox"/>					0	
num_kills	<input type="checkbox"/>	Stri...	<input checked="" type="checkbox"/>					0	
num_friends	<input type="checkbox"/>	Stri...	<input checked="" type="checkbox"/>					0	

Fonte: Talend. Adaptado pelo autor.

O próximo passo é a unificação das informações de cada jogador, para obter uma média de seus dados em cada um dos dias. Desta forma, cada um dos arquivos é lido novamente, e todos eles são ligados a um mapeamento, sendo o arquivo de base de dados mais antigo o mandante na ligação entre eles. À cada arquivo é aplicado um filtro para que sejam desconsiderados jogadores cujo sexo seja *God*, sendo estes, conforme dito anteriormente, administradores do jogo e por isso irrelevantes para análise. Para a ligação entre os arquivos é utilizado *inner join* a partir do *ID do jogador*, de forma que exista a garantia de que sejam levados para o arquivo final apenas aqueles jogadores que possuem dados nos três arquivos de base de dados. Desta forma, um total de 111 jogadores foram gravados no arquivo de saída. A figura 19 ilustra a etapa de tratamento dos arquivos.

**Figura 19 – Tratamento dos arquivos de “fotografia” da base de dados.**



Fonte: Talend. Adaptado pelo autor.

O arquivo final, sugerido por este trabalho, para servir de entrada no algoritmo *k-means* tem a seguinte estrutura:

- *id* – ID do jogador
- *experience* – Média de ganho de experiência no jogador dentre as três “fotografias” de dados. São considerados apenas valores positivos.
- *level* – Nível do jogador na arquivo de dados mais recente.
- *var\_level* – Variação total de níveis que o jogador obteve nos três dias, sendo a subtração entre o nível do terceiro dia e o nível do primeiro dia.
- *num\_items* – Média de ganho de itens do jogador dentre os três dias jogados.
- *num\_gold* – Média de ganho de dinheiro (*gold coins*) do jogador dentre os três dias jogados.
- *num\_kills* – Média de *kills* (assassinatos de outro personagem) que o jogador executou dentre os três dias jogados.
- *num\_friends* – Número total de outros jogadores que o jogador em questão possui em sua lista de amigos, sendo considerada a informação do arquivo de base de dados mais recente.

**Figura 20 – Arquivo de saída pronto para ser aplicado ao *k-means*.**

id	experience	level	var_level	num_items	num_gold	num_kills	num_friends
435	189596	52	5	8	234928	0	16
727	1604151	68	48	65	48985	0	1
1059	2984816	86	45	112	33830	0	0
1099	2866918	90	33	34	93458	0	0
1633	215283	69	3	5	247332	0	25
1250	2116896	255	2	12	11245	4	2
297	3207756	95	32	25	69603	0	2

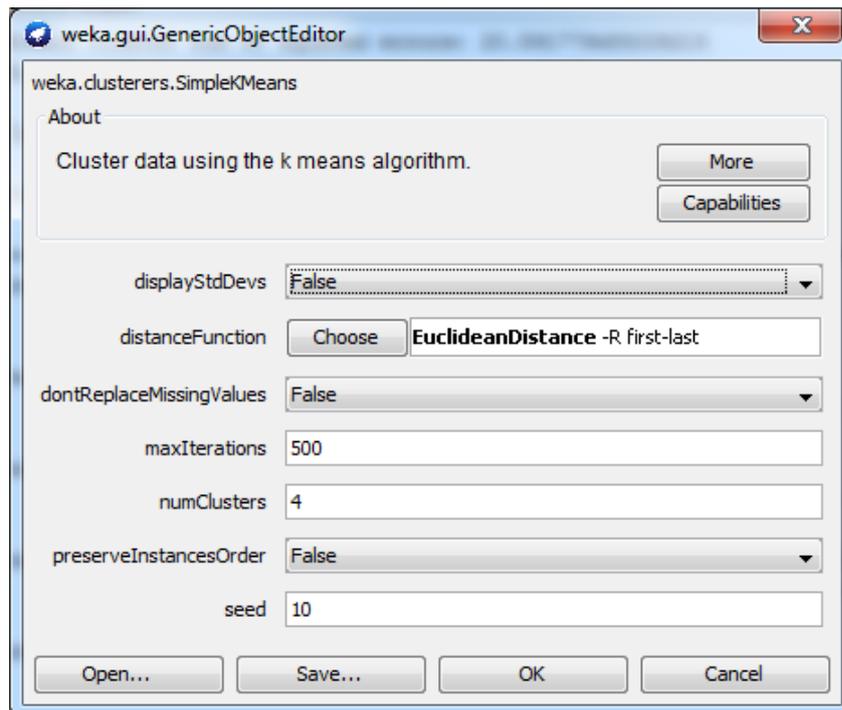
Fonte: Excel. Adaptado pelo autor.

### 5.1.2. Aplicação do algoritmo *k-means*

Conforme já abordado anteriormente, as informações analisadas compreendem um total de 111 jogadores, os quais se conectaram ao jogo obrigatoriamente nos três dias que foram extraídas as “fotografias” dos dados. A aplicação do algoritmo *k-means* sobre o arquivo gerado no processo de extração de dados foi feita através da ferramenta **WEKA 3.6**.

Para o primeiro teste, foi configurado o algoritmo *SimpleKMeans* com a definição de 4 *clusters* e um máximo de 500 iterações. Como função de distância para comparação de instâncias foi utilizado a função de distância Euclidiana. Foram considerados os atributos *experience*, *level*, *var\_level*, *num\_items*, *num\_gold*, *num\_kills* e *num\_friends*.

Figura 21 – Configuração do *k-means* dentro do WEKA 3.6.



Fonte: WEKA. Adaptado pelo autor.

A quantidade de *clusters* define quantos centroides serão criados, e em quantos grupos as informações serão reunidas. Foram definidos quatro *clusters* devido o objetivo de verificar se os agrupamentos gerados pelo *k-means* possuem relação com os perfis de jogadores propostos. A figura 22 mostra como foram criados os *clusters* neste teste.

Figura 22 – Clusters criados pelo algoritmo *k-means*.

Cluster centroids:

Attribute	Full Data (111)	Cluster#			
		0 (33)	1 (17)	2 (41)	3 (20)
experience	4271684.045	2438688.9091	44076.8235	9030609.5854	1133794.8
level	185.955	125.6364	52	271.6098	223.75
var_level	8.1532	15.2424	0.5294	8.7805	1.65
num_items	32.3964	28.4545	10.6471	52.4634	16.25
num_gold	127748.5135	57852.4848	229807.8824	196842.7073	14683.4
num_kills	1.1802	0.0909	0	0.122	6.15
num_friends	4.8288	1.4848	15.8235	4.2927	2.1

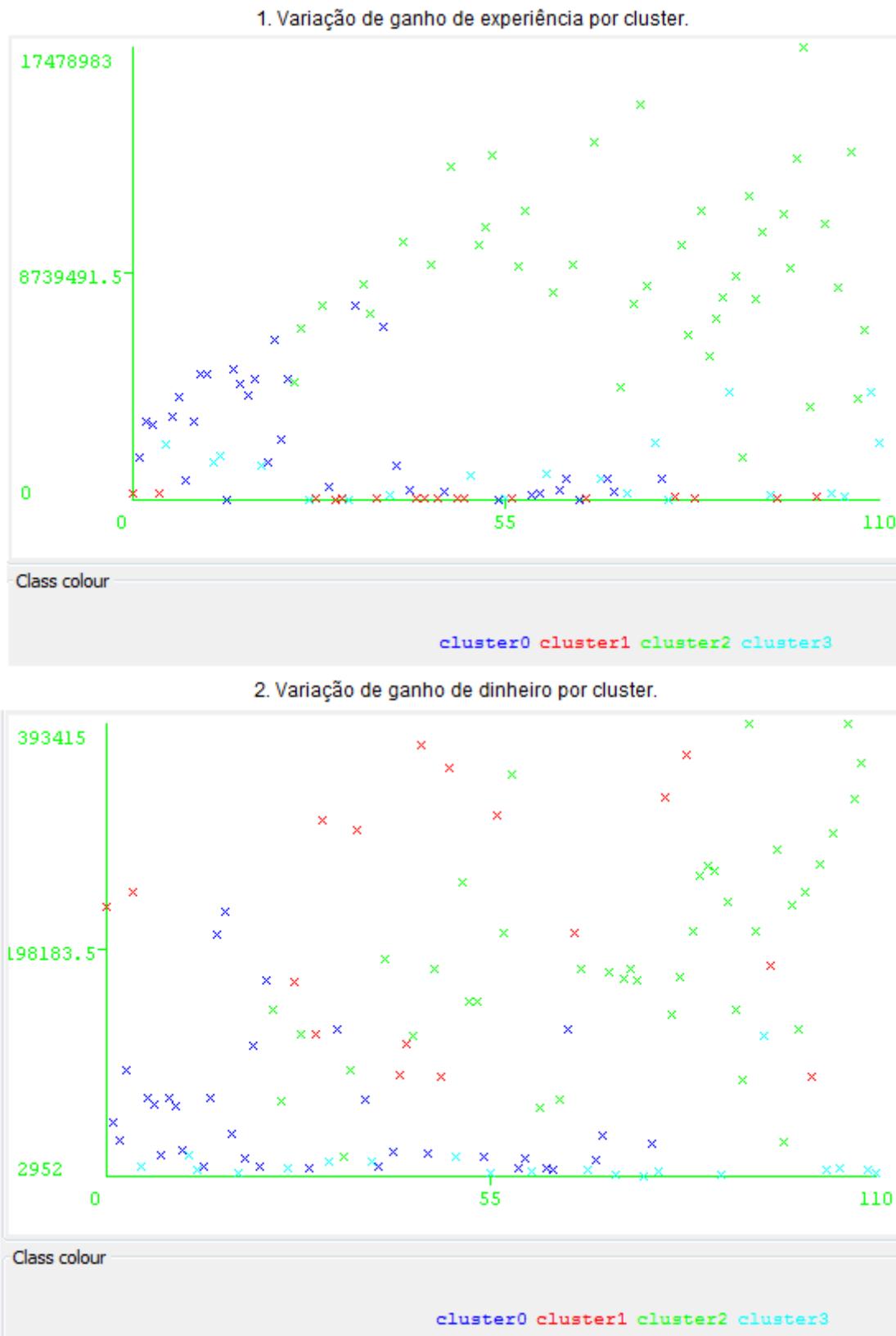
Fonte: WEKA. Adaptado pelo autor.

Analisando os valores para cada um dos itens, foi possível atribuir os *clusters* aos perfis de jogadores da seguinte forma:

- *Cluster 0* – jogadores deste *cluster* tem um valor de *level* relativamente baixo, o mesmo para a variação de *itens* e de *dinheiro*. Possui um número de *amigos* baixo, e um número de *assassinatos* quase inexistente. Estas características poderiam colocar estes jogadores como *Explorers*. Porém, jogadores destes *clusters* tiveram uma variação de *experiência* relativamente alta, e a maior variação de *níveis* entre todos os *clusters*, características estas de *Achievers*.
- *Cluster 1* – o baixo ganho de *experiência* e *nível*, a quantidade nula de *assassinatos*, associados ao grande número de *amigos*, um número razoável de variação de *itens* e o maior ganho de *gold* entre *clusters* indica que este jogador se encaixa no perfil de *Socializers*. As características de grande número de *amigos* e de ganho alto de *dinheiro* ajudam nessa definição.
- *Cluster 2* – o maior ganho de *experiência* e de *nível* entre os *clusters*, somando a uma grande aquisição de *dinheiro* e de *itens*, e uma razoável variação de *níveis* coloca estes jogadores facilmente como *Achievers*, que buscam o ganho de *itens*, *dinheiro* e pontos de *experiência*.
- *Cluster 3* – a alta média de número de *assassinatos* e o alto *nível* dos jogadores, associado aos baixos ganhos de *itens* e *dinheiro*, e o também baixo número de *amigos* ajudam a definir este *cluster* como jogadores da categoria *Killers*, que buscam o confronto com outros jogadores antes de outras ações dentro do jogo.

Conforme a figura 23, observando o gráfico que traz a variação de experiência por jogador, foi verificado que a grande maioria dos ganhos de experiência consideráveis são relativos aos jogadores do *Cluster 2*, sendo possível indicar os membros deste *cluster* como *Achievers*. Também na figura 23, agora observando o gráfico que mostra a quantidade de *dinheiro* adquirido por jogador, vemos que as maiores médias de aquisição de *dinheiro* estão entre o *Cluster 1* (*Socializer*) e o *Cluster 2* (*Achievers*). Porém, ainda analisando a figura 22, é possível verificar a existência de jogadores do *Cluster 0* em posições consideráveis tanto em ganho de *dinheiro* quanto em ganho de *experiência*.

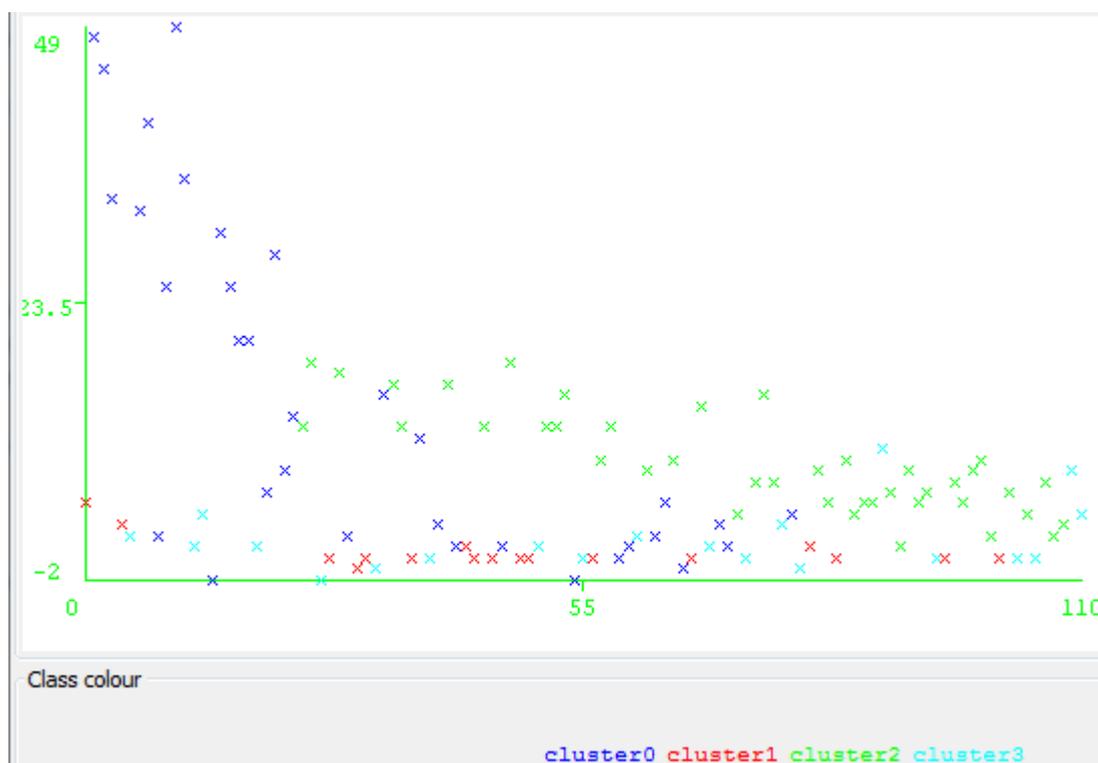
Figura 23 – Gráficos de (1) variação de ganho de experiência por jogador e (2) variação de ganho de dinheiro por jogador.



Fonte: WEKA. Adaptado pelo autor.

Sendo assim, foi possível observar certa desordem nas informações do *Cluster 0*. Pelos resultados apresentados neste *cluster*, os jogadores neste podem se enquadrar tanto como *Explorers* (devido à menor aquisição de *itens* e *dinheiro*, e ao menor *nível*) quanto como *Achievers* (em alusão a alta variação de *níveis* e quantidade significativa de ganho de *experiência*). Como mostra a figura 24, a variação de nível por jogador é maior no *Cluster 0* dentre todos os *clusters*.

Figura 24 – Gráfico de variação de nível (y) por jogador.

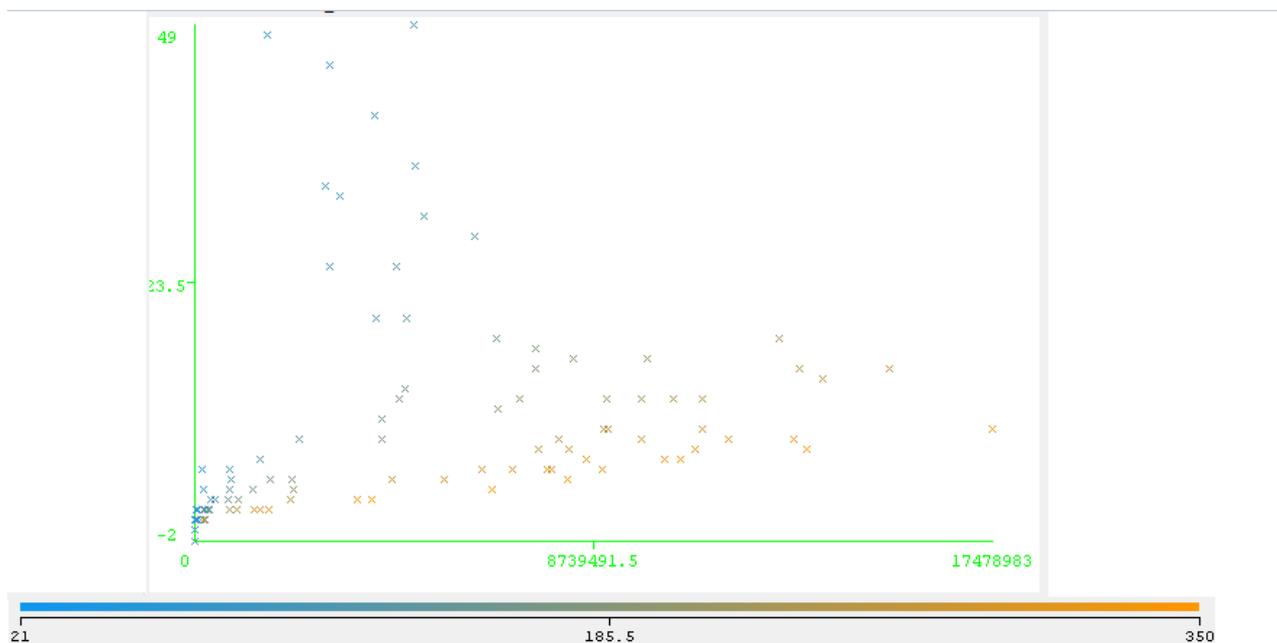


Fonte: WEKA. Adaptado pelo autor.

Aprofundando a análise sobre o problema existente no *Cluster 0*, temos a figura 25, que mostra a relação entre a média de *experiência* (em X), a variação de *nível* (em Y), e o *nível* do jogador (com a variação de cor). Sendo assim, foi possível observar que o ganho de *experiência* não é diretamente proporcional ao ganho de *nível*, sendo que o *nível* que o jogador se encontra também impacta nesta variação. Foi observado que, a medida que o jogador está em *nível* mais baixo (pontos azulados no gráfico), a variação de *nível* é maior mesmo com um ganho de *experiência* menor. Jogadores de níveis mais altos (pontos alaranjados no gráfico) tem variações de nível menor mesmo com ganhos altos de *experiência*. Constata-se assim que, a medida que o *nível* do jogador é mais alto, mais *experiência* ele

necessita para alcançar o próximo *nível*, o que justifica variações de níveis menores em níveis mais altos.

**Figura 25 – Média de experiência (x) por variação de nível (y), com nível do jogador na cor.**



Fonte: WEKA. Adaptado pelo autor.

Para tentar corrigir este problema e obter um resultado mais coerente, foram realizados outros testes (figura 26): (1) aplicação do *k-means* ignorando *experiência*, (2) aplicação no *k-means* ignorando *level* e variação de *level* e (3) aplicação do *k-means* ignorando *level*. Como observado na figura 26, houve um aumento de incoerência nas informações com a exclusão de campos, especialmente em relação aos perfis de *Killers* e *Explorers*, concluindo que todos os campos utilizados até o momento são de valia para a análise.

**Figura 26 – Outros testes realizados.**

1. aplicação do k-means ignorando experiência

Cluster centroids:

Attribute	Full Data (111)	Cluster#			
		0 (38)	1 (17)	2 (20)	3 (36)
level	185.955	181.9474	52	129.5	284.8056
var_level	8.1532	2.0526	0.5294	26.65	7.9167
num_items	32.3964	12.1316	10.6471	63.15	46.9722
num_gold	127748.5135	20187.6316	229807.8824	94263.55	211693.0556
num_kills	1.1802	3.3158	0	0.05	0.1111
num_friends	4.8288	1.8421	15.8235	2.3	4.1944

2. aplicação no k-means ignorando level e variação de level

Cluster centroids:

Attribute	Full Data (111)	Cluster#			
		0 (10)	1 (8)	2 (50)	3 (43)
experience	4271684.045	59023.7	986449.375	8208878.56	1284445.7907
num_items	32.3964	8	15	55.2	14.7907
num_gold	127748.5135	155751.2	342769.75	175240.66	26008.8837
num_kills	1.1802	0	0	0.1	2.9302
num_friends	4.8288	16.6	13.875	3.66	1.7674

3. aplicação no k-means ignorando level

Cluster centroids:

Attribute	Full Data (111)	Cluster#			
		0 (10)	1 (8)	2 (56)	3 (37)
experience	4271684.045	59023.7	986449.375	7728180.625	889107.6216
var_level	8.1532	0.8	0.75	14.7143	1.8108
num_items	32.3964	8	15	52.75	11.9459
num_gold	127748.5135	155751.2	342769.75	163965.2143	18874.6757
num_kills	1.1802	0	0	0.0893	3.4054
num_friends	4.8288	16.6	13.875	3.375	1.8919

Fonte: WEKA. Adaptado pelo autor.

Como solução para o problema apresentado na classificação do *Cluster 0*, foi aplicado um coeficiente sobre o valor de *experiência*, com o objetivo de aumentar o ganho de *experiência* em jogadores que se enquadrariam como *Achievers*, porém são de níveis mais baixos. Após testes realizados e análises sobre os resultados da comparação entre ganho de *experiência* e *nível* do jogador, foi elaborada uma regra que considera o *nível* do jogador e a *variação de nível* deste, de forma que, caso o jogador seja de um *nível* mais baixo mas tenha um ganho de níveis significativo, ele obtenha mais valorização em seu ganho de *experiência*. Abaixo segue a regra desenvolvida.

Tabela 4 – Regra para aplicação de coeficiente sobre experiência ganha.

Passo	Operação
1	SE level < 100 E var_level > 10 ENTÃO
2	exp_factor = experience * 2
3	SE level >= 100 E level < 190 E var_level > 10 ENTÃO
4	exp_factor = experience * 1,5
5	SE NÃO
6	exp_factor = experience

Fonte: O autor.

A partir deste momento, as análises pararam de considerar a *experiência* original do jogador, e passaram a considerar sua experiência calculada a partir da regra, e aplicada ao campo *exp\_factor*. A figura 27 mostra a análise aplicada a essa nova estrutura de dados.

Figura 27 – Clusters obtidos com nova estrutura de dados.

Cluster centroids:

Attribute	Full Data (111)	Cluster#			
		0 (18)	1 (16)	2 (57)	3 (20)
level	185.955	127.2778	52.125	228.7895	223.75
var_level	8.1532	1.8889	0.5625	14.5439	1.65
num_items	32.3964	7.2222	10.5	52.1579	16.25
num_gold	127748.5135	28926.7778	236921.6875	167982.2456	14683.4
num_kills	1.1802	0.1667	0	0.0877	6.15
num_friends	4.8288	2	16.3125	3.4561	2.1
exp_factor	4758756.8964	568514.7778	46070.375	8676767.1491	1133794.8

Fonte: WEKA. Adaptado pelo autor.

Segue uma análise dos *clusters* criados por esta nova análise:

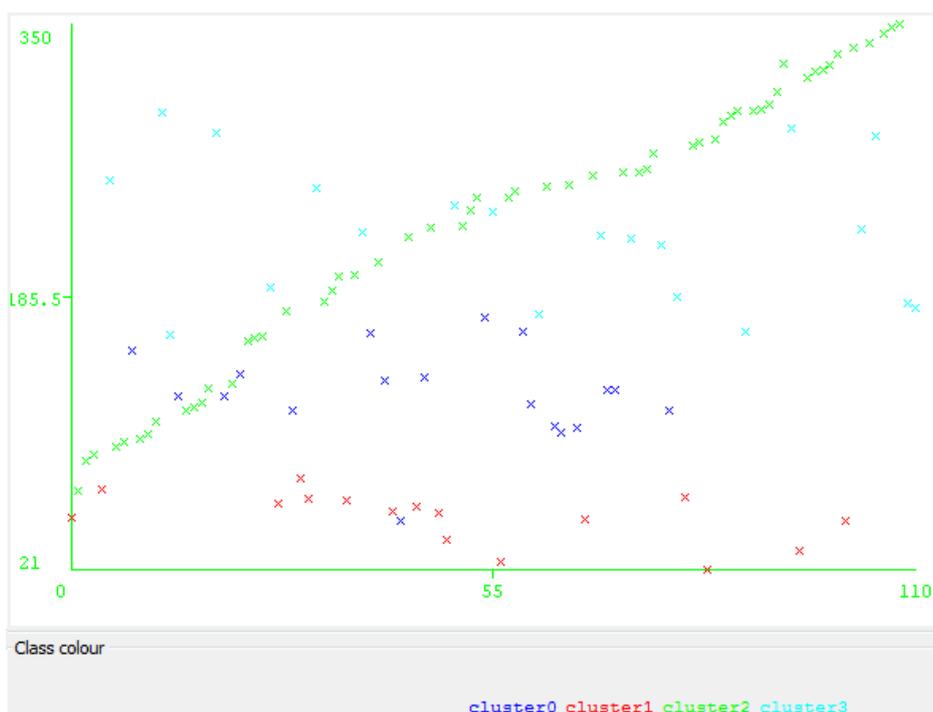
- *Cluster 0* – Podemos observar os jogadores deste *cluster* com uma quantidade de ganho de *experiência* bastante abaixo da média geral, assim como variações de *level*. Vemos também uma aparente pequena preocupação com ganho de *itens* e *dinheiro*. Estas características facilmente colocam estes jogadores como *Explorers*.
- *Cluster 1* – Esta nova regra, considerando o *coeficiente de experiência* trouxe pouca alteração para o *Cluster 1*. Suas características de grande ganho de

*dinheiro* e grande número de *amigos*, aliados a valores baixos nos demais indicadores, colocam esses jogadores como fortes candidatos a serem *Socializers*.

- *Cluster 2* – Em adição ao grande ganho de *experiência*, *itens* e *dinheiros* já existentes neste *cluster*, observou-se agora também uma elevação na *variação de níveis*, provavelmente referentes a jogadores que saíram do *Cluster 0* para integrar agora este *cluster*. As características dos jogadores deste grupo coincidem com as características definidas para *Explorers*.
- *Cluster 3* – Comparando com a primeira análise apresentada, não ocorreu nenhuma alteração neste *cluster*. Suas características, novamente, coincidem com o que se espera de um jogador categorizado como *Killer*.

A aplicação deste coeficiente mostrou uma melhor definição na separação dos *clusters*, e uma relação muito mais precisa comparando estes com as categorias de jogadores propostas. Observando a figura 28, que traz uma análise de jogador por *nível*, podemos observar que agora o *Cluster 2 (Achievers)* contém jogadores de uma abrangência maior de níveis, ao contrário da análise anterior, onde *Achievers* eram somente encontrados em níveis mais altos.

Figura 28 – Gráfico de jogador (x) por nível (y).



Ainda na figura 28, podemos observar também que *Killers* (*Cluster 3*) normalmente possuem *nível* acima da média. A possibilidade de eles terem sido *Achievers* durante algum tempo é grande, ao menos até eles atingirem níveis nos quais eles se sintam confortáveis a praticarem ações de *Killers*. A média de níveis de *Explorers* e *Socializers* é baixa em relação aos demais, já que suas atribuições não exigem que eles possuam níveis altos.

### 5.1.3. Avaliação dos resultados

Conforme foi observado nos testes, os resultados apresentados pela aplicação do algoritmo *k-means* sobre as informações extraídas da base de dados do Tibia são coerentes quanto aos perfis propostos. Porém algumas considerações devem ser levadas em conta.

Primeiramente, não podemos afirmar com absoluta certeza que a classificação do perfil de *Explorers* foi feita corretamente a partir da análise dada sem existir uma análise sobre as ações do jogador dentro do jogo. Observando apenas seus status, é impossível afirmar que o jogador atuou realmente como um *Explorer*, efetivamente explorando o ambiente do jogo e realizando *quests*, ou se é apenas um jogador sem experiência de jogo e conseqüentemente com evolução mais lenta. Outro problema para esta análise sobre *Explorers* é a inexistência das informações de *quests* na base de dados, o que traz ainda mais dúvidas para os resultados referentes a este perfil de jogador. Conseqüentemente, muitos dos jogadores considerados nesta categoria podem pertencer a outras categorias, porém não possuem experiência de jogo para que isso se reflita em seus dados.

Por outro lado, as análises sobre os outros perfis de jogadores aparentam ser precisas, mesmo sem termos as ações que o jogador executou em sua totalidade. Pegando inicialmente os *Achievers*, cujo perfil determina a busca por evolução de *nível*, *itens* e *dinheiro*. Em uma análise de sua evolução observando apenas as “fotografias” das informações do jogador, é possível definir se ele se enquadra neste perfil a partir da variação de seus indicadores característicos. Uma análise das ações do jogador seria mais precisa, porém para este caso não é determinante para o sucesso dos resultados.

Da mesma forma, *Killers* e *Socializers* também conseguem ser bem definidos apenas pelas bases de dados analisadas. Quanto aos primeiros, o número de vezes que um jogador mata outro personagem é o fator determinante para sua definição no perfil. Já para os *Socializers*, o número de *amigos* e o grande ganho de *dinheiro* com o ganho de *experiência* ausente facilitam sua identificação. Ainda sobre os *Socializers*, a existência de um histórico de mensagens ajudaria ainda mais na precisão das informações.

Por fim, os indicadores chaves para uma análise da base de dados estática podem variar de acordo com o perfil que se deseja categorizar. Porém, se destacam a variação de *nível*, o ganho de *experiência*, o ganho de *dinheiro* e o número de *kills*, estas informações variando de forma abrupta de um perfil de jogador para o outro.

## 5.2. Análise sobre base de dados dinâmica.

Para a análise do algoritmo *k-means* sobre uma base de dados dinâmica foi utilizado como fonte um arquivo gerado pelo **Zereal Agent-based Simulator** (Tveit, 2003). Este software tem como intuito simular as ações de diferentes categorias de jogador utilizando um ambiente baseado em agentes, programado em *Python*. A simulação contém agentes personagens (que simulam os jogadores) e agentes monstros (criaturas de comportamento hostil e alvos de ataque dos jogadores).

**Figura 29 – Base de dados gerada pelo Zereal Agent-based Simulator.**

47	1	08/12/2013 18:52	1000498	walk	(99,30)	(99,31)	PlainAgent
47	1	08/12/2013 18:52	1000499	attack	(60,89)	(60,88)	Killer
48	1	08/12/2013 18:52	1000425	walk	(50,25)	(51,26)	MarkovKiller
48	1	08/12/2013 18:52	1000426	walk	(74,75)	(74,74)	PlainAgent
48	1	08/12/2013 18:52	1000428	walk	(105,112)	(106,112)	PlainAgent
48	1	08/12/2013 18:52	1000429	pickuppotion	(51,3)	(51,4)	MarkovKiller
48	1	08/12/2013 18:52	1000431	walk	(99,48)	(100,47)	Killer
48	1	08/12/2013 18:52	1000432	walk	(108,43)	(109,44)	PlainAgent

Fonte: Excel. Adaptado pelo autor.

A base adquirida para testes foi obtida em uma versão do software que considera três perfis de jogadores (*Killer*, *MarkovKiller* e *PlainAgent*), sendo que cada agente pode tomar uma das sete ações a seguir: *EnterWorld* (acesso do jogador ao mundo), *LeaveWorld* (desconexão do jogador do mundo), *Attack* (ação

de atacar monstro), *PickFood* (ação de pegar item do tipo comida), *PickPotion* (ação de pegar item do tipo poção), *PickKey* (ação de pegar item do tipo chave) e *Walk* (ação de se movimentar pelo mapa). Cada perfil de jogador é designado a ter um padrão de comportamento:

- *Killer* – coloca como maior prioridade encontrar e atacar monstros, desta forma, terá um grande índice de ações de ataque (*Attack*). Terá uma quantidade não tão alta de ações *Walk*, *PickFood* e *PickPotion*, e terá frequência baixa de ações de *PickKey* e *LeaveWorld*.
- *MarkovKiller* – simula um jogador mais experiente, que analisa seu estado antes de tomar uma ação. Terá a frequência de ações de *Walk* e *Attack*, *PickKey* e *LeaveWorld* médias, e visará as ações de *PickFood* e *PickPotion*.
- *PlainAgent* – foca em encontrar chaves e sair do mundo, engajando pouco em batalhas. Terá uma grande frequência de ações *Walk* e *PickUpKey*, raramente terá ações de *Attack*, *PickFood* e *PickPotion*, e sempre buscará a saída do mundo.

Comparando os perfis definidos pelo Zereal com os propostos por Bartle, podemos verificar a ausência do tipo *Socializer*, dada a inexistência de ações de interação com outros jogadores. Porém, é possível estabelecer relações dos demais perfis de Bartle com os propostos pelo Zereal, da seguinte forma:

- *Achiever* – é possível relacioná-lo com o *MarkovKiller*, pois representa um jogador mais experiente, que tem por objetivos engajar em batalhas de forma mais controlada e adquirir itens.
- *Explorer* – sua melhor correspondência são os *PlainAgents*, que caminham bastante pelo mapa, buscando chaves para explorar novos locais.
- *Killer* – ignorando o fato de Bartle ter proposto este perfil com o intuito de identificar jogadores que atacam outros jogadores, sua relação fica bastante clara, pois ambos sempre priorizam o engaje em batalhas.

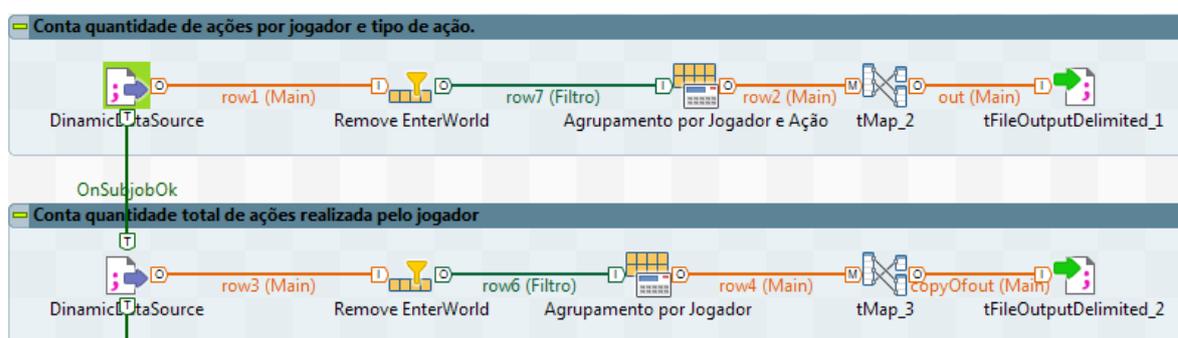
Para fins de análise, a ação *EnterWorld* foi ignorada, sendo ela comum entre todos os jogadores. O arquivo utilizado foi gerado considerando 100 jogadores e apenas um mundo. A análise sobre as informações dinâmicas é composta de extração e tratamento das informações, aplicação do algoritmo *k-means* e avaliação do resultado, sendo as etapas descritas a seguir.

### 5.2.1. Extração e transformação das informações

Da forma como geradas, as informações fornecidas pelo Zereal não eram capazes de, ao aplicadas ao *k-means*, fornecer resultados coerentes. Sendo assim, foi necessário um trabalho sobre este arquivo base para que fosse formatado em uma estrutura capaz de ser analisado com sucesso.

Antes de conseguir definir o perfil de ações de cada jogador, e encaixá-los nas três categorias propostas, foi necessário saber o quanto cada ação impacta no total de ações que o jogador executa, definindo assim a porcentagem que cada uma das ações representa em um todo. Para isto, inicialmente foram realizadas duas etapas: (1) soma do total de ações realizadas, agrupadas por ID do jogador e tipo de ação, e (2) soma de ações agrupando por ID de jogador. A figura 30 mostra estas etapas.

**Figura 30 – Etapas de (1) agrupamento de ações por jogador e tipo de ação e (2) agrupamento de ações por jogador.**



Fonte: Talend. Adaptado pelo autor.

No próximo passo, as informações de saída das duas etapas anteriores foram unificadas, sendo que, para cada ação, é feita a divisão da quantidade de vezes que a ação foi executada pelo total de ações executadas pelo jogador, assim obtendo o percentual que cada ação representa sobre o total de ações do jogador, em números inteiros. A saída deste processo é um arquivo que tem como colunas o *id do jogador*, o *tipo de ação* e o *percentual desta ação*.

Porém, para ser analisada com sucesso pelo algoritmo foi necessário abrir cada uma das ações em uma coluna diferente. Neste processo, também foi adicionado o *tipo de jogador* à coluna, para facilitar a análise e validação da aplicação do algoritmo. A saída do processo de extração e transformação foi um

arquivo de formato CSV, separado por vírgulas. A figura 31 mostra como o arquivo ficou estruturado.

**Figura 31 – Arquivo pronto para ser aplicado ao algoritmo *k-means*.**

IdJogador	TipoJogador	leavetheworld	walk	pickuppotion	attack	pickupfood	pickupkey
1000401	PlainAgent	1	91	0	0	1	7
1000402	Killer	0	42	1	57	0	0
1000403	MarkovKiller	0	85	4	6	4	1
1000404	MarkovKiller	0	91	2	3	4	0
1000405	MarkovKiller	0	90	2	5	3	0
1000406	PlainAgent	1	94	0	0	0	5
1000407	PlainAgent	1	95	0	0	0	4

Fonte: Excel. Adaptado pelo autor.

Segue a seguir a descrição de cada uma das colunas contidas no arquivo de saída:

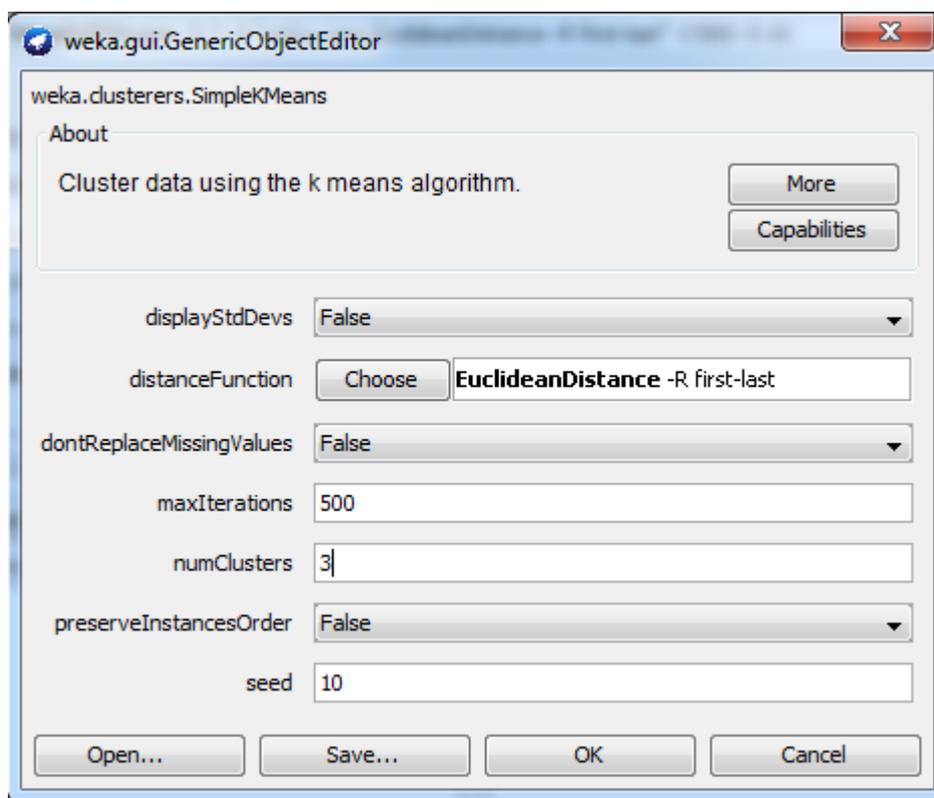
- *idJogador*: ID de identificação do jogador
- *TipoJogador*: tipo de jogador definido pelo Zereal, podendo ser *PlainAgent*, *Killer* ou *MarkovKiller*, relacionados respectivamente com *Explorers*, *Killers* e *Achievers*.
- *leavetheworld*: percentual de ações do tipo *leavetheworld* sobre o total de ações realizadas pelo personagem.
- *walk*: percentual de ações do tipo *walk* sobre o total de ações realizadas pelo personagem.
- *pickuppotion*: percentual de ações do tipo *pickuppotion* sobre o total de ações realizadas pelo personagem.
- *attack*: percentual de ações do tipo *attack* sobre o total de ações realizadas pelo personagem.
- *pickfood*: percentual de ações do tipo *pickfood* sobre o total de ações realizadas pelo personagem.
- *pickupkey*: percentual de ações do tipo *pickupkey* sobre o total de ações realizadas pelo personagem.

### 5.2.2. Aplicação do algoritmo k-means

Na base estática, as informações analisadas compreendem os registros de simulação de 100 jogadores. A aplicação do algoritmo *k-means* sobre o arquivo gerado no processo de extração de dados foi feita através da ferramenta WEKA 3.6.

Para o primeiro teste, foi configurado o algoritmo *SimpleKMeans* com a definição de 3 *clusters* e um máximo de 500 iterações. Como função de distância para comparação de instâncias foi utilizado a função de distância Euclidiana. Foram considerados os atributos *walk*, *leavetheworld*, *pickuppotion*, *pickupkey*, *pickupfood* e *attack*.

Figura 32 – Configuração do *k-means* dentro do WEKA 3.6.



Fonte: WEKA. Adaptado pelo autor.

Conforme definido, não pode ser considerado o perfil de *Socializers* na análise de informações dinâmicas, devido a inexistência de ações que caracterizem este tipo de jogador na fonte de dados. Desta forma, os três *clusters* definidos irão verificar se as informações contidas neles coincidem com os perfis de *Achievers*, *Explorers* e *Killers*. A figura 33 mostra como foram criados os *clusters* neste teste.

**Figura 33 – Clusters criados em teste sobre base de dados dinâmicos.**

Cluster centroids:

Attribute	Full Data (100)	Cluster#		
		0 (46)	1 (28)	2 (26)
leavetheworld	0.33	0	1	0.1923
walk	79.52	91.3913	92.1786	44.8846
pickuppotion	1.18	1.7609	0.75	0.6154
attack	15.48	2.8043	0.5	54.0385
pickupfood	1.57	2.6522	1.0357	0.2308
pickupkey	1.9	1.3696	4.5357	0

Fonte: WEKA. Adaptado pelo autor.

Segue a definição dos *clusters* encontrados nesta primeira análise.

- *Cluster 0*: Possui um grande percentual de ações de *walk*, e percentuais acima da média em relação a *pickupfood* e *pickuppotion*. Também possui uma ligeira ocorrência de ações de *attack*, enquanto a ação de *pickupkey* está abaixo da média. Por este padrão, considerando a existência (mesmo que pequena) de ações de *ataque*, e a maior incidência de ações de *recolhimento de poções* e *comida*, podemos pressupor que este jogador se encaixa na categoria de *Achievers*.
- *Cluster 1*: Possui os maiores percentuais de ação de *walk* e *pickupkey* dentre os *clusters* criados. Além disso, possui frequências baixas nas ações de *ataque*, *recolhimento de poções* e *comida* (*attack*, *pickuppotion* e *pickupfood*, respectivamente). A partir destas características podemos concluir que jogadores deste *Cluster* fazem parte da categoria de *Explorers*.
- *Cluster 2*: Possui a menor incidência de ações de *movimento*, e a maior incidência de ações de *ataque*. Apenas por estas características já podemos definir que os jogadores presentes neste *cluster* se enquadram como *Killers*.

Em uma primeira análise, as atribuições parecem bastante coerentes. Para validar, foi realizado mais um teste. O WEKA permite a atribuição de um campo da análise para nomenclatura dos *clusters*. Sendo assim, foi feita esta atribuição através do tipo de jogador. Os resultados apresentados mostraram que o *Cluster 2* representa perfeitamente os jogadores do tipo *Killers*. Porém, houve certa discordância entre as informações dos *Clusters 0* e *1*, conforme mostra a figura 34.

**Figura 34 – Atribuição de *clusters* a tipos de jogador.**

```

Class attribute: TipoJogador
Classes to Clusters:

  0  1  2  <-- assigned to cluster
11 23  0  | PlainAgent
  0  0 26  | Killer
35  5  0  | MarkovKiller

Cluster 0 <-- MarkovKiller
Cluster 1 <-- PlainAgent
Cluster 2 <-- Killer

Incorrectly clustered instances :      16.0      16      %

```

Fonte: WEKA. Adaptado pelo autor.

Para corrigir este problema foi realizado um novo teste, desta vez ignorando a ação *leavetheworld*. O algoritmo foi configurado da mesma forma, com 3 *clusters*, máximo de 500 iterações e utilizando a função de distância Euclidiana. A figura 35 traz os *clusters* criados neste teste.

**Figura 35 – *Clusters* resultantes de análise desconsiderando ação *leaveworld*.**

Attribute	Full Data (100)	Cluster#		
		0 (40)	1 (26)	2 (34)
walk	79.52	90.425	44.8846	93.1765
pickuppotion	1.18	2.275	0.6154	0.3235
attack	15.48	3.575	54.0385	0
pickupfood	1.57	3.275	0.2308	0.5882
pickupkey	1.9	0.325	0	5.2059

Fonte: WEKA. Adaptado pelo autor.

Conforme podemos observar analisando a figura 35, aparentemente os resultados parecem estar mais próximos da realidade:

- *Cluster 0*: Possui percentual de *ataque*, mesmo que baixo, além de ter os maiores percentuais para as ações de *pickupfood* e *pickuppotion*. Estas características, em comparação com os outros *clusters*, indicam mais precisão ao afirmar que jogadores deste *cluster* são *Achievers*.
- *Cluster 1*: facilmente identificado como *Killers*, devido a alta incidência de ações de *ataque*, e a menor preocupação com o recolhimento de *itens*.

- *Cluster 2*: seus percentuais de ações mais relevantes são de *movimento* e *recolhimento de chaves*, com percentuais bastante baixos nas ações de *recolhimento de outros itens*, e percentual nulo em ações de *ataque*. Estas características, conforme definimos, coincidem com o perfil de *Explorers*.

Figura 36 – Ações de movimento (x) por ações de ataque (y).



Fonte: WEKA. Adaptado pelo autor.

A figura 36 apresenta um gráfico que mostra a relação entre o percentual de ações de *movimento* (em x) pelo percentual de ações de *ataque* (em y). Podemos assim verificar que os personagens do *Cluster 1 (Killers)* possuem os maiores índices de ações de *ataque*, e os menores de ações de *movimento*. Os pertencentes ao *Cluster 0 (Achievers)* possuem um percentual pequeno de *ataques*, com grande quantidade de ações de *movimento*, enquanto o *Cluster 2* traz os maiores índices de *movimento* e informações quase nulas para *ataque*. Um fato preocupante sobre este gráfico é a perfeita padronização das informações, com comportamentos de ações bem definidos, o que não seria encontrado em uma análise sobre uma fonte de dados de um jogo real.

Fazendo novamente a atribuição das categorias definidas aos *clusters*, através do WEKA, foi possível verificar que todos os resultados de *clusters* coincidem com apenas um perfil definido, sendo assim a classificação feita a partir destes testes um sucesso.

**Figura 37 – Atribuição de *clusters* a tipos de jogador após ajuste na análise.**

```

Class attribute: TipoJogador
Classes to Clusters:

  0  1  2  <-- assigned to cluster
  0  0 34 | PlainAgent
  0 26  0 | Killer
 40  0  0 | MarkovKiller

Cluster 0 <-- MarkovKiller
Cluster 1 <-- Killer
Cluster 2 <-- PlainAgent

Incorrectly clustered instances :      0.0      0      %

```

Fonte: WEKA. Adaptado pelo autor.

### 5.2.3. Avaliação dos resultados

Como verificado durante os testes realizados, o algoritmo foi capaz de distinguir com sucesso as categorias de jogadores, e atribuir estas categorias aos *clusters* gerados. Ele também encontrou bastante facilidade em distinguir as informações relevantes, sendo o número de iterações necessárias para se atingir o resultado final bastante baixo. É possível concluir que o algoritmo *k-means* pode ser aplicado na tarefa de classificação de perfis de jogadores em análise de bases de dados dinâmicas. Porém, a pesar do sucesso dos resultados, algumas ressalvas são necessárias.

Primeiramente, para as informações dinâmicas foi utilizado um simulador baseado em agentes como fonte. Sendo a base para análise gerada a partir desta fonte, podemos esperar informações melhores definidas e com pouca sujeira, o que facilita em muito qualquer análise. É seguro afirmar que, em uma análise com informações reais, a complexidade e a variação de valores não permitiriam resultados tão precisos.

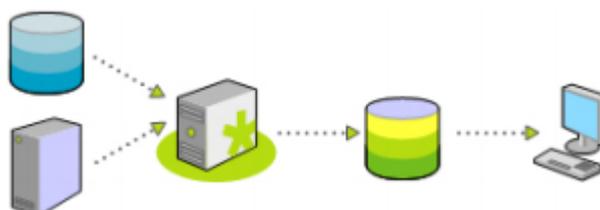
Outro problema encontrado nesta análise foi a carência de ações diferentes. Em MMORPGs reais a escala de ações existente é muito maior do que as sete opções de ações fornecidas pelo resultado da simulação do software Zereal. Além das ações analisadas impossibilitarem a análise do perfil *Socializer*, elas acabam empobrecendo os resultados da análise como um todo.

## 5.3. Ferramentas utilizadas

### 5.3.1. Talend Open Studio for Data Integration

Ferramenta de integração *open source* desenvolvida pela Talend e lançada em outubro de 2006. A ferramenta opera como um gerador de código, produzindo *scripts* de transformação de dados, operando em *Java*. Sua GUI permite acesso a repositórios de metadados e a um designer gráfico.

Figura 38 – Funcionamento básico do Talend.



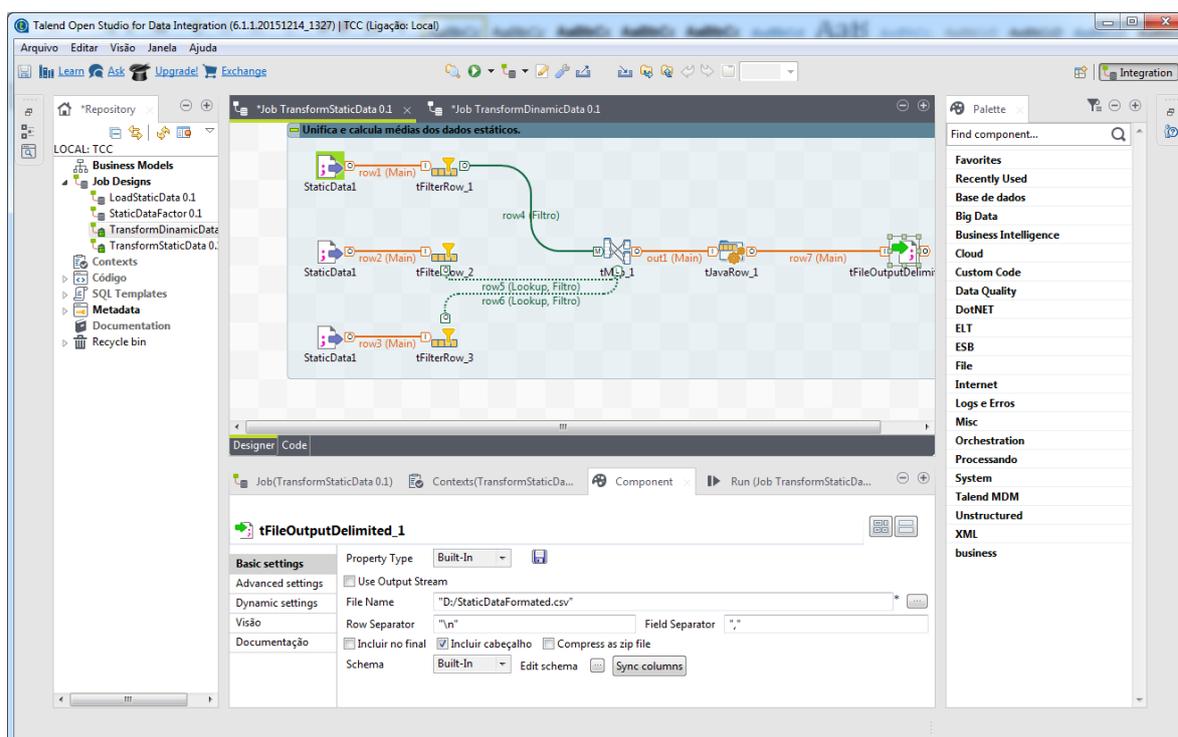
Fonte: Talend Open Studio for Data Integration User Guide

Por padrão o Talend Open Studio permite acesso à dezenas de bases de dados, assim como leitura e escrita em arquivos, acesso ao SAP BW e páginas na Web. Também conta com diversas ferramentas internas que dão suporte às tarefas de tratamento de dados. Dentre os usos do Talend, é possível citar:

- Sincronização ou replicação de bases de dados;
- ETL (*Extract, Transform and Load*) para análise de dados;
- Migração de dados;
- Cargas e transformações complexas de dados;
- Análise de qualidade de dados;
- *Big Data*

A figura 38 ilustra uma das principais funções do *Talend for Data Integration*, com a leitura de distintas fontes de informação (seja um cubo, base de dados, arquivo, página da web), a unificação e tratamento destas informações através do Talend, a geração de uma base de dados tratada, e por fim o consumo destas informações por outros sistemas.

Figura 39 – Imagem da tela inicial do Talend.



Fonte: Talend. Adaptado pelo autor.

### 5.3.2. WEKA 3

WEKA (*Waikato Environment for Knowledge Analysis*) é um software que contém um conjunto de algoritmos de aprendizagem de máquina para tarefas de mineração de dados, desenvolvido pelo grupo de aprendizagem de máquina da Universidade de Waikato. Começou a ser desenvolvido em 1992 e foi lançado no SourceForge em 2000. Hoje, WEKA é reconhecido como um marco em sistemas de mineração de dados e aprendizagem de máquina.

A ferramenta permite aos usuários comparar os resultados de diferentes métodos de aprendizado de máquina sobre uma mesma base de informações. Sua arquitetura modular e extensiva permite processos sofisticados de mineração de dados, a serem construídos tendo como base a grande coleção de algoritmos e ferramentas fornecidas. O formato de dados padrão estabelecido pelo WEKA é o ARFF, sendo um formato de arquivo de entrada específico para a ferramenta e tendo a forma de uma tabela relacional simples. Também possui uma API que permite a integração de novos algoritmos à interface gráfica do WEKA.

Figura 40 – Imagem da tela inicial do WEKA, em sua função *Explorer*.

The screenshot shows the WEKA Explorer window with the 'Visualize' tab selected. The interface includes a menu bar (Preprocess, Classify, Cluster, Associate, Select attributes, Visualize), a toolbar with buttons for file operations and analysis, and a main workspace. The 'Current relation' is 'finaldataset-weka.filters.unsupervised.attribute.Remove-R1' with 100 instances and 7 attributes. The 'Attributes' list includes 'TipoJogador', 'leavetheworld', 'walk', 'pickuppotion', 'attack', 'pickupfood', and 'pickupkey'. The 'walk' attribute is selected, showing statistics: Minimum (40), Maximum (96), Mean (79.52), and StdDev (20.796). The 'Class: pickupkey (Num)' is selected, and a histogram is displayed with bars at 40 (height 26), 68 (height 0), and 96 (height 74). The status bar shows 'Status OK' and a 'Log' button.

Fonte: WEKA. Adaptado pelo autor.

## 6. CONCLUSÕES

A elaboração da pesquisa realizada para o Trabalho de Conclusão I teve como base artigos, livros e outros trabalhos de conclusão, com foco no entendimento de conceitos de mineração de dados, e sua aplicação sobre jogos computacionais, sendo esta pesquisa aprofundada no Trabalho de Conclusão II e servindo como alicerce para o que foi desenvolvido. Com este estudo, verificou-se que o Data Mining surgiu para suprir a carência de extrair informações úteis de bases de dados, em um primeiro momento empresariais, devido ao crescente uso de sistemas computacionais por parte de empresas, gerando assim quantidades grandes de dados, estas não suportadas por algoritmos tradicionais.

No estudo constatou-se também que a utilização de técnicas de mineração de dados sobre informações extraídas de jogos computacionais é relativamente nova, porém de grande importância. Através dessas técnicas, verificou-se um novo leque de possibilidades, indo desde o conhecimento do público que consome os jogos, e o que os retém no jogo, até a extração de informações para auxiliar no desenvolvimento de futuros jogos.

Também foi verificada a importância do tratamento das informações antes de sua aplicação nas técnicas de mineração de dados, tanto para adequar a base de dados de forma a ser analisada com sucesso pelo algoritmo de mineração, quanto para um melhor entendimento dos dados analisados.

O estudo sobre o histórico dos jogos realizado no Trabalho de Conclusão I e aprofundado no Trabalho de Conclusão II, assim como conceitos dentro da categoria de jogos MMORPG, provou-se de grande valia, a fim de ser adquirido maior conhecimento das mecânicas do jogo e das informações que foram trabalhadas na mineração de dados.

A realização do trabalho encontrou uma grande barreira na dificuldade de obtenção de fontes de dados para serem analisadas. Por serem informações críticas para as companhias desenvolvedoras de jogos, estas bases de dados são guardadas com bastante sigilo. Por fim, foi possível obter bases para a análise, mas de forma bastante tardia, o que prejudicou a extensão da análise para mais técnicas de mineração de dados.

O trabalho comprovou a importância da etapa de tratamento das informações das bases de dados a fim de estruturá-las para uma melhor leitura do algoritmo. Os resultados obtidos na análise foram satisfatórios, comprovando a possibilidade do uso do algoritmo *k-means* através da ferramenta WEKA 3.6 para a atribuição de perfis a jogadores em jogos do tipo MMORPG, tanto em análise sobre fontes de dados estáticas quanto em fontes de dados dinâmicas. A característica do algoritmo *k-means* de agrupamento por similaridade foi chave para o sucesso destas análises. Porém, ressalvas devem ser consideradas.

Primeiramente, o uso de informações dinâmicas geradas a partir de um simulador baseado em agentes torna esta análise mais fácil para o algoritmo, já que a base de dados apresenta informações melhor padronizadas e estruturadas, com o número de informações bastante limitada. Os resultados em uma base de dados real não seriam tão acurados.

Outro ponto importante é a subjetividade dos resultados da análise sobre a base de dados estática. Com os valores estáticos de *status* do jogador é possível ter uma noção de seu perfil de comportamento, porém sem saber as ações realizadas por eles dentro do jogo é impossível precisar o sucesso do resultado das análises. Para ter maior segurança em uma análise sobre uma base estática, seriam necessárias bases de dados de mais dias, podendo assim observar melhor o perfil comportamental dos jogadores.

Por fim, o trabalho realizado trouxe um aprofundamento no conhecimento sobre as alternativas de aplicação de mineração de dados e das técnicas em si, assim como a validação da possibilidade do uso destas técnicas em bases de dados de jogos do tipo MMORPG. Além disso, este trabalho pode servir como base para trabalhos futuros serem realizados nesta área, expandindo a análise sobre perfis de jogadores ou abrangendo novas análises, neste ou em outros tipos de jogos.

## REFERÊNCIAS

AMORIM, T. *Conceitos, técnicas, ferramentas e aplicações de Mineração de Dados para gerar conhecimento a partir de bases de dados*. Pernambuco. 2006. Disponível em: <<http://www.cin.ufpe.br/~tg/2006-2/tmas.pdf>>

BARTLE, R. *Hearts, Clubs, Diamonds, Spades: Players who suit MUDs*. MUSE Ltda, 1996. Disponível em: <<http://mud.co.uk/richard/hcdfs.htm>>

BRAMER, M. *Undergraduate Topics in Computer Science - Principles of Data Mining*. Springer, 2007.

BURGES, C. J. C. *A Tutorial on Support Vector Machines for Pattern Recognition*. *Data Mining and Knowledge Discovery*, 2008. Disponível em: <<http://research.microsoft.com/pubs/67119/svmtutorial.pdf>>

CAMILO, C. O.; SILVA, J. C. *Mineração de Dados: Conceitos, Tarefas, Métodos e Ferramentas*. Universidade Federal de Goiás, 2009. Disponível em: <[www.inf.ufg.br/sites/default/files/uploads/relatorios-tecnicos/RT-INF\\_001-09.pdf](http://www.inf.ufg.br/sites/default/files/uploads/relatorios-tecnicos/RT-INF_001-09.pdf)>

CHUNG, Y. et. al. *Game Bot Detection Approach Based on Behavior Analysis and Consideration of Various Play Styles*. Sungkyunkwan University, Seoil University, ETRI, 2013. Disponível em: <<https://etrij.etri.re.kr/etrij/journal/getPublishedPaperFile.do?fileId=SPF-1385635263573>>

CIOS, K. J; et. al. *Data Mining: A knowledge Discovery Approach*. Springer, 2007.

COLTON, S. et. al. *Mining Rules from Player Experience and Activity Data*. Imperial College London, University of York, Rebellion Developments Ltd, 2012. Disponível em: <[http://ccg.doc.gold.ac.uk/papers/gow\\_aiide12.pdf](http://ccg.doc.gold.ac.uk/papers/gow_aiide12.pdf)>

CORNELIUS JUNIOR, R. *Uso de mineração de dados na identificação de alunos com perfil de evasão do ensino superior*. [Trabalho de Conclusão de Curso]. Universidade de Santa Cruz do Sul, 2015.

DANTAS, E. *O Uso da Descoberta de Conhecimento em Base de Dados para Apoiar a Tomada de Decisões*. João Pessoa, 2008. Disponível em: <[http://www.aedb.br/seget/artigos08/331\\_331\\_Artigo\\_SEGET\\_EJDR\\_Versao\\_Final\\_010808.pdf](http://www.aedb.br/seget/artigos08/331_331_Artigo_SEGET_EJDR_Versao_Final_010808.pdf)>

DICERBO, K. E.; KIDWAI, K. *Detecting Player Goals from Game Log Files*. University of Southern Maine, 2012. Disponível em: <[http://educationaldatamining.org/EDM2013/proceedings/paper\\_36.pdf](http://educationaldatamining.org/EDM2013/proceedings/paper_36.pdf)>

ENGUN, H.; IVERSEN, J. V.; REIN, Ø. Zereal: A semi-realistic simulator of Massively Multiplayer Online Games. Norwegian University of Science and Technology, 2003. Disponível em: <<http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.11.8780>>

FAYYAD, U.; PIATETSKY-SHAPIO, G.; SMYTH, P. *From Data Mining to Knowledge Discovery in Databases*. American Association for Artificial Intelligence, 1996.

FRANK, E.; et. al. *The WEKA Data Mining Software: An Update*. University of Waikato, 2009. Disponível em: <[http://www.kdd.org/exploration\\_files/p2V11n1.pdf](http://www.kdd.org/exploration_files/p2V11n1.pdf)>

HAN, J.; PEI, J.; YIN, Y. *Mining Frequent Patterns Without Candidate Generation*. Simon Fraser University, 2000. Disponível em: <[hanj.cs.illinois.edu/pdf/sigmod00.pdf](http://hanj.cs.illinois.edu/pdf/sigmod00.pdf)>

HAN, J.; KAMBER, M; PEI, J. *Data Mining: Concepts and Techniques*. Elsevier, 2012.

HATA, K.; THAWONMAS, R. Aggregation of Action Symbol Sub-sequences for Discovery of Online-Game Player Characteristics Using KeyGraph. Ritsumeikan University, 2005. Disponível em: <<http://dl.ifip.org/db/conf/iwec/icec2005/ThawonmasH05.pdf>>

HO, J.; MATSUMOTO, Y.; THAWONMAS, R. Identification of Player Types in Massively Multiplayer Online Games. Ritsumeikan University, 2003. Disponível em: <<http://www.ice.ci.ritsumei.ac.jp/~ruck/PAP/isaga03.pdf>>

IVERSEN, J.; MATSKIN, Ø. R.; TVEIT, A. *Scalable Agent-Based Simulation of Players in Massively Multiplayer Online Games*. Norwegian University of Science and Technology, 2003. Disponível em: <[https://www.researchgate.net/publication/247444604\\_Scalable\\_Agent-Based\\_Simulation\\_of\\_Players\\_in\\_Massively\\_Multiplayer\\_Online\\_Games](https://www.researchgate.net/publication/247444604_Scalable_Agent-Based_Simulation_of_Players_in_Massively_Multiplayer_Online_Games)>

JAIN, A. K.; DUBES, R. C. *Algorithm for Clustering Data*. Patience Hall, 1988.

KENT, S. L. *The Ultimate History of Video Games*. Three Rivers Press, New York, 2001.

LINDEN, R. *Técnicas de Agrupamento*. Revistas de Sistemas de Informação da FSMA, 2009. Disponível em: <[https://www.researchgate.net/publication/267710538\\_Tecnicas\\_de\\_Agrupamento](https://www.researchgate.net/publication/267710538_Tecnicas_de_Agrupamento)>

MACIEL, T. V., et. al. *Mineração de dados em triagem de risco de saúde*. Revista Brasileira de Computação Aplicada, 2015. Disponível em: <<http://www.upf.br/seer/index.php/rbca/article/viewFile/4651/3331>>

MCCUE, C. *Data Mining and Predictive Analysis - Intelligence Gathering and Crime Analysis*. Elsevier, 2007.

MACQUEEN, J. *Some methods for classification and analysis of multivariate observations*. University of California, 1967. Disponível em: <<http://sci2s.ugr.es/keel/pdf/algorithm/congreso/1967-MacQueen-MSP.pdf>>

NAVEGA, S. *Princípios Essenciais do Data Mining*. Intellwise Research and Training, 2002. Disponível em: <<http://www.intellwise.com/reports/i2002.htm>>

OLSON, D. L.; DELEN, D. *Advanced Data Mining Techniques*. Springer, 2008.

OVERMARS, M. *A Brief History of Computer Games*. Utrecht University, 2012. Disponível em: <[http://www.cs.uu.nl/docs/vakken/b2go/literature/history\\_of\\_games.pdf](http://www.cs.uu.nl/docs/vakken/b2go/literature/history_of_games.pdf)>

QUINLAM, J. R. *Improved Use of Continuous Attributes in C4.5*. University of Sydney, 2006. Disponível em: <<http://home.eng.iastate.edu/~julied/classes/ee547/Handouts/quinlan96a.pdf>>

TAN, P.; STAINBACH, M.; KUMAR, V. *Introdução ao Data Mining*. Ciencia Moderna, 2009.

TALEND. *Talend Open Studio for Data Integration Guide*. 2012. Disponível em: <[http://courses.ischool.berkeley.edu/i290-rmm/s12/TalendOpenStudio\\_DI\\_UG\\_50b\\_EN.pdf](http://courses.ischool.berkeley.edu/i290-rmm/s12/TalendOpenStudio_DI_UG_50b_EN.pdf)>

WEBER, B. G.; MATEAS, M. *A Data Mining Approach to Strategy Prediction*. 2009. Disponível em: <[https://games.soe.ucsc.edu/sites/default/files/cig\\_2009.pdf](https://games.soe.ucsc.edu/sites/default/files/cig_2009.pdf)>

WEBER, B. G.; MATEAS, M.; JHALA, A. *Using Data Mining to Model Player Experience*. UC Santa Cruz, 2011. Disponível em: <[http://alumni.soe.ucsc.edu/~bweber/pubs/submission\\_3\\_epex\\_final.pdf](http://alumni.soe.ucsc.edu/~bweber/pubs/submission_3_epex_final.pdf)>

WEBER, B. G.; et. al. *A Modeling Player Retention in Madden NFL 11*. Association for the Advancement of Artificial Intelligence, 2011. Disponível em: <<http://alumni.soe.ucsc.edu/~bweber/pubs/madden11retention.pdf>>

WEI, Q. *Gamers and the Games They Play*. Worcester Polytechnic Institute, 2010. Disponível em: <[https://www.wpi.edu/Pubs/E-project/Available/E-project-050409-135413/unrestricted/MQP\\_Report\\_Final\\_Edited.pdf](https://www.wpi.edu/Pubs/E-project/Available/E-project-050409-135413/unrestricted/MQP_Report_Final_Edited.pdf)>

WENDER, S. *Data Mining and Machine Learning with Computer Game Logs*. University of Auckland, 2007. Disponível em: <<https://www.cs.auckland.ac.nz/research/gameai/projects/Stefan%20Wender%20Data%20Mining%20Game%20Logs.pdf>>

WITTEN, I. H.; FRANK, E. *Data Mining - Practical Machine Learning Tools and Techniques*. Elsevier, 2005.

WORLD OF WARCRAFT, *World of Warcraft*. Disponível em: <<http://us.battle.net/wow/>>

WU, X.; et. al. *Top 10 algorithms in data mining*. Springer-Verlag London Limited, 2007. Disponível em: <<http://www.cs.uvm.edu/~icdm/algorithms/10Algorithms-08.pdf>>