



Universidade de Brasília – UnB
Faculdade UnB Gama – FGA
Engenharia de Software

Plataforma Web de Análise de Mercado de Jogos para Desenvolvedoras

Autor: João Paulo Busche da Cruz
Orientador: Prof. Matheus Faria

Brasília, DF
2018



João Paulo Busche da Cruz

Plataforma Web de Análise de Mercado de Jogos para Desenvolvedoras

Monografia submetida ao curso de graduação em Engenharia de Software da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Software.

Universidade de Brasília – UnB

Faculdade UnB Gama – FGA

Orientador: Prof. Matheus Faria

Coorientador: Prof. Dr. Edson Alves da Costa Júnior

Brasília, DF

2018

João Paulo Busche da Cruz

Plataforma Web de Análise de Mercado de Jogos para Desenvolvedoras/ João Paulo Busche da Cruz. – Brasília, DF, 2018-
52 p. : il. (algumas color.) ; 30 cm.

Orientador: Prof. Matheus Faria

Trabalho de Conclusão de Curso – Universidade de Brasília – UnB
Faculdade UnB Gama – FGA , 2018.

1. Mercado de Jogos. 2. Análise de Dados. I. Prof. Matheus Faria. II. Universidade de Brasília. III. Faculdade UnB Gama. IV. Plataforma Web de Análise de Mercado de Jogos para Desenvolvedoras

CDU 02:141:005.6

João Paulo Busche da Cruz

Plataforma Web de Análise de Mercado de Jogos para Desenvolvedoras

Monografia submetida ao curso de graduação em Engenharia de Software da Universidade de Brasília, como requisito parcial para obtenção do Título de Bacharel em Engenharia de Software.

Trabalho aprovado. Brasília, DF, 01 de junho de 2013 – Data da aprovação do trabalho:

Prof. Matheus Faria
Orientador

Titulação e Nome do Professor
Convidado 01
Convidado 1

Titulação e Nome do Professor
Convidado 02
Convidado 2

Brasília, DF
2018

Agradecimentos

A Deus por me dar conhecimento e oportunidades para realizar meus sonhos, e por me dar força para superar as dificuldades.

Aos meus pais por sempre me apoiarem, idenpendente do quão insano fosse meu objetivo, e por sempre estarem ao meu lado nos momentos de dificuldades.

Ao meu orientador Matheus Faria, que sempre entendeu minha dificuldades e sempre se mostrou disponível para qualquer dificuldade que eu enfrentei no decorrer do trabalho.

Ao meu coorientador Edson Alves por suas correções e incentivos.

E a todos que contribuíram diretamente ou indiretamente para a minha formação, o meu muito obrigado.

Resumo

O mercado de jogos nos dias atuais é um dos mercados mais lucrativos e também um dos que mais cresce nos decorrer dos anos. O objetivo desse trabalho é desenvolver uma plataforma web que possa auxiliar desenvolvedoras de jogos na hora da criação de um jogo. Para desenvolver essa primeiramente é preciso ter conhecimento sobre *business intelligence* e *game analytics*. *Business intelligence* é o ato de extrair dados de negócios e transformá-los em conhecimento, já *game analytics* é pegar esse conhecimento e transformar em métricas de jogos. O desenvolvimento foi feito pelo levantamento dos requisitos, depois foram feitos protótipo para o software de extração e para a plataforma web. O protótipo do software de extração é capaz de extrair dados de um log, manipulá-los e inseri-los no Elasticsearch.

Palavras-chaves: *game analytics*, *business intelligence*, análise de dados, métricas de jogos, plataforma web.

Abstract

The gaming market is one of most profitable and one with most growing over the years. The goal of this project is develop a web platform that can help game developers when creating a game. To develop this platform, first is need to have knowledgeable about business intelligence and game analytics. Business intelligence is the act of extract business data and turning it in knowledge, and game analytics is picking up that knowledge and turning it in game metrics. This development was done by requirements definition and development of prototypes for the extraction software and for the web platform. The prototype of extraction software is capable to extract datas from a log, manipulating it and insert into Elasticsearch..

Key-words: game analytics, business intelligence, data analytics, game metrics, web platform.

Lista de ilustrações

Figura 1 – Matriz Rastreabilidade Épico 1	30
Figura 2 – Matriz Rastreabilidade Épico 2	30
Figura 3 – Matriz Rastreabilidade Épico 3	30
Figura 4 – Arquitura Geral do Projeto	33
Figura 5 – Arquitura do Software de Extração	34
Figura 6 – Número total de jogos	38
Figura 7 – Número médio de donos	38
Figura 8 – Média das porcentagens de avaliações positivas	39
Figura 9 – Mês com mais lançamentos	39
Figura 10 – Lançamentos x Mês	39
Figura 11 – Média dos preços	39
Figura 12 – Número de jogos publicados pela própria desenvolvedora	40
Figura 13 – Número de jogos publicados por outra empresa	40
Figura 14 – Outra Publicadora x Própria Desenvolvedora	40
Figura 15 – Desenvolvedoras x Número de Jogos	41
Figura 16 – Linguagens Suportados x Número de Jogos	41
Figura 17 – Resultado da Compilação do <i>Handler</i>	44
Figura 18 – Tela Inicial	45
Figura 19 – Tela de Métricas	46
Figura 20 – Tela de <i>Rankings</i>	46
Figura 21 – Tela de Sugestões	47

Lista de tabelas

Tabela 1 – Vantagens e Desvantagens Steam DB	27
Tabela 2 – Vantagens e Desvantagens Steam Spy	27
Tabela 3 – Vantagens e Desvantagens DFC Intelligence	28
Tabela 4 – Número de donos e sua porcentagem	41
Tabela 5 – Porcentagem de avaliação	42
Tabela 6 – Cronograma do Trabalho	50

Lista de Códigos

3.1	Código do <i>Handler</i>	43
3.2	Código do <i>Plugin</i>	44

Lista de abreviaturas e siglas

BI	<i>Business Intelligence</i>
GUR	<i>Game User Research</i>
API	<i>Application Programming Interface</i>
XP	<i>Extreme Programming</i>
MVC	<i>Model-View-Controller</i>
REST	<i>Representational State Transfer</i>
TCC	Trabalho de Conclusão de Curso

Sumário

	Lista de Códigos	15
	Introdução	21
1	FUNDAMENTAÇÃO TEÓRICA	23
1.1	<i>Business Intelligence</i>	23
1.2	<i>Game Analytics</i>	24
1.2.1	Telemetria	25
1.2.2	Game Metrics	25
1.3	Softwares Correlatos	26
1.3.1	Steam DB	26
1.3.2	Steam Spy	27
1.3.3	DFC Intelligence	27
2	METODOLOGIA	29
2.1	Requisitos	29
2.2	Análise de Ferramentas	31
2.2.1	Banco de Dados	31
2.2.2	Extração de Dados	31
2.2.3	Frequência dos Dados	32
2.3	Arquitetura do Projeto	33
2.3.1	Logs	33
2.3.2	Software de Extração	34
2.3.3	Banco de Indexação	34
2.3.4	API de Consumo	35
2.3.5	Plataforma Web	35
2.4	Software Extração	35
2.5	API de Consumo	37
2.6	Plataforma Web	38
3	RESULTADOS OBTIDOS	43
3.1	Software de Extração	43
3.2	Plataforma Web	45
4	CONSIDERAÇÕES FINAIS	49
4.1	Trabalhos Futuros	49

REFERÊNCIAS	51
--------------------	-----------

Introdução

A indústria de jogos é uma das áreas de mercado mais lucrativa atualmente. No mercado brasileiro isto não seria diferente: sendo apenas o 13º maior mercado, no ano de 2017 movimentou 1,3 bilhões de dólares de acordo com o Newzoo ([NEWZOO, 2017a](#)), empresa que estuda o mercado de jogos, um valor pequeno se comparado ao mercado chinês, o maior mercado de jogos do mundo, o qual movimentou 27,5 bilhões de dólares no ano de 2017 ([NEWZOO, 2017b](#)).

Sendo um mercado bastante lucrativo, houve um grande aumento no número de empresas desenvolvedoras de jogos. No mercado brasileiro, houve um aumento de 600 % entre 2008 e 2016 no número de desenvolvedoras ([SILVEIRA, 2016](#)). Este aumento não se limita apenas ao mercado brasileiro: de acordo com o site Steamspy ([GALYONKIN, 2018](#)), plataforma web que exibe dados de jogos, apenas no ano de 2017 foram lançados 7.672 jogos na plataforma Steam, plataforma onde desenvolvedoras disponibilizam seus jogos, uma média de 21 jogos por dia. Porém com este grande crescimento fica cada vez mais difícil desenvolver um jogo que seja aceito pela comunidade de jogadores e que venda um número considerável de cópias.

Levando em conta a dificuldade de se criar um jogo nos tempos atuais, muitas desenvolvedoras iniciantes buscam por métricas que auxiliem na hora do desenvolvimento. Porém essas métricas não estão de fácil acesso, as exigindo assim, que as equipes montem essas métricas com os dados que possuem ou apenas as ignorem e desenvolvam o jogo sem elas. Esta dificuldade de achar métricas ocorre principalmente pela falta de uma ferramenta especializada nesta área.

Por isso o objetivo deste trabalho é a elaboração de métricas de negócios, como o número médio de vendas de determinado gênero, e a criação de uma plataforma web onde essas métricas serão disponibilizadas em conjunto com sugestões para as desenvolvedoras.

Objetivos

O objetivo deste trabalho é desenvolver uma plataforma que disponibilize métricas e sugestões para desenvolvedoras de jogos.

Os objetivos específicos são:

- elaborar métricas a partir das informações dos jogos;
- automatizar as sugestões de melhorias para a concepção do jogo.;

- elaborar uma arquitetura responsável por fazer a extração e manipulação dos dados dos jogos;
- desenvolver um software responsável por extrair e manipular os dados que serão utilizados.

Estrutura do Documento

Este documento será dividido em 4 capítulos. O Capítulo 1 é responsável pelo referencial teórico do projeto, com um estudo sobre *game analytics* e *business intelligence*, e análise de concorrentes e ferramentas que serão utilizadas no projeto. O Capítulo 2 é responsável pela metodologia, mostrando como será feito o projeto. O Capítulo 3 é responsável pelos resultados obtidos no projeto. O Capítulo 4 é responsável pela conclusão do projeto, também será mostrados possíveis trabalhos futuros.

1 Fundamentação Teórica

Neste capítulo será abordada a fundamentação teórica para o entendimento do propósito da implementação do projeto. Nele são explicados os conceitos de *game analytics* e *business intelligence* e também serão abordados softwares parecidos com o que será desenvolvido.

1.1 *Business Intelligence*

No decorrer dos anos houve uma grande mudança em relação a criação, coleta e do uso de dados. Enquanto houve uma grande evolução na maneira que esses dados eram gerenciados, sempre houve um desejo de extrair valores de negócios nas grandes pilhas de dados, que são capturadas hoje em diversas fontes, sejam elas estruturas de dados ou arquivos.

Os resultados da análise destas pilhas de dados e da criação de conhecimentos sobre esta análise, criou-se uma vantagem no mercado de negócios inimagináveis. Com esse conhecimento foi possível criar políticas de escolhas de ações para negócios em diferentes cenários. A exposição e exploração destes conhecimento é apenas uma das vantagens do uso de BI.

Uma das maiores vantagens do uso do BI está na otimização do processo de tomadas de decisão, pois para cada processo de negócios é associado a sua performance, e num mundo perfeito cada escolha deve ser a mais otimizada, ou seja, a que têm a melhor performance ([LOSHIN, 2012](#)).

A utilização de BI no âmbito de trabalho apresenta uma evolução na performance nas seguintes dimensões de negócio:

- No valor financeiro associado ao crescimento da lucratividade, sejam elas derivadas de custos ou do aumento de receitas.
- No valor de produtividade associado a diminuição da carga de trabalho, diminuição do tempo necessário para a execução de processos ponta-a-ponta e no aumento da porcentagem de produtos de alta qualidade.
- No valor de confiança, como maior satisfação do cliente, funcionário ou fornecedor, assim como aumento na confiança de previsões, consistência operacional e relatórios gerenciais, reduções no tempo gasto com “paralisia de análise” e melhor resultados de decisões.

- No valor de risco associado com a uma melhor visibilidade da exposição do crédito, confiança no investimentos em capitais e conformidade auditável com a jurisdição e normas e regulamentos da indústria.

The Data Warehousing Institute ([ECKERSON, 2002](#)), uma instituição especializada na educação e treinamento em armazenamento de dados, define que BI é os processos, as tecnologias e as ferramentas necessárias para transformar dados em informação, informação em conhecimento e conhecimento em planos que dirigem rentáveis planos de negócios, ou seja BI engloba armazenamento de dados, ferramentas de análíticas e gestão de informação/conhecimento.

Business intelligence geralmente considerar as informações dos dados como ativos, por isso, pode-se valer a pena examinar o uso de informações no contexto de como o valor é criado dentro de uma organização. Para isso existem três tipos diferentes de perspectivas, sendo elas:

- **Perspectiva funcional:** Neste tipo de perspectiva, os processos focam nas tarefas relacionadas a algum tipo particular de negócio, como vendas, *marketing*, entre outros. Processos funcionais confiam nos dados que operam dentro dos padrões de atividades comerciais.
- **Perspectiva interfuncional:** Como a maioria das empresas funcionam como um aglomerado de processos funcionais, e isto reflete em informações mais complexas. Para esta perspectiva a atividade foi um sucesso quando todas as tarefas foram completadas. Pela sua própria natureza, os processos envolvidos compartilham informações em diferentes funções, e o sucesso é medido tanto em termos de conclusão bem-sucedida, bem como as características do desempenho geral
- **Perspectiva empresarial:** Num ponto de vista organizacional e observando as características de desempenho dos processos interfuncionais, pode-se informar arquitetos empresariais e analistas de negócios maneiras na qual a organização pode mudar e melhorar o jeito que as coisas são feitas. Neste ponto de vista, o dado não é mais usado apenas para executar negócios, dados são utilizados para melhorar os negócios

1.2 *Game Analytics*

O desenvolvimento de jogos hoje pode se mostrar como um grande desafio, e grande parte deste desafio se dá pelo fato do grande número de jogos publicados. Para auxiliar as desenvolvedoras a criarem jogos eficientemente foram criados várias ferramentas e técnicas, um destes métodos é o *analytics*.

Analytics é o processo de descobrir e comunicar padrões em dados, solucionando problemas de negócios ou suportar decisões de gerenciamento de empresas. Esta metodologia possui seus fundamentos em mineração de dados, na matemática, estatística, programação e operações de busca, como também na visualização dos dados, de forma a comunicar padrões relevantes. Vale mencionar que o *analytics* não é apenas perguntar e relatar dados de BI, e sim análises atual daqueles dados (DAVEPORT; HARRIS, 2007).

Game analytics é uma aplicação do *analytics* para o contexto de desenvolvimento de jogos (ANDERS; EL-NASR; CANOSSA, 2013). Um dos maiores benefícios em utilizar o *game analytics* é o suporte na hora de fazer decisões em todos os níveis e áreas organizacionais. Este método é direcionado tanto como a análise de um jogo com um produto, tanto como a análise de um jogo como projeto.

A aplicação padrão do *game analytics* é na hora de informar o GUR. GUR é a aplicação de várias técnicas e metodologias para avaliar a maneira na qual os jogadores jogam, e o nível de interação entre o jogador e o jogo. Vale mencionar que *game analytics* não é só GUR, já que o GUR é focado nos dados obtidos a partir dos usuários, já o *game analytics* considera todos os tipo de dados obtidos no desenvolvimento do jogo.

1.2.1 Telemetria

Telemetria são os dados obtidos à distância, geralmente digitais, porém qualquer dado transmitido à distância e telemetria. No contexto de jogos, telemetria seria algum jogo transmitindo dados sobre a interação do jogador com o jogo.

Telemetria de jogos é o termo utilizado para qualquer dado obtidos a distância que pertence durante o desenvolvimento ou evolução de um jogo, e isto inclui o monitoramento e análise de: servidores, dispositivos celulares e comportamento dos usuários. A fonte que produz mais dados por telemetrias, são os de usuário, por exemplo, interação com jogos, comportamento de compra e interações com outros jogadores ou aplicativos (BOHANNON, 2010).

1.2.2 Game Metrics

Em sua forma pura, os dados obtidos a partir da telemetria, não são de muito auxílio, por isso estes dados devem ser transformado em várias métricas interpretativas, como: o número de jogadores ativos por dia, bugs arrumados por semana, entre outros. Essas métricas são chamadas de game metrics. *Game metrics* possuem o mesmo potencial que outras fontes de BI. *Game metrics* geralmente são definidas como um medição quantitativa de um ou mais atributos, de um ou mais objetos que operem no contexto de um jogo.

Métricas podem ser variáveis ou agregações mais complexas, como a soma de várias

variáveis, em outras palavras as métricas podem ser simples variáveis que geram uma análise básica, ou a combinação de várias variáveis para gerar uma análise mais complexa e completa. Métricas que não estão relacionadas diretamente ao jogo, são chamadas de métricas de negócios. Durante a utilização do *game analytics* é essencial a distinção entre as métricas de negócio e as *textitgame metrics*.

As *game metrics* foram categorizadas em três tipos por Mellon ([MELLON, 2009](#)), ou seja, as *game metrics* podem ser definidas como:

- **Métricas de usuário:** São métricas relacionadas aos usuários que jogam aquele jogo, pela perspectiva de jogadores, ou de clientes. A perspectiva de cliente é utilizada quando as métricas são relacionadas a receita. A perspectiva de jogador é utilizada para investigar como é a interação das pessoas com o sistema do jogo e seus componentes.
- **Métricas de performance:** São métricas relacionadas a performance da tecnologia e arquitetura utilizada no jogo, muito relevantes para jogos online. Essas métricas geralmente são utilizadas no monitoramento dos impactos causado por alguma atualização no jogo.
- **Métricas de processo:** São métricas relacionadas ao processo de desenvolvimento de jogos. Similares as métricas de performance, são utilizadas para gerenciar e monitorar métodos que foram adotados, ou que foram adotados na hora do desenvolvimento do jogo.

1.3 Softwares Correlatos

Nesta parte de documento é levantado os softwares correlacionados, que possuem características ou objetivos parecidos com o software a ser desenvolvido. Os principais são o Steam DB, uma ferramenta que disponibiliza informações sobre o banco de dados da Steam; a Steam Spy, uma plataforma web que disponibiliza informações sobre os jogos e suas vendas por região e o DFC Intelligence, uma ferramenta de pesquisa sobre o mercado de jogos.

1.3.1 Steam DB

Steam DB é uma ferramenta *open source third-party* com objetivo de dar um melhor conhecimento sobre as aplicações e pacotes disponíveis no banco de dados da Steam ([STEAMDB, 2010](#)). Esta ferramenta disponibiliza *rankings* de jogos e gráficos de jogos para que o usuário tenha uma melhor visualização destes. Uma diferença crucial entre o Steam DB e o software proposto está no aspecto que o projeto não se especifica num jogo apenas. Na tabela 1 podemos ver as vantagens e desvantagens do Steam DB.

Vantagens	Desvantagens
<i>Open source</i>	Não possui política de contribuição
<i>Rankings</i> e gráficos	Demonstra informações de um jogo apenas
Gratuito	Precisa de login na Steam
Disponibiliza informações gerais de um usuário	

Tabela 1 – Vantagens e Desvantagens Steam DB

Vantagens	Desvantagens
Disponibiliza número de donos e vendas por região	Não é totalmente gratuito
Gráficos genéricos e confusos	Não apresenta métricas sobre os jogos
Disponibiliza API para seus dados	Não é <i>open source</i>

Tabela 2 – Vantagens e Desvantagens Steam Spy

1.3.2 Steam Spy

Steam Spy é uma plataforma Web que a partir de informações sobre os usuários da Steam, disponibiliza informações como o número de donos de algum jogo ou vendas por região de determinado jogo, de maneira eficiente e bonita (GALYONKIN, 2018). Um dos objetivos especificados pelo Steam Spy é o auxílio a desenvolvedores *indies*, porém para se conseguir todos os dados disponíveis pela plataforma é preciso pagar por eles. Uma diferença crucial entre o Steam Spy e o software é que o software a ser implementado está focado apenas no auxílio de desenvolvedores e será uma ferramenta *open source* e gratuita. Na tabela 2 podemos ver as vantagens e desvantagens do Steam Spy.

1.3.3 DFC Intelligence

DFC Intelligence é uma ferramenta paga que auxiliam desenvolvedoras a tomar conhecimentos sobre estatísticas de seus jogos no mercado. Para ser mais específico, uma desenvolvedora que utilize essa ferramenta receberá informações sobre o número de vendas de seus jogos, picos de vendas, regiões que mais venderam, entre outras. Também será disponibilizado gráficos e métricas que informam como está o mercado de jogos (COLE, 1994). Uma diferença deste trabalho para o DFC Intelligence, além do fato do projeto ser gratuito, e que o projeto não irá focar apenas numa desenvolvedora e sim no mercado inteiro de jogos. Na tabela 3 podemos ver as vantagens e desvantagens do Steam DB.

Vantagens	Desvantagens
Disponibiliza informações sobre o mercado de jogos	Não é gratuita
Auxiliam desenvolvedores na hora da criação de um jogo	Não mostrar o mercado de jogos como um todo
	Não é <i>open source</i>
	Foca apenas nos jogos de uma desenvolvedora

Tabela 3 – Vantagens e Desvantagens DFC Intelligence

2 Metodologia

A metodologia escolhida para o desenvolvimento do projeto será a metodologia ágil, mais especificamente será utilizado o Kanban. A escolha dessa metodologia se dá pelas seguintes características: o projeto será desenvolvido de maneira incremental, ou seja, ele poderá ser modificado no decorrer da implementação; a equipe consiste em apenas uma pessoa, o que descarta a possibilidade de utilizar o Scrum ou XP; o escopo do projeto será divididos em tarefas, e a utilização do Kanban facilita no descobrimento de gargalos.

A metodologia Kanban surgiu no Japão com o TPS (OHNO, 1997) para controlar a fabricação de automóveis e foi inserida no meio de desenvolvimento de software no ano de 2007. Kanban é um termo japonês para sinal visual e uma das grandes características dessa metodologia é evidenciar os problemas existentes no processo.

A metodologia ágil surgiu no ano de 2001, com a reunião de especialistas em processos de desenvolvimento de software para discutir maneiras de melhorar o desempenho em projetos, com isso foi criado o Manifesto Ágil (BECK et al., 2001). Uma das características das metodologias ágeis são sua capacidade de adaptar a novos fatores durante o desenvolvimento do projeto, ao invés de tentar prever o que pode acontecer e o que não pode

2.1 Requisitos

Os requisitos do projeto serão classificados em três níveis de hierarquia: épicos, *features* e *user stories*. Os épicos são compostos por *features* e representam macro entregáveis do projeto e geralmente são descritos como o nível mais abstrato. As *features* são agrupamentos de *user stories* e representam funcionalidades do sistema. User stories são pequenas partes de uma funcionalidade do sistema e que serão implementadas. *User stories* também podem ser classificadas como *technical stories* e estas *technical stories* representam atividades que não agregam muito valor ao cliente, porém agregam melhoras internas no software.

No contexto do projeto as *features* serão equivalentes às *milestones*, e as *stories* (*users* ou *technical*) serão as *issues*. Para uma melhor visualização dos requisitos do projeto, será montado uma matriz de rastreabilidade para cada epico, sendo mostrado nas figuras 1, 2 e 3.

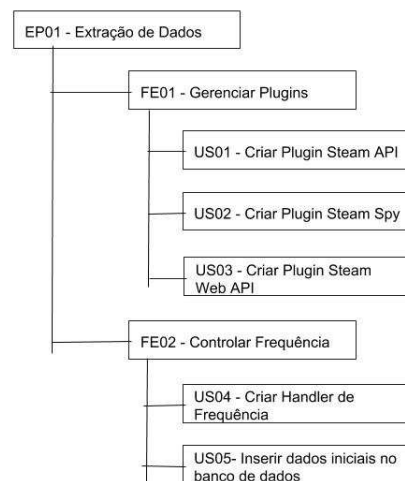


Figura 1 – Matriz Rastreabilidade Épico 1

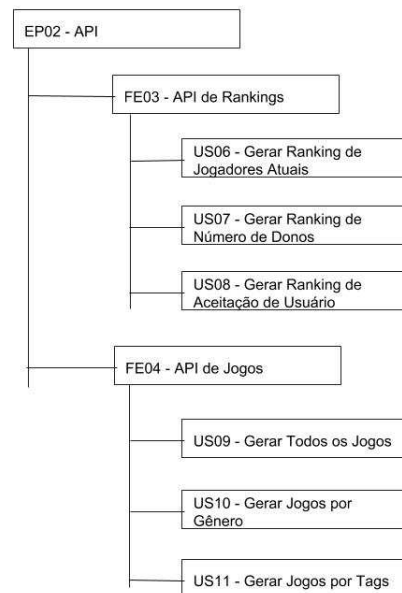


Figura 2 – Matriz Rastreabilidade Épico 2

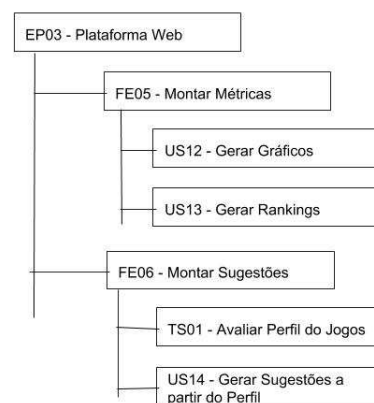


Figura 3 – Matriz Rastreabilidade Épico 3

2.2 Análise de Ferramentas

Nesta seção são feito estudos sobre as possíveis ferramentas que poderão ser utilizadas durante o desenvolvimento do trabalho.

2.2.1 Banco de Dados

A ferramenta de banco de dados é responsável pelo armazenamento dos dados extraídos dos *logs*. No contexto desse projeto, este banco deverá armazenar os dados de todos os jogos disponibilizados pela API da Steam.

PostgreSQL

Postgre é uma ferramenta *open source* de banco de dados que utiliza a linguagem SQL em conjunto com outras funcionalidades que guardam e manipulam os mais complicados dos dados. Postgre é uma ferramenta antiga tendo sua origem datada no ano de 1986 (POSTGRESQL, 1997).

Postgre é uma ferramenta que possui funcionalidades que auxiliam desenvolvedores e administradores a manter a integridade dos dados e criar sistemas com tolerância a falhas.

Elasticsearch

Elasticsearch é uma ferramenta *open source*, desenvolvida pela Elastic ¹, de análise e busca REST² capaz de resolver um grande número de casos. É a parte principal de Elastic Stack, servindo como um centro de armazenamento de dados (ELASTIC, 2010a).

Elasticsearch suporta qualquer tipo de dado, além de agregar grande quantidades de dados para se ter uma visão melhor. Entre suas características as que mais se destacam são sua rapidez de busca, capacidade de detecção de falhas, múltiplos tipos de dados e suporte a múltiplas linguagens de programação.

2.2.2 Extração de Dados

A ferramenta de extração de dados é responsável pela extração dos dados dos *logs*, além de ter que fazer a comunicação com o banco de dados. Essa ferramenta deverá extrair os dados das APIs que serão utilizadas, manipulá-los e inserir no banco de dados.

¹ <<https://www.elastic.co/>>

² REST é uma abstração da arquitetura da Web

Logstash

Logstash é a ferramenta de extração de dados, desenvolvida pela Elastic, ele faz parte de Elastic Stack em conjunto com o Elasticsearch. Entre as características as que mais destacam são sua capacidade de conseguir extrair qualquer tipo de dado, além de permitir a criação de filtros para transformar e manipular os dados extraídos e também permite uma gama de outputs para onde o dado será enviado (ELASTIC, 2010b).

Owner

Outra opção de extração de dados é a criação de um software específico para a plataforma. Este software seria criado na linguagem Ruby³ que também possui ligação com o Elasticsearch. As vantagens desse tipo de software seria a possível implementação de uma arquitetura de *plugins*, de modo que outros tipos de dados seriam aceitos na plataforma, além de que, caso futuramente seja necessário a mudança do banco de dados, a adaptação para o novo banco seria menos custosa.

2.2.3 Frequência dos Dados

A ferramenta da frequência de dados é mais um auxiliar na hora da extração de dados. Como serão extraídos dados que possuem uma taxa de modificação grande, é necessário utilizar uma ferramenta que crie rotinas para essa extração.

Cronjob

*Cronjob*⁴ é uma ferramenta de agendamento que permite controlar tarefas a serem executadas em tempos pré-configurados. Através dela é possível configurar tarefas automáticas a serem executadas em um horário específico. *Cronjobs* são configurados manualmente pelo terminal, podendo ser configurados para atualizações por minutos, horas, dias (do mês ou da semana) e meses

Whenever

Whenever é uma *gem*⁵ do *ruby*, mais especificamente para o *framework rails*. Está *gem* utiliza os *cronjobs* como fundo para gerenciar e controlar tarefas a serem executadas em rotinas ou em um horário específico (MAKHMALI, 2018).

³ <<https://www.ruby-lang.org/pt/>>

⁴ <<https://cron-job.org/en/>>

⁵ Gems são como bibliotecas para a linguagem Ruby

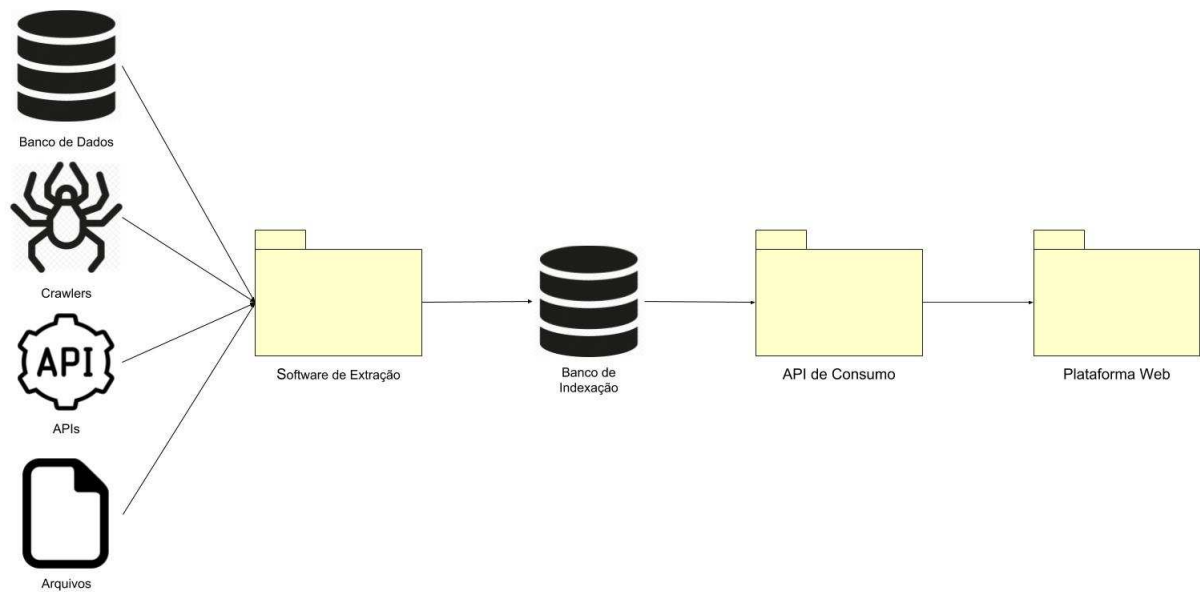


Figura 4 – Arquitetura Geral do Projeto

Clockwork

Clockwork é uma *gem* do *ruby*. Esta *gem* também utiliza os *cronjobs*, porém diferente de *whenever*, ele não exige que o software possua uma arquitetura MVC⁶. Ele é utilizado especificamente para a criação de rotinas para tarefas que serão executadas automaticamente pela *gem* (WIGGINS, 2018).

2.3 Arquitetura do Projeto

A arquitetura do projeto é dividida em cinco partes: os *logs*, o software de extração, o banco de indexação, a API de consumo e a plataforma web. A ligação entre as partes e sua posição na arquitetura ficam evidenciado na figura 4.

2.3.1 Logs

Logs é a parte onde se concentra os dados que serão utilizados para o desenvolvimento da plataforma web, como mostrado na figura 4, *logs* representam de onde virão os dados, sejam elas de um banco de dados, de funções *crawlers*, de alguma API ou de um arquivo local.

Esses *logs* serão guardados e manipulados pelo software de extração. Para o escopo inicial do projeto serão utilizados três *logs*, sendo eles provenientes de três APIs: a API do Steam Spy, a API do Steam Store e a API do Steam Web.

⁶ É um padrão de arquitetura de software

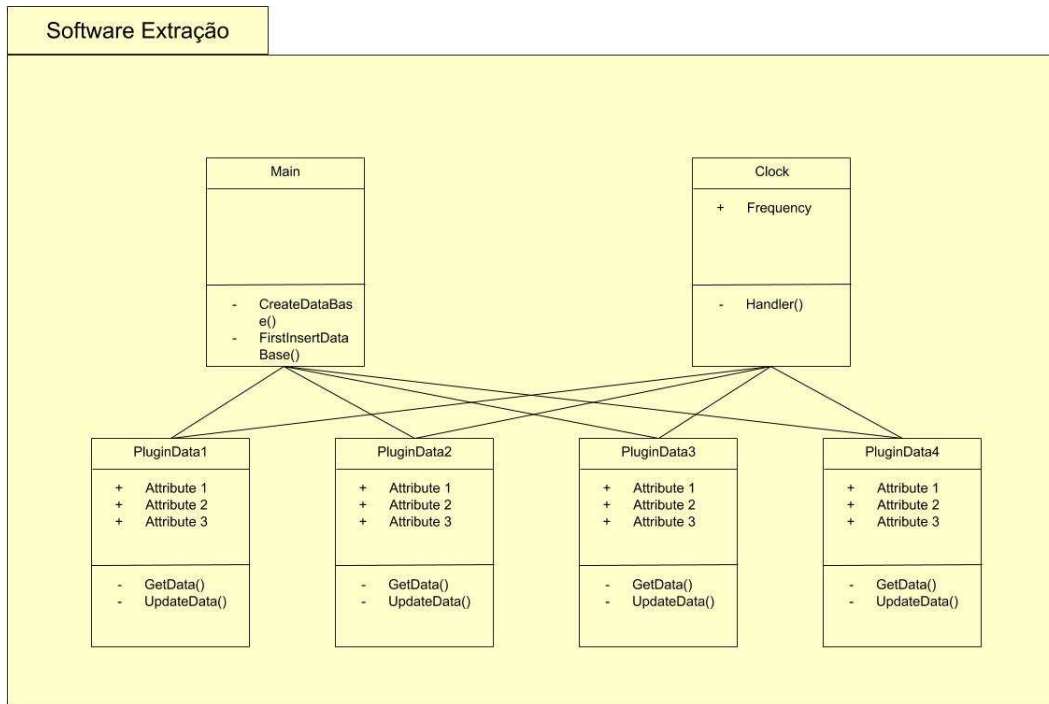


Figura 5 – Arquitura do Software de Extração

2.3.2 Software de Extração

O software de extração é dividido em três componentes: o *handler*, a classe *main* e os *plugins*. A arquitetura do software de extração e a interação entre seus componentes são demonstrados na figura 5. A classe *Main* é responsável por criar o banco de indexação e fazer a primeira população no mesmo. Como demonstrado na figura 4 a classe *main* irá receber os dados para a primeira inserção dos *plugins*.

O *handler* é responsável por atualizar as informações no banco de indexação. Essas informações virão do *plugin* que, além de mandar a informação, deve indicar com que frequência esses dados devem ser atualizados.

Já os *plugins* são responsáveis por extrair os dados dos *logs*, manipulá-los caso seja necessário e enviar para o banco de dados. Para que o software de extração possua uma melhor organização, cada *plugin* deve ser responsável por um tipo de *log*, pois isto permite que outros *plugins* possam ser adicionados no software de extração.

2.3.3 Banco de Indexação

O banco de indexação será responsável por guardar os dados extraídos do *log* pelo software de extração. Como a Steam possui muito jogos, o banco de indexação que será usado deverá suportar um grande número de dados e também possuir uma rapidez na hora da busca. O banco de dados escolhido foi o Elasticsearch, por que além de suportar grandes quantidades de dados e ser rápido, ele também disponibiliza certa estatísticas

sobre os dados e ser suportado por muitas linguagens de programação.

2.3.4 API de Consumo

A API de consumo permite que mais de uma plataforma possam utilizar seus dados. Esta API irá consumir os dados do banco de indexação e, a partir destes dados gerar *links* de API. Os principais *links* que serão desenvolvidos no escopo do projeto serão os de *rankings* e os de jogos.

2.3.5 Plataforma Web

A plataforma web é o principal componente da arquitetura, pois é nele que será mostrado as métricas levantadas para o auxílios dos desenvolvedores. Esta plataforma irá consumir os dados disponibilizados pela API, montar métricas a partir desses dados e disponibilizá-los de maneira que seja fácil a interpretação dos usuários. A plataforma também será responsável por montar sugestões para as desenvolvedoras.

2.4 Software Extração

O software de extração é a primeira parte propriamente dita que faz parte do projeto, nela é onde será feito a extração dos dados dos *logs*, a manipulação necessária e a inserção dos dados no banco de indexação numa determinada frequência. Primeiramente iremos falar sobre os *logs*. *Logs*, como já citados na arquitetura do projeto, são os arquivos que contém os dados que serão inseridos no banco de indexação. No escopo inicial serão utilizados três *logs*:

- **Steam Web API:** É uma API REST disponibilizada pela Steamworks⁷, ou seja, é uma API oficial da Steam (STEAMWORKS, 2018). Ela é uma API que possui tanto métodos públicos, tanto métodos privados. Os métodos públicos são abertos para qualquer pessoas visualizar, já os métodos privados é necessário uma chave de desenvolvedor cedido pela própria Steamworks. Para acessar seus dados é preciso, além da chave, a interface na qual aquele dados está guardado, o id do jogo, caso a informação seja de um jogo, ou do id do usuário, caso a informação seja de algum usuário.
- **Steam Store API:** É uma API REST disponibilizada pela Steam, que não possui uma página oficial para ela. Esta API disponibiliza os dados dos jogos que estão guardados no banco de dados da Steam. Para acessar seus dados é preciso saber apenas o id do jogo, porém é possível passar filtros também.

⁷ <<https://partner.steamgames.com/>>

- **Steam Spy API:** É uma API REST disponibilizada pelo Steam Spy ([GALYONKIN, 2018](#)), ela é muito parecido com a Steam Store API, porém ela disponibiliza dados que apenas o Steam Spy possui, como o número de donos de um jogo ou o número de avaliações positivas e negativas de um jogo. Para acessar seus dados é preciso saber apenas o id do jogo, porém também é possível passar filtros.

Com os *logs* definidos, é preciso saber quais dados dos *logs* serão inseridos no banco de indexação, pois muitas APIs possuem dados repetidos e nem todos os dados são interessantes para se manter. Com isso os principais dados que serão mantidos no banco de indexação são:

- **Nome:** É o nome do jogo em questão, este dado é guardado apenas para uma referência, pois ele não será utilizado na montagem das métricas.
- **Steam Id:** É o id do jogo, esse número é necessário pois é a partir dele que serão extraídas as informações das APIs.
- **Descrição:** É uma breve descrição sobre o jogo, este dado será guardado apenas para uma referência, pois ele não será utilizado na montagem das métricas.
- **Desenvolvedora:** É o nome da empresa que desenvolveu o jogo, este dado será guardado, pois será levantado o número de desenvolvedoras que são a própria publicadora.
- **Publicadora:** É o nome da empresa que publicou aquele jogo, este dado será guardado, pois será levantado o número de desenvolvedoras que são a própria publicadora.
- **Preço:** É o preço em reais daquele jogo, este dado será guardado, pois será levantando o preço médio que um jogo em determinado perfil geralmente têm.
- **Categorias:** São as categorias que aquele jogo abrange, geralmente são mostradas como características que aquele jogo têm, este dado será guardado, pois a partir de determinadas categorias que um determinado perfil de jogos, poderá levantar sugestões para a melhoria daquele perfil.
- **Gêneros:** São os gêneros daquele jogo, este dado será guardado, pois a partir de um determinado gênero, novas métricas serão montadas.
- **Data de Lançamento:** É a data de lançamento daquele jogo, este dado será guardado, pois com ele será montado uma métrica de quais meses um jogo com um determinado perfil foi mais lançado.

- **Linguagens Suportadas:** São as línguas que aquele jogo suporta, este dado será guardado, pois será levantado quais linguagens um jogo com determinado perfil possui mais suporte.
- **Número de Donos:** É o número médio de donos que aquele jogo têm, este dado será guardado, pois será levantada uma média de número de donos que um determinado perfil atingiu.
- **Avaliações Positivas e Negativas:** São o número de avaliações positivas e negativas que um jogo têm, este dado será guardado pois com ele será possível determinar a porcentagem de avaliações positivas que um jogo teve, e com isso será levantado uma média desta porcentagem que um determinado perfil atingiu.
- **Número de Jogadores Atuais:** É o número de jogadores que estão jogando um determinado jogo naquele momento, este dado será guardado, pois com ele será possível gerar *rankings* de quais jogo de um determinado perfil estão sendo mais jogados.

Com os dados definidos, é preciso definir como será feito o software de extração. O software de extração será feito utilizando a linguagem de programação Ruby, com integração com o Elasticsearch. Para a atualização frequente dos dados será utilizado a *gem Clockwork*. A escolha do Ruby como linguagem de programação, se justifica pelo fato, que o integrante do projeto possui mais familiaridade com a linguagem.

2.5 API de Consumo

A API de consumo será responsável por consumir o banco de indexação, e partir dos dados gerar *lend points* REST para que qualquer aplicação possa utilizá-la. A API deverá atualizar seus *end points* com a mesma frequência com o qual o software de extração atualiza o banco de indexação. Esta API será desenvolvida utilizando o *framework* Ruby on Rails, isto se justifica pelo fato, que o integrante do projeto possui mais familiaridade com o *framework*.

A API possuirá dois *end points*: uma para gerar informações sobre os jogos no banco de indexação e outro para gerar *rankings* para um determinado perfil.

O *end point* de informações dos jogos poderá receber vários filtros, sendo os mais comuns: pegar todos os jogos disponíveis no banco de indexação, pegar todos os jogos de um ou mais gêneros, pegar todos os jogos com uma ou mais determinadas categorias. Os *end points* mais comuns são:

- <https://api/games/all>

NÚMERO TOTAL DE JOGOS: 10

Figura 6 – Número total de jogos

MÉDIA DE DONOS: 4628.500

Figura 7 – Número médio de donos

- <https://api/games/genre=action>
- <https://api/games/categories=atmospheric>

O *end points* de *rankings* gerarão *rankings* a partir de determinados filtros, sendo os mais comuns: gerar um *rankings* pelos jogos mais jogados naquele momento, gerar um *rankings* com os jogos que possui as maiores porcentagens de avaliações positivas. A API também deverá gerar qualquer ranking com os filtros de gêneros e categorias, ou seja, deve-se gerar um ranking qualquer para um determinado perfil de jogo. Os *end points* mais comuns são:

- <https://api/rankings/all>
- <https://api/rankings/current>
- <https://api/rankings/avaliation>

2.6 Plataforma Web

A Plataforma Web é a parte principal do projeto, nela é onde será mostrado as métricas que serão montadas a partir dos dados extraídos dos *logs*. A plataforma será desenvolvida em algum *framework* de JavaScript, retirando os dados da API de consumo.

Outra funcionalidade importante para a plataforma web é a habilidade de fazer sugestões a partir de um determinado perfil de jogo. As sugestões serão a priori simples, sendo as mais comuns, a adição ou remoção de um gênero ou categoria no perfil, para que este tenha uma aumento no seus número de vendas ou no número da porcentagem de avaliações positivas.

As métricas que serão disponibilizadas inicialmente pela plataforma são:

- Número total de jogos que atendam aquele determinado perfil. A figura 6 é uma representação da métrica.
- Número médio de donos que um jogo que atende aquele determinado perfil. A figura 7 é uma representação da métrica.



Figura 8 – Média das porcentagens de avaliações positivas

MÊS COM MAIS LANÇAMENTOS: ABRIL

Figura 9 – Mês com mais lançamentos

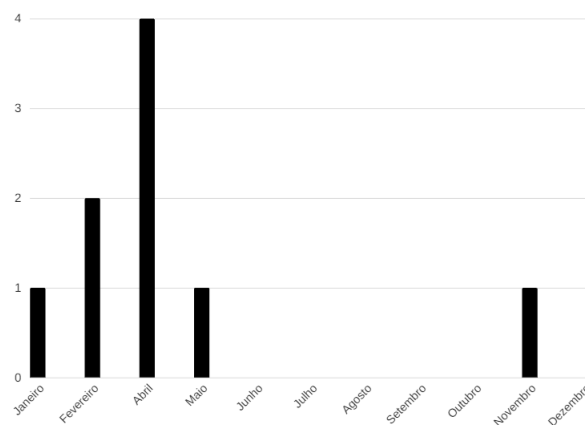


Figura 10 – Lançamentos x Mês

- Média das porcentagens de avaliações positivas de um jogo que atende aquele determinado perfil. A figura 8 é uma representação da métrica.
- Mês que possui o maior número de lançamento que atendam aquele determinado perfil. A figura 9 é uma representação da métrica.
- Gráfico de lançamentos por mês que atendam aquele determinado perfil. A figura 10 é uma representação da métrica.
- Preço médio de um jogo que atende aquele determinado perfil. A figura 11 é uma representação da métrica.
- Número de jogos que foram publicados pela própria desenvolvedora que atendam aquele determinado perfil. A figura 12 é uma representação da métrica.

PREÇO MÉDIO: R\$ 85,99

Figura 11 – Média dos preços

NÚMERO DE JOGOS OWN PUBLISHER: 7

Figura 12 – Número de jogos publicados pela propria desenvolvedora

NÚMERO DE JOGOS ANOTHER PUBLISHER: 3

Figura 13 – Número de jogos publicados por outra empresa

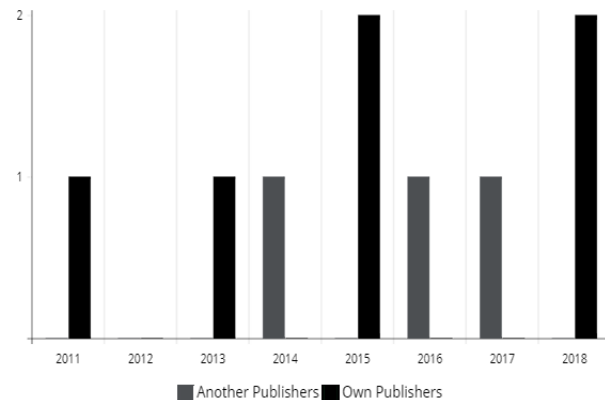


Figura 14 – Outra Publicadora x Própria Desenvolvedora

- Número de jogos que foram publicado por outra empresa que atendam aquele determinado perfil. A figura 13 é uma representação da métrica.
- Gráfico que compara os números de jogos que foram publicados pela desenvolvedora pelo números de jogos que foram publicados por outra empresa por ano. A figura 14 é uma representação da métrica.
- Tabela de quantos jogos daquele determinado perfil atingiram um número N de donos e sua porcentagem. A representação desta tabela fica evidenciada pela tabela 4.
- Tabela de quantos jogos daquele determinado perfil atingiram um número N de avaliações positivas e sua porcentagem. A representação desta tabela fica evidenciada pela tabela 5
- Gráfico de desenvolvedoras por número de jogos desenvolvidos. A figura 15 é uma representação da métrica.
- Gráfico de linguagens suportados por número de jogos que a suportam. A figura 16 é uma representação da métrica.

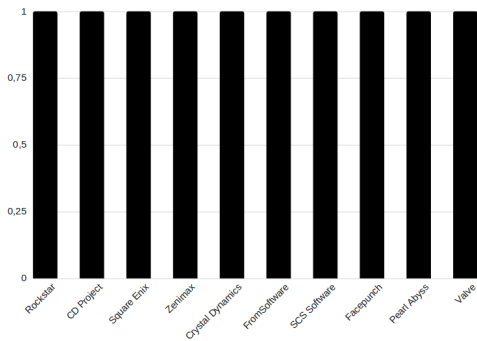


Figura 15 – Desenvolvedoras x Número de Jogos

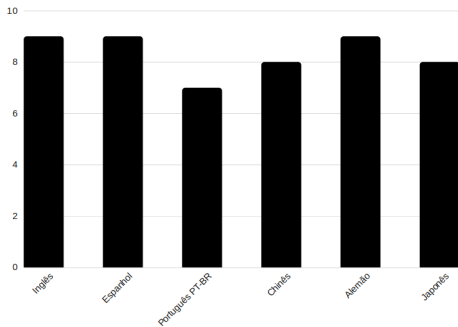


Figura 16 – Linguagens Suportados x Número de Jogos

Número de Donos	Quantidade de Jogos	% de Jogos
0 - 10000	10	100 %
10000 - 50000	10	100 %
50000 - 100000	9	90 %
100000 - 500000	9	90 %
500000 - 1000000	9	90 %
1000000 - 5000000	8	80 %
5000000 - 10000000	3	30 %
10000000 - 50000000	1	10 %

Tabela 4 – Número de donos e sua porcentagem

Porcentagem de Avaliação	Quantidade de Jogos	% de Jogos
0 %- 10 %	10	100 %
11 %- 20 %	10	100 %
21 %- 30 %	10	100 %
31 %- 40 %	10	100 %
41 %- 50 %	10	100 %
51 %- 60 %	10	100 %
61 %- 70 %	10	100 %
71 %- 80 %	8	80 %
81 %- 90 %	6	60 %
91 %- 100 %	4	40 %

Tabela 5 – Porcentagem de avaliação

3 Resultados Obtidos

Neste capítulo serão exibidos os resultados obtidos nesta primeira parte do projeto. Será falado dos resultados sobre o software de extração e da plataforma web.

3.1 Software de Extração

Na parte de extração dos dados foram criados dois codigos teste. O código 3.1 representa o uso da *gem clockwork* como um *handler* de frequência, o resultado obtido pelo uso do código pode ser visto na figura 17. Já o código 3.2 representa uma estrutura inicial de um *plugin*, nele é feito a conexão com o Elasticsearch, extraídos os dados de uma API, manipula esses dados e inserir no Elasticsearch.

Código 3.1 – Código do *Handler*

```
require 'clockwork'
require 'active_support/time'

# O modulo necessita ter o nome Clockwork
module Clockwork

  count_frequency = 0
  count_games = 0

  # Handler dos jobs
  handler do |job|
    if job.eql? 'frequency'
      count_frequency += 1
      print "A frequencia e: #{count_frequency}\n"
    elsif job.eql? 'games'
      count_games += 1
      print "O numero de jogos e: #{count_games}\n"
    end
  end

  # Define a frequencia dos jobs
  every(30.seconds, 'frequency')
  every(1.minutes, 'games')
end
```

```

~/Documentos/Clockwork
▶ clockwork handler.rb
I, [2018-07-03T11:49:46.965562 #4932] INFO -- : Starting clock for 2 events: [ frequency games ]
I, [2018-07-03T11:49:46.965822 #4932] INFO -- : Triggering 'frequency'
A frequência é: 1
I, [2018-07-03T11:49:46.965876 #4932] INFO -- : Triggering 'games'
O número de jogos é: 1
I, [2018-07-03T11:50:16.001333 #4932] INFO -- : Triggering 'frequency'
A frequência é: 2
I, [2018-07-03T11:50:46.001290 #4932] INFO -- : Triggering 'frequency'
A frequência é: 3
I, [2018-07-03T11:50:46.001424 #4932] INFO -- : Triggering 'games'
O número de jogos é: 2
I, [2018-07-03T11:51:16.001320 #4932] INFO -- : Triggering 'frequency'
A frequência é: 4
I, [2018-07-03T11:51:46.001298 #4932] INFO -- : Triggering 'frequency'
A frequência é: 5
I, [2018-07-03T11:51:46.001432 #4932] INFO -- : Triggering 'games'
O número de jogos é: 3

```

Figura 17 – Resultado da Compilação do *Handler*Código 3.2 – Código do *Plugin*

```

require 'httparty'
require 'elasticsearch'

# Classe que funciona como uma hash
class Game < Hash
  def put(key, value)
    self[key] = value;
  end
end

game = Game.new
# Conecta com o Elasticsearch
client = Elasticsearch::Client.new log: true
# Verifica se aquele index ja existe
if !client.indices.exists? index: 'test'
  client.indices.create index: 'test'
end

id = 730
url = 'http://store.steampowered.com/api/appdetails/?appids=' + id.to_s
# Extrai dados da API
response = HTTParty.get(url)
response.parsed_response
data = response["730"]["data"]
game.put("name", data["name"])
game.put("description", data["detailed_description"])
# Insere os dados no Elasticsearch
client.update index: 'test', type: 'game', id: 730, body: game.to_json

```

No decorrer do desenvolvimento da primeira parte do projeto, houve uma mudança na política de privacidade da Steam, e isso modificou alguns dados que seriam extraídos, o



Figura 18 – Tela Inicial

principal foi o de número de donos. Antes de mudança a API do Steam Spy, providenciava um número quase exato do número de donos, após a mudança, agora a API providência apenas uma média estimada do número de donos. Isto alterar o projeto, já que agora não é possível mais fazer uma métrica exata da média de donos daquele gênero.

3.2 Plataforma Web

Para a plataforma web foi criado um protótipo que exibe como será a visualização dos dados, é também como será o fluxo dentro da plataforma. A figura 18 é a tela inicial da plataforma, dela pode-se ir para as métricas ou para a aba sobre. A figura 19 é a tela onde será apresentada as métricas, nesta página o usuário poderá definir os gêneros e as categorias que definirão o perfil do jogo. A figura 20 é a tela quando o usuário clica na aba *rankings*, nesta aba serão exibidas os *rankings* de jogos daquele determinado perfil. A figura 21 é a tela quando o usuário clica na aba sugestões, nesta aba serão exibidas algumas sugestões para a evolução daquele perfil. O protótipo poderá ser visualizado no seguinte *link*: <<https://marvelapp.com/3209ad6>>.



Figura 19 – Tela de Métricas



Figura 20 – Tela de Rankings

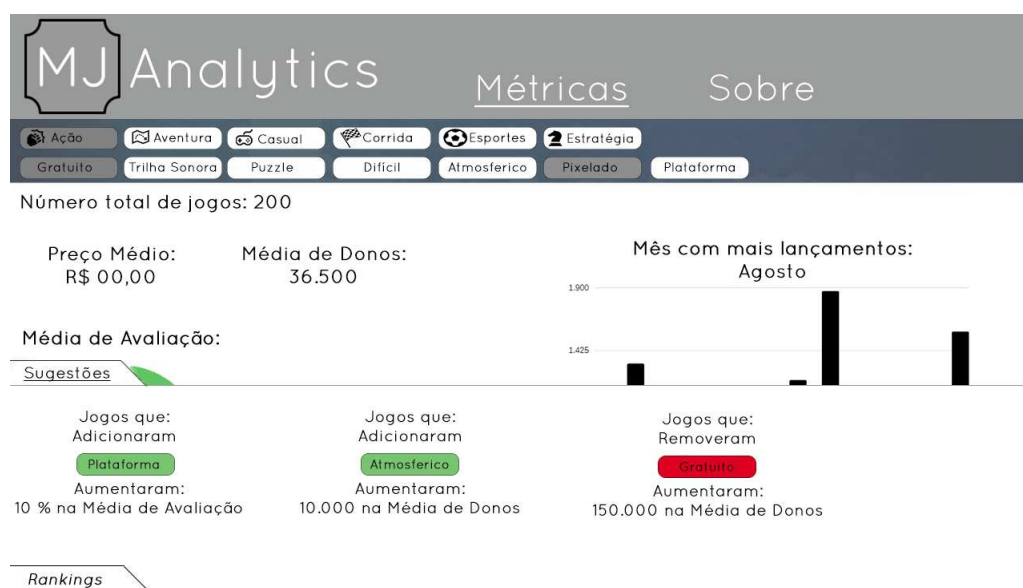


Figura 21 – Tela de Sugestões

4 Considerações Finais

No decorrer desta primeira parte, tive muita dificuldade em entender quais são as métricas e como iria montá-las, porém em conversas com o orientador e com membros de equipes de desenvolvimento de jogos, pude montar métricas que satisfaziam o escopo inicial do software. Na parte de extração foi possível a criação inicial do *handler* e de um *plugin*. Embora o que foi construído possua um caráter básico, a primeira parte do projeto foi um sucesso.

4.1 Trabalhos Futuros

Em relação aos trabalhos futuros, podemos listar as seguintes atividades:

1. Definição dos requisitos do sistema.
2. Definição de quais dados serão extraídos.
3. Definição das métricas do sistema.
4. Estudo sobre *game analytics* e *business intelligence*.
5. Implementação do protótipo da plataforma web.
6. Escrita do TCC1.
7. Implementação do *handler* e *main* do software de extração
8. Implementação dos *plugins* do software de extração.
9. Implementação da API de jogos.
10. Implementação da API de *rankings*.
11. Teste dos módulos da API.
12. Implementação da visualização das métricas.
13. Implementação da visualização dos *rankings*.
14. Implementação das sugestões da plataforma w
15. Escrita do TCC2.

O cronograma do trabalho está apresentado na tabela [6](#).

2018										
	MAR	ABR	MAI	JUN	JUL	AGO	SET	OUT	NOV	DEZ
1										
2										
3										
4										
5										
6										
7										
8										
9										
10										
11										
12										
13										
14										
15										
Legendas										
Tarefas realizadas										
Tarefas em andamento										
Tarefas não realizadas										

Tabela 6 – Cronograma do Trabalho

Referências

- ANDERS, D.; EL-NASR, M. S.; CANOSSA, A. Game analytics: Maximizing the value of player data. Londres, 2013. Citado na página 25.
- BECK, K. et al. *Manifesto Ágil*. 2001. Disponível em: <<http://agilemanifesto.org/iso/ptbr/manifesto.html>>. Citado na página 29.
- BOHANNON, J. Game-miners grapple with massive data. 2010. Citado na página 25.
- COLE, D. *DFC Intelligence*. 1994. Disponível em: <<https://www.dfciint.com>>. Citado na página 27.
- DAVEPORT, T. H.; HARRIS, J. G. Competing on analytics: The new science of winning. Boston, 2007. Citado na página 25.
- ECKERSON, W. The rise of analytic applications: Build or buy. 2002. Citado na página 24.
- ELASTIC. *Elasticsearch*. 2010. Disponível em: <<https://www.elastic.co/products/elasticsearch>>. Citado na página 31.
- ELASTIC. *Logstash*. 2010. Disponível em: <<https://www.elastic.co/products/logstash>>. Citado na página 32.
- GALYONKIN, S. *Steam Spy*. 2018. Disponível em: <<http://steamspy.com>>. Citado 3 vezes nas páginas 21, 27 e 36.
- LOSHIN, D. Business intelligence: The savvy manager's guide. 2012. Citado na página 23.
- MAKHMALI, J. *Whenever*. 2018. Disponível em: <<https://github.com/javan/whenever>>. Citado na página 32.
- MELLON, L. Apply metrics driven development to mmo costs and risks. República Tcheca, 2009. Citado na página 26.
- NEWZOO. *The Brazilian Gamer 2017*. 2017. Disponível em: <<https://newzoo.com/insights/infographics/the-brazilian-gamer-2017>>. Citado na página 21.
- NEWZOO. *The Chinese Gamer 2017*. 2017. Disponível em: <<https://newzoo.com/insights/infographics/chinese-gamer-2017>>. Citado na página 21.
- OHNO, T. O sistema toyota de produção. Porto Alegre, 1997. Citado na página 29.
- POSTGRESQL. *PostgreSQL*. 1997. Disponível em: <<https://www.postgresql.org/about/>>. Citado na página 31.
- SILVEIRA, D. *Crescimento Desenvolvedoras*. 2016. Disponível em: <<https://g1.globo.com/economia/negocios/noticia/numero-de-desenvolvedores-de-games-cresce-600-em-8-anos-diz-associacao.gh.html>>. Citado na página 21.

STEAMDB. *SteamDB*. 2010. Disponível em: <<https://steamdb.info>>. Citado na página 26.

STEAMWORKS. *Steam Web API*. 2018. Disponível em: <https://partner.steamgames.com/doc/webapi__overview>. Citado na página 35.

WIGGINS, A. *Clockwork*. 2018. Disponível em: <<https://github.com/Rykian/clockwork>>. Citado na página 33.