Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis
Campus "José Santilli Sobrinho"

JOÃO VICTOR VIEL PEREIRA PINTO

O USO DE MICROSSERVIÇOS PARA IMPLEMENTAÇÃO DE BUSINESS INTELLIGENCE

Assis/SP
2019

Fundação Educacional do Município de Assis
Instituto Municipal de Ensino Superior de Assis
Campus "José Santilli Sobrinho"

JOÃO VICTOR VIEL PEREIRA PINTO

O USO DE MICROSSERVIÇOS PARA IMPLEMENTAÇÃO DE BUSINESS INTELLIGENCE

Trabalho de conclusão de curso apresentado ao curso de Ciência da Computação do Instituto Municipal de Ensino Superior de Assis – IMESA e a Fundação Educacional do Município de Assis – FEMA, como requisito parcial à obtenção do Certificado de Conclusão.

Orientando: João Victor Viel Pereira Pinto
Orientador: Douglas Sanches da Cunha

Assis/SP
2019

FICHA CATALOGRÁFICA

PINTO, João Victor Viel Pereira.
O uso de microsserviços para a implementação de Businnes Intelligence /
João Victor Viel Pereira Pinto. Fundação Educacional do Município de Assis –FEMA – Assis, 2019.
32p.

1. Business Intelligence. 2. Microsserviços.

CDD: 005.74
Biblioteca da FEMA

O USO DE MICROSSERVIÇOS PARA IMPLEMENTAÇÃO DE BUSINESS INTELLIGENCE

JOÃO VICTOR VIEL PEREIRA PINTO

Trabalho de Conclusão de Curso apresentado ao Instituto Municipal de Ensino Superior de Assis, como requisito do Curso de Graduação, avaliado pela seguinte comissão examinadora:

Orientador: _____
Douglas Sanches da Cunha

Examinador: _____
Célio Desiró

Assis/SP
2019

DEDICATÓRIA

Dedico este trabalho aos meus pais que estiveram desde o inicio ao meu lado apoiando com todo sacrifício e também ao meu tio que durante todo o curso me apoiou das melhores maneiras. Dedico também aos meus amigos e professores que estiveram sempre ao meu lado colaborando para meu crescimento.

AGRADECIMENTOS

Primeiramente agradeço a **Deus**, que me deu forças e oportunidades todos esses anos para a realização de um bom curso, podendo aproveitar tudo oferecido.

Agradeço aos meus pais Celso Pereira Pinto e Adriana Paula Viel que sempre estão ao meu lado e sempre incentivando o meu crescimento.

Ao meu tio Celso, que me ofereceu toda ajuda financeira no decorrer do curso, cujo serei eternamente grato.

À todos os meus familiares, que sempre sentem orgulho da pessoa que me tornei.

Aos meus grandes amigos que adquiri no decorrer do curso, tais como Renato Virto Moreira, Matheus Cavalcanti, Lucas Correia e Gabriel Alan, onde espero que essa grande amizade continue por muitos anos.

Aos professores que estiveram mais próximos no decorrer do curso, obrigado por todo o apoio, Dra. Marisa Atsuko Nitto, Célio Desiró, Msc Diomara Barros, Msc Guilherme de Cleva Farto, Dr. Luiz Ricardo Begosso, Dr. Luiz Carlos Begosso, Dr. Almir Rogério Camolesi.

Ao meu grande amigo e professor Douglas, pessoa que admiro e aprendi muito, obrigado por ser a pessoa que que viu potencial em mim e deu uma oportunidade.

E por fim, agradeço a todos que colaboram no meu crescimento no decorrer do curso e execução deste trabalho.

RESUMO

Atualmente, empresas, universidades ou instituições estão preocupados com seus negócios, atitudes a tomar, onde investir dinheiro. Com isso, faz se a necessidade de informações confiáveis, onde possam ser entendidas de forma simples e ajudar a tomada de decisão. Visto essa necessidade, o conceito de Business Intelligence vem para auxiliar esses órgãos e seus gestores a controlar o que ocorre dentro de suas empresas, principalmente no momento de se tomar uma decisão.

O objetivo deste trabalho é realizar um estudo exploratório relacionado aos conceitos de Business Intelligence, e na pratica, implementar uma plataforma em formato de API e microsserviços para que aplicações que já funcionam em meio a esse cenário, possam implementar ou consumir esses recursos para atender as necessidades dentro deste cenário.

Palavras-chave: Business Intelligence, Microsserviços.

ABSTRACT

Today, business, universities, or institutions is concerned about its business, attitudes, where to invest money. This makes the need for reliable information where it can be understood simply and help in decision making. Given this need, the concept of Business Intelligence comes to help these agencies and their managers control what happens within their companies, especially when making a decision.

The objective of this paper is to conduct an exploratory study related to the concepts of Business Intelligence, and in practice, implement a platform in API and microservices format so that applications that already work in this scenario, can implement or consume these resources to meet the needs within this scenario.

Keywords: Business Intelligence, Microservices.

LISTA DE ILUSTRAÇÕES

SUMÁRIO

1. INTRODUÇÃO

Por muito tempo, as empresas tomaram decisões importantes baseadas apenas na intuição de seus líderes, o que tinha como guia suas experiências passadas e conhecimento do mercado. Apesar de impérios terem sido construídos assim, o avanço da tecnologia e os resultados que isso trouxe (aumento da competitividade, surgimento de novos mercados) tornou o cenário de negócios muito mais competitivo.

Assim, fazia necessário encontrar uma forma mais inteligente e consistente de tomar decisões, uma que não dependesse tanto da capacidade dedutiva de gestores e diretores executivos para funcionar de forma sistemática e escalável. Entendendo-se que boas escolhas dependem das informações certas, surgiu uma série de conceitos que permitem coletar, gerenciar e distribuir os dados de uma empresa para transformá-los em *insights*. Esses conceitos formam o que conhecemos hoje como *Business Intelligence (BI).*

Devido à grande competitividade e à exigência do mercado em empresas mais preparadas para disputarem entre si, o BI é uma forma inteligente de otimizar falhas e manter o negócio em destaque (SITEWARE, 2018).

BI serve para analisar os fenômenos acerca do negócio. Isso significa que *Business Intelligence* precisa ser uma plataforma capaz não só de aglutinar as informações transacionais, mas também de exibi-las de forma contextual, fazendo com que fenômenos escondidos se tornem visíveis (BRAGUITTONI, 2017). A diminuição nos custos também acontecerá porque ele terá mais agilidade, processos otimizados e perderá menos tempo com análises infrutíferas. E, caso houver um equívoco, a medição de desempenho e resultados, outra premissa do *Business Intelligence*, facilitará a correção dos rumos do projeto em questão (FIA, 2018).

Pensando em ferramentas de BI que sejam mais flexíveis, escaláveis e com manutenção mais simples do que as arquiteturas de sistemas monolíticas normalmente utilizadas (OPUS, 2018). Pode ser utilizada a arquitetura de microsserviços para viabilizar o contexto de desenvolvimento e também de uso.

Este projeto tem o objetivo de implementar uma ferramenta baseada na metodologia de *Business Intelligence*, apoiada pela arquitetura de microsserviços com uma abordagem de desenvolvimento de um aplicativo único como uma suíte de pequenos serviços, cada um executando seu próprio processo e se comunicando através de mecanismos leves, através de uma API REST com recursos HTTP.

1.1 OBJETIVOS

O objetivo geral dessa pesquisa é o de explorar os conceitos de *Business Inteligence*, com o foco nas áreas administrativas e acadêmicas de pequenas ou grandes universidades e instituições. Além deste estudo exploratório, especificamente pretende-se modelar e implementar uma plataforma amparada pela arquitetura de microsserviços, onde a mesma possa realizar uma mineração de dados relevantes para a instituição.

1.2 JUSTIFICATIVA

Houve um tempo em que a intuição e o achismo eram a base para tomar decisões em uma empresa. Sempre que surgia uma oportunidade ou dificuldade, os gestores precisavam arriscar, por não ter ferramentas adequadas para a tomada de decisão (ALVES, 2017).

Atualmente com a quantidade de informação gerada diariamente em empresas e instituições, não há total conhecimento sobre elas, com isso, as empresas apostam no uso de tecnologias para ter total conhecimento dessas informações (GOIS, 2017).

Com uma plataforma de *Business Intelligence* os usuários conseguem extrair informações e elaborar relatórios por conta própria, sem dependência da área de TI. Podem até mesmo acessar essas informações a qualquer hora e em qualquer lugar através dos aplicativos mobile. Os executivos já conseguem tirar bons resultados, mas nem todo o potencial é aproveitado. Com isso, BI faz necessário para conhecimento das informações geradas diariamente.

1.3 MOTIVAÇÃO

Estamos vivendo em uma época que a informação é extremamente valiosa para qualquer empresa, onde as tomadas de decisões não podem ser realizadas sem um auxilio que prove que realmente se tornará necessário ou que dará certo. Os conceitos de *Business Intelligence* abordam justamente isso, onde, com uma análise ou mineração realizada de forma correta, pode trazer informações que realmente, tornam-se importantes para um gerente ou administrador de uma empresa na tomada de decisão.

Para esta implementação, apoiada pela arquitetura de microsserviços, facilitará na implantação e desenvolvimento da ferramenta, diferente de aplicações monolíticas geralmente utilizadas, onde, o conceito de separar a aplicação em pequenos serviços viabiliza o processamento de tantas informações.

1.4 PERSPECTIVAS DE CONTRIBUIÇÃO

Este trabalho contribuirá para projetos futuros que abordem os contextos de *Businnes Intelligence* e suas implementações. Por ter a proposta de desenvolvimento de uma ferramenta com o uso da arquitetura de microsserviços e uma API REST, podem contribuir também para facilitar a implementação e recuperação de informações para futuros trabalhos propostos nesse ambiente.

1.5 METODOLOGIA

Para alcançar tais objetivos, a metodologia inicialmente amparada pela revisão bibliográfica de caráter exploratório, com o intuito de proporcionar maior familiaridade com o tema. Adotar-se-ão fontes confiáveis como artigos técnico-científicos, monografias, dissertações, teses, livros e capítulos de livros, resumos e artigos de periódicos, além de páginas da Web de conteúdo seguro. Após a condução da revisão da literatura quanto aos tópicos que relacionam com este trabalho, será validado a necessidade, bem como o interesse e/ou a relevância em aprofundar a verificação do estado da arte por meio de um Estudo de Mapeamento Sistemático.

Constitui, como etapa da metodologia deste trabalho, a concepção e a modelagem (design) arquitetural de uma abordagem composta por plataformas, linguagens de programação e demais tecnologias envolvidas. A definição de uma arquitetura tecnológica objetiva apoiar a implementação da plataforma de objetos de aprendizagem e contribuir com a aplicabilidade e a avaliação experimental dos conceitos explorados teoricamente. Desta forma, não somente o contexto do trabalho será consolidado, como também estratégias e métodos que se referem à engenharia de software, bem como aos princípios de arquitetura e de desenvolvimento de aplicações.

Por fim, os elementos envolvidos na pesquisa, como tecnologias, abordagens, arquiteturas, plataformas e demais resultados e contribuições, serão relatados em forma de artigos e materiais de apoio, fornecendo uma base fundamental.

2. *BUSINESS INTELLIGENCE (BI)*

O conceito surgiu na década de 1990, e se refere a processos de organização, coleta, análise, monitoramento e compartilhamento das informações que são a base da gestão de negócios. O objetivo principal é facilitar a leitura e interpretação dos dados, identificar novas oportunidades de negócio e apoiar a empresa nas decisões e estratégias, fazendo com que melhore sua competitividade no mercado (AUSLAND, 2015).

Esse processo passa por coleta, organização e análise dos dados, elaboração de relatórios ou *dashboards* e todo o acompanhamento e atualização. Esses são os fundamentos, a base para entender como tudo funciona. Entendendo o conceito, a empresa e seus gestores estarão com o poder de tomada de decisão orientada a dados e evidências (PITON, 2017).

BI pode ser entendido como, um processo que auxilia a transformação dos dados brutos em uma empresa, em informações compreensíveis para o analista ou gestor, e significativas para posteriormente ocorrer uma análise do negócio. As empresas que trabalham com inteligência empresarial são capazes de ajudar na organização de um grande número de dados desestruturados. Essa organização é capaz de auxiliar a empresa a criar novas estratégias para seu negócio (PITON, 2017).

2.1 VANTAGENS

Quando uma empresa opta por um sistema baseado em BI, ela dedica um tempo maior à análise crítica dos resultados obtidos, ao invés de gastar tempo com a coleta de dados e a geração de relatórios. Estes podem ser gerados com apenas uma ação.

O BI aumenta o conhecimento dos gestores acerca da empresa, corrigindo até mesmo erros e falhas que, através de um relatório manual, podem ser emitidos ou passar despercebidos. Por esse motivo, muitos temem que o BI venha substituir o trabalho humano de análise de negócios, porém, é necessário ter a visão que BI é apenas uma ferramenta para facilitar e otimizar o trabalho humano, tornando não substituível completamente (ADV, 2016).

O desenvolvimento e eficiência da empresa é notável com o uso de BI, a empresa passa a ser mais eficiente na solução de problemas e análises de futuros investimentos e

estratégias, focando em sua real necessidade, sem perda de tempo com rotinas repetitivas e cansativas, que são completamente otimizado pela ferramenta.

Dentro deste contexto, podem ser citados algumas das vantagens que uma empresa pode adquirir com o uso de BI.

- **Aumento nas vendas:** A inteligência de negócios está bastante ligada com o departamento comercial e a melhoria dos processos de vendas. O BI analisa os clientes, o período ou duração das transações dentro de um grande número delas dentro do funil de vendas. Uma visão ampla de todos os componentes da venda vai fazer com que as mudanças no processo sejam contínuas. A experimentação será com mais fundamento e com experiência de acertos (ADV, 2016).
- **Maior eficiência no marketing:** Todo departamento de marketing precisa encontrar novas maneiras de auxiliar a empresa a crescer. É cada vez mais necessário ter a capacidade de analisar os retornos das campanhas, rendimentos, custos e outros detalhes que aumentem o retorno dos investimentos. As capacidades analíticas da inteligência de negócios podem fornecer para o marketing as diretrizes para determinar onde cada centavo do departamento de marketing será mais bem gasto (ADV, 2016).
- **Acompanhamento em tempo real:** Não basta saber quanto dinheiro vai entrar, mas também quando ele vai entrar. Quanto antes e mais precisa for a previsão das entradas, melhor será o planejamento estratégico da empresa. Usando os dados de Business Intelligence, o gestor pode coletar informações indispensáveis para prever o comportamento do fluxo de caixa. As planilhas do Excel atuais já não têm capacidade de lidar com tantas variáveis, principalmente com a rapidez necessária (ADV, 2016).
- **Auxílio para a área de compras:** O setor de compras é outro departamento que pode realmente beneficiar-se do BI. Um software de Business Intelligence pode ajudar o departamento a gerenciar coisas como fornecedores, fontes de matéria prima, estoques e muito mais. A análise desses dados passa a ser feita por um processo mais automático e menos subjetivo. Como o BI fornece métricas objetivas e analíticas, é possível ver onde os problemas realmente estão e trabalhar em conjunto para melhorar os processos de negócios (ADV, 2016).

2.2 ARQUITETURA

Para um melhor entendimento sobre BI, é necessário analisar e compreender sua arquitetura. Por definição, o BI não está restrito a uma tecnologia específica e, por isso, a concepção de sua dimensão muitas vezes torna mais difícil o entendimento.

O sistema de BI possui um acervo de possibilidades e requer uma percepção bem apurada da situação em questão para sua correta implementação. Ele contém certa abstração em seu conceito, permitindo, assim, flexibilidade e adaptações a cada novo projeto. Sua estrutura final vai depender do contexto que está inserida a solução. Em uma arquitetura genérica de BI, ou seja, aquela onde não é aplicada um estrutura pensada exclusivamente para um caso, a composição é simples. Em uma visão macro, o BI precisa necessariamente de três marcos: Fonte de dados, Consolidação e Decisão. A Figura 1 ilustra a visão geral de uma arquitetura de Business Intelligence (ELIAS, 2015).

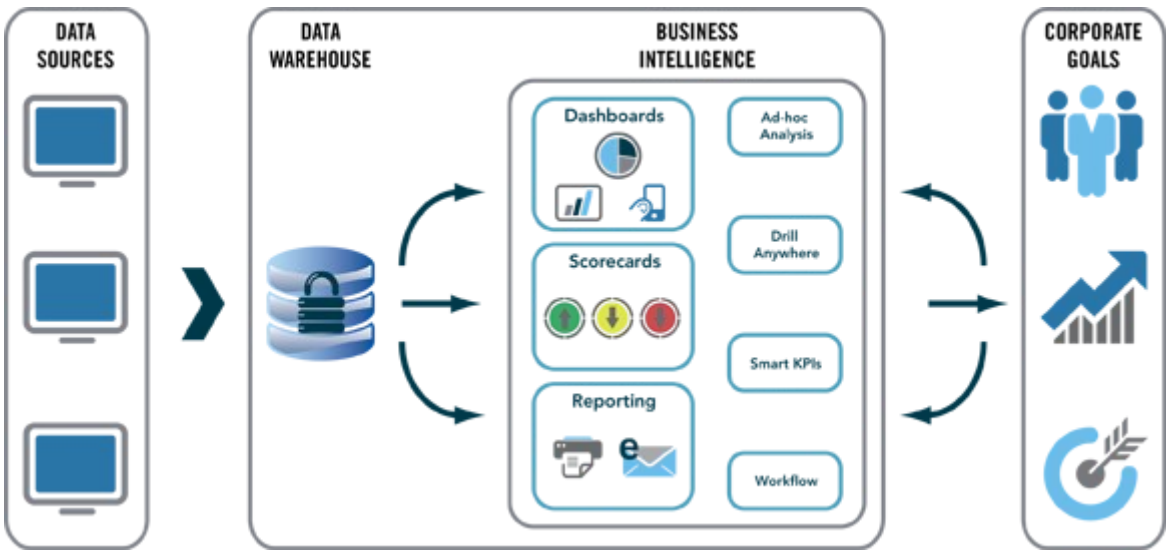

Figura 1: Arquitetura de Business Intelligence (ELIAS, 2015).

Conforme mostra a Figura 1, com uma arquitetura bem planejada e com essas três fases bem delineadas, se pode dar o correto encaminhamento para um projeto de BI. Localizando as origens das informações, consolidando os dados e os disponibilizando ao decisor de forma efetiva.

3. MICROSSERVIÇOS

O termo Arquitetura de microsserviços surgiu nos últimos anos para descrever uma maneira específica de desenvolver software como suítes de serviços com deploy independente (LEWIS, 2015).

Os microsserviços são uma abordagem de arquitetura para a criação de aplicações. O que diferencia a arquitetura de microsserviços das abordagens monolíticas tradicionais é como ela decompõe a aplicação por funções básicas. Cada função é denominada um serviço e pode ser criada e implantada de maneira independente. Isso significa que cada serviço individual pode funcionar ou falhar sem comprometer os demais (REDHAT, 2018). A Figura 2 exemplifica uma comparação entre uma estrutura monolítica e uma de microsserviços.

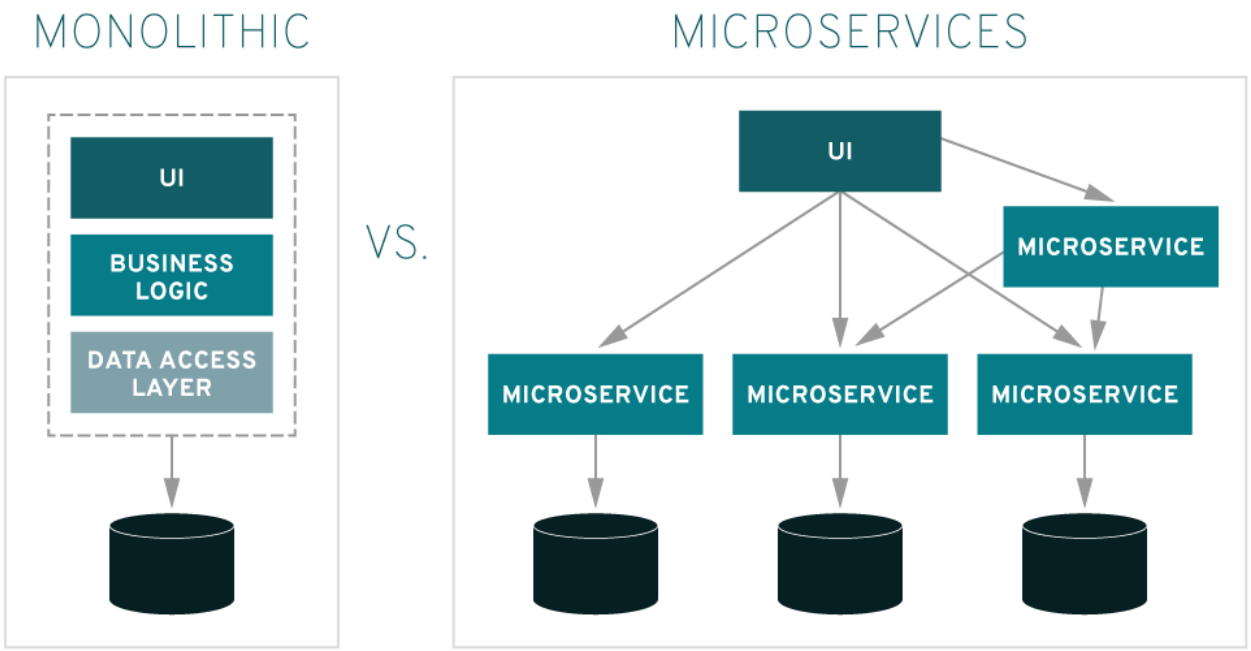

Figura 2: Monolítico VS Microsserviços (OPUS, 2018).

Portanto, um microsserviço é uma função essencial de uma aplicação e é executado independentemente dos outros serviços. No entanto, a arquitetura de microsserviços é mais complexa do que o mero acoplamento flexível das funções essenciais de uma aplicação. Trata-se da restruturação das equipes de desenvolvimento e da comunicação entre serviços de modo a preparar a aplicação para falhas inevitáveis, escalabilidade futura e integração de funcionalidades novas (OPUS, 2018).

Nos primórdios do desenvolvimento de aplicações, até mesmo as alterações mais insignificantes em uma aplicação pronta exigiam uma atualização da versão de atacado, com um ciclo próprio de garantia da qualidade (QA). Isso, provavelmente, atrasava o trabalho de muitas subequipes. Muitas vezes, essa abordagem é chamada de “monolítica” porque o código-fonte da aplicação toda era incorporado em uma única unidade de implantação, como .war ou .ear. Se a atualização de alguma das partes causasse erros, era necessário desativar a aplicação inteira, reverter a escala e corrigir o problema. Embora essa abordagem ainda seja viável para aplicações menores, as empresas em ampla expansão não podem se dar ao luxo de sofrer com tempo de inatividade (SITEWARE, 2018).

A arquitetura orientada a serviço serve pra resolver essa questão, pois estrutura as aplicações em serviços distintos e reutilizáveis que comunicam por meio de um *Enterprise Service Bus* (ESB). Nessa arquitetura, os serviços individuais, cada um deles organizado em torno de um processo de negócios específico, aderem a um protocolo de comunicação que quando reunidos, esse pacote de serviços, integrados por meio de um ESB, formam uma aplicação (REDHAT, 2018).

3.1 ARQUITETURA SOA E MICROSSERVIÇOS

Os microsserviços podem comunicar *entre si*, normalmente de maneira *stateless* , ou seja, que não mantém estados de variáveis, portanto, as aplicações criadas dessa maneira podem ser mais tolerantes a falhas e depender menos de um único ESB. Além disso, as equipes de desenvolvimento podem escolher as ferramentas que desejarem, pois os microsserviços podem se comunicar por meio de interfaces de programação de aplicações (APIs) independentes de linguagem.

Levando em consideração a história da SOA, os microsserviços não são uma ideia completamente nova. Porém, eles tornaram mais viáveis graças aos avanços nas tecnologias de conteinerização. Com os containers Linux, agora é possível executar várias partes de uma aplicação de maneira independente no mesmo hardware e com um controle muito maior sobre os componentes individuais e ciclos de vida (OPUS, 2018).

3.2 BENEFÍCIOS

Com os microsserviços, as equipes e tarefas rotineiras podem tornar mais eficientes por meio do desenvolvimento distribuído. Além disso, é possível desenvolver vários microsserviços ao mesmo tempo. Isso significa que é possível ter mais desenvolvedores trabalhando simultaneamente na mesma aplicação, o que resulta na redução do tempo gasto com desenvolvimento. Assim, há uma lista de benefícios que a arquitetura de microsserviços oferece:

- **Lançamento no mercado com mais rapidez:** Os ciclos de desenvolvimento são reduzidos, com isso, a arquitetura é compatível com implantações e atualizações mais ágeis (REDHAT, 2018).
- **Altamente escalável:** Conforme a demanda dor determinados serviços aumentarem, é possível realizar a implantação em vários servidores e infraestruturas para atender as necessidades (REDHAT, 2018).
- **Resiliente:** Como os serviços são independentes, se construídos corretamente, não afetam uns aos outros, todo o restante da aplicação continua em funcionamento, diferente da arquitetura SOA (REDHAT, 2018).
- **Acessível:** Como a aplicação é decomposta em partes menores, é mais fácil entender e aprimorar determinadas partes, o que resulta em um ciclo de desenvolvimento mais rápido (REDHAT, 2018).
- **Mais open source:** Por se tratar de APIs, os desenvolvedores tem liberdade para escolher a melhor linguagem e tecnologia para a função necessária (REDHAT, 2018).

4. FERRAMENTAS

No decorrer deste projeto foram usados alguns framekorks para a implementação de uma plataforma de BI, sendo que, o conceito de framework entende-se como um conjunto de técnicas, ferramentas ou conceitos pré-definidos usados para resolver um problema de um projeto ou domínio específico. É, basicamente, uma estrutura de trabalho que atua com funções pré-estabelecidas que se adaptam à situação e à organização em questão.

Para a construção do projeto foram selecionadas algumas tecnologias para sua execução. Como linguagem de programação foi utilizado Java na versão 8, pois a quantidade de informações que podem trafegar entre os serviços, é necessário uma linguagem robusta, e fortemente tipada para facilitar a implementação em alguns aspectos. A IDE Eclipse Photon foi utilizada para a implementação.

Foram selecionados alguns *frameworks*, como já citado anteriormente, tal como Spring Boot em sua versão 2.1.5, e alguns recursos oferecidos por ele como: Spring Data, JDBC Template, Devtools.

O banco de dados utilizado foi Postgresql, considerando leve e rápido, onde a criação de *views* e manipulação dos dados é feita de forma simples. Acompanhado pela ferramenta PgAdmin III, para o gerenciamento do banco de dados.

O framework Swagger foi utilizado para a documentação e testes das API's, oferecendo um bom recurso visual, e facilidade nos teste dos serviços, dando a oportunidade de simulação de diversos cenários.

O Servidor de aplicação em ambiente de desenvolvimento foi o TomCat em sua versão 9, oferecendo praticidade e deixando o projeto mais leve, já que este consegue executar vários projetos ao mesmo tempo sem ocupar muito processamento.

4.1 SPRING FRAMEWORK

O Spring é um framework Java criado com o objetivo de facilitar o desenvolvimento de aplicações, explorando, para isso, os conceitos de Inversão de Controle e Injeção de Dependências. Dessa forma, ao adotá-lo, temos à disposição uma tecnologia que fornece não apenas recursos necessários à grande

parte das aplicações, como módulos para persistência de dados, integração, segurança, testes, desenvolvimento web, como também um conceito que nos permite criar soluções menos acopladas, mais coesas e, consequentemente, mais fáceis de compreender e manter (DEVMEDIA, 2018).

O Spring foi criado por causa das dificuldades que os programadores enfrentavam ao criar determinado tipo de aplicação, mais precisamente, aplicações corporativas. Na época, a plataforma Java voltada para isso, de nome J2EE, ainda era jovem, com ótimas ideias para a construção de aplicações leves, distribuídas, com um amplo leque de opções/ferramentas, mas com algumas limitações. Essas limitações levavam a uma programação dependente de muitas interfaces e com muitas configurações. Ao final, era comum ter uma solução pesada e que trazia consigo muito mais do que o que realmente era necessário. Com isso o Spring surge para facilitar o desenvolvimento de aplicações distribuídas (SPRING, 2017).

4.2 SPRING BOOT

Spring Boot é um projeto da Spring que veio para facilitar o processo de configuração e publicação das aplicações. É escolhido os módulos que deseja através dos starters que inclui no pom.xml do projeto. Eles, basicamente, são dependências que agrupam outras dependências. Apesar do Spring Boot, através da convenção, já deixar tudo configurado, nada impede que possa criar customizações caso sejam necessárias (SPRING, 2019).

Trata-se de mais um framework, mas talvez a melhor denominação seja micro framework. Seu objetivo não é trazer novas soluções para problemas que já foram resolvidos, mas sim reaproveitar estas tecnologias e aumentar a produtividade do desenvolvedor. Sendo assim, trata-se também de uma excelente ferramenta que pode-se adotar na escrita de aplicações que fazem uso da arquitetura de microsserviços.

Se desenhar um diagrama arquitetural do Spring Boot, este seria uma fina camada sobre tecnologias já consagradas pelo mercado, tal como podemos verificar na Figura 1. A grande mudança está no modo como agora são empacotados e acessados estas soluções (DEVMEDIA, 2015).

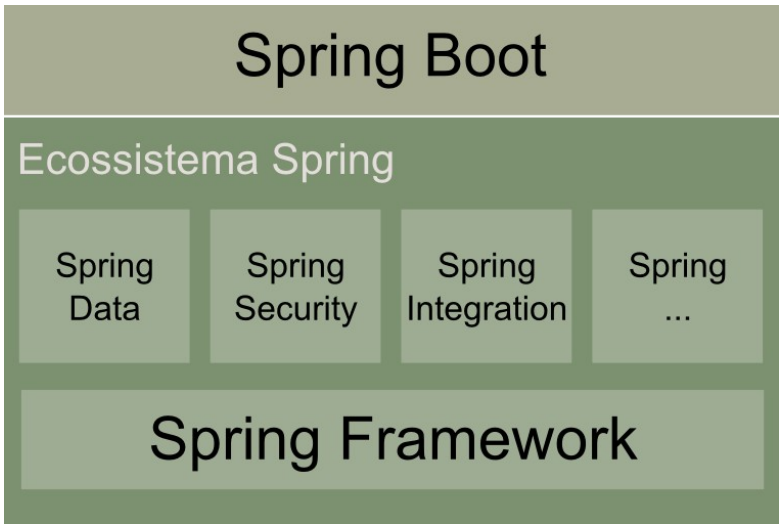

Figura 3: Posicionamento do Spring Boot no ecossistema Spring (SPRING, 2019).

”O Spring Boot facilita a criação de aplicativos baseados em Spring autônomos e de produção que você pode executar. Adotamos uma visão opinativa da plataforma Spring e de bibliotecas de terceiros para que você possa começar com o mínimo de trabalho. A maioria dos aplicativos Spring Boot precisa de uma configuração de Spring muito pequena.” (SPRING, 2018).

4.3 SPRING CLOUD

Spring Cloud é capaz de coordenar a implementação de microsserviços, simplificando a construção de sistemas distribuídos. A construção destes sistemas distribuídos não precisa ser complexa e propensa a erros. O Spring Cloud oferece um modelo de programação simples e acessível para os padrões de sistema distribuídos mais comuns, ajudando os desenvolvedores a criar aplicativos resilientes, confiáveis e coordenados. O Spring Cloud é desenvolvido com base no Spring Boot, facilitando o trabalho dos desenvolvedores e tornando-os produtivos rapidamente (SPRING, 2018). A figura 4

exemplifica a arquitetura com o uso de Spring Cloud para a implementação de microsserviços.

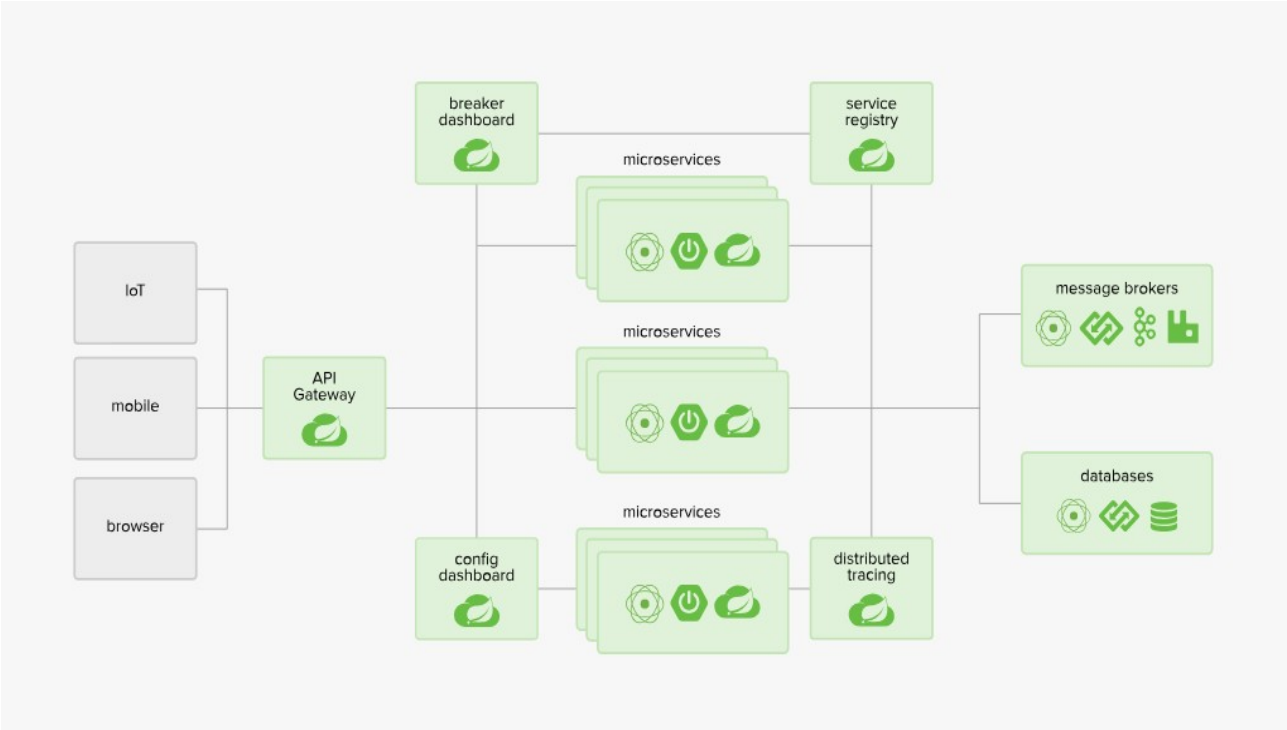

Figura 4: Arquitetura de microsserviços com Spring Cloud (SPRING, 2018).

Sendo assim, este projeto irá realizar a implementação com arquitetura semelhante a figura 4, voltado para os conceitos de BI.

4.4 DOCKER

Docker é uma plataforma Open Source escrito em Go, que é uma linguagem de programação de alto desempenho desenvolvida dentro do Google, que facilita a criação e administração de ambientes isolados.

Docker não é um sistema de virtualização tradicional. Enquanto em um ambiente de virtualização tradicional se tem um S.O. completo e isolado, dentro do Docker se tem recursos isolados que utilizando bibliotecas de kernel em comum (entre host e container) (DIEDRICH, 2015). A figura 5 ilustra tal diferença.

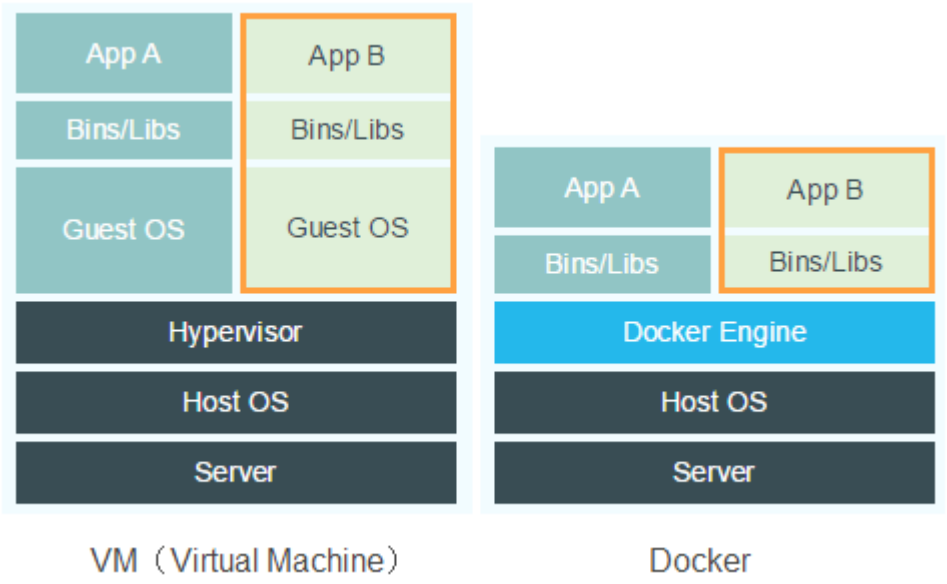

Figura 5: Diferença entre Virtualização e Docker (DOCKER, 2019).

Como mostra a figura 5, Docker possibilita o empacotamento de uma aplicação ou ambiente inteiro dentro de um container, e a partir desse momento o ambiente inteiro torna-se portável para qualquer outro Host que contenha o Docker instalado. Isso reduz drasticamente o tempo de deploy de alguma infraestrutura ou até mesmo aplicação, pois não há necessidade de ajustes de ambiente para o correto funcionamento do serviço, o ambiente é sempre o mesmo, configure-o uma vez e replique-o quantas vezes quiser, o que torna um cenário adequado para o deploy de aplicações distribuídas (DOCKER, 2019).

5. IMPLEMENTAÇÃO

No decorrer deste trabalho foram implementadas as funcionalidades usando as tecnologias e conceitos citados nos capítulos anteriores. Os conceitos de *Business Intelligence* foram aplicados na criação de serviços *REST e* microsserviços para que aplicações *Front-End* consumam estes, para a criação de *Dashboards* dinâmicos.

A aplicação foi construída com o propósito inicial de atender necessidades de uso do conceito de BI para a área acadêmica. Pensando até em um uso interno na própria faculdade, onde os diretores, coordenadores e demais usuários dos sistemas acadêmicos necessitam de informações atualizadas; em qualquer lugar, de forma simples e compreensível, para a tomada de decisões e conhecimento exato sobre o que se passa dentro da faculdade.

Para que as informações sejam disponibilizadas de forma dinâmica para a criação dos *Dashboards*, foi implementado utilizando o conceitos de *views* no banco de dados, sabendo que, uma *view* é uma maneira alternativa de observação de dados de uma ou mais tabelas, que compõem uma base de dados. Pode ser considerada como uma tabela virtual ou uma consulta armazenada (DEVMEDIA, 2015).

Através deste recurso, as informações são disponibilizadas de forma dinâmica, onde, sempre que o usuário encontre necessidade de novas informações, para a montagem de *Dashboards*, gráficos e tabelas, será sempre necessário acionar um responsável pelo banco de dados para a criação da *view* que disponibilizará os dados necessários.

A aplicação mapeia todas as *views* criadas no banco de dados em que esta conectada, e disponibiliza dinamicamente em forma de lista, as suas informações e detalhes. Ao recuperar a *view* que deseja, é possível visualizar todos os dados recuperados de uma ou mais tabelas, ou seja, sempre que criado, as informações já estarão disponibilizadas para que o usuário organize seu *Dashboard*, gráficos e tabelas.

5.1. ESTRUTURA DO PROJETO

Os projetos foram criados através da funcionalidade que o framework Spring Boot oferece para projetos Maven, conforme mostra a figura 6. A ferramenta para a criação destes projetos está disponível em **start.spring.io.**

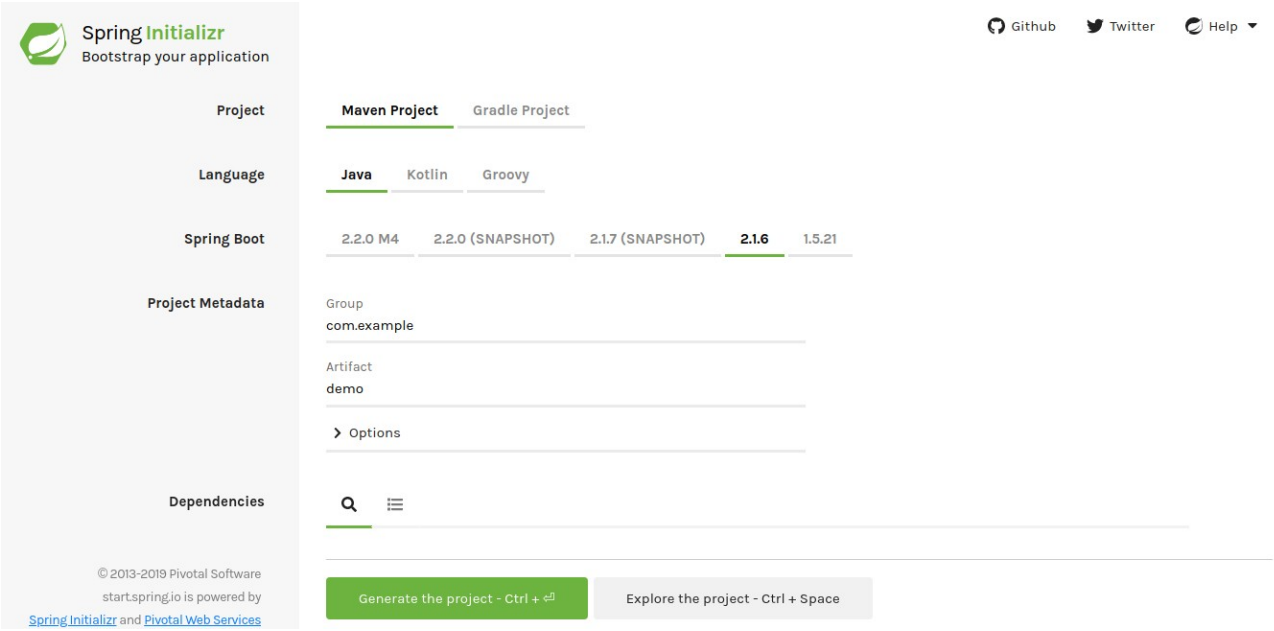

Figura 6: Iniciando um projeto com Spring Boot (SPRING, 2019).

Este projeto esta baseado na arquitetura de microsserviços, assim, foi proposto a criação de quatro projetos independentes, quando estes estiverem em pleno funcionamento, possam funcionar de forma que, se um serviço for interrompido, os outros consigam continuar fornecendo recursos e informações para sua utilização. A figura 7, mostra a estrutura e nome dos projetos criados.

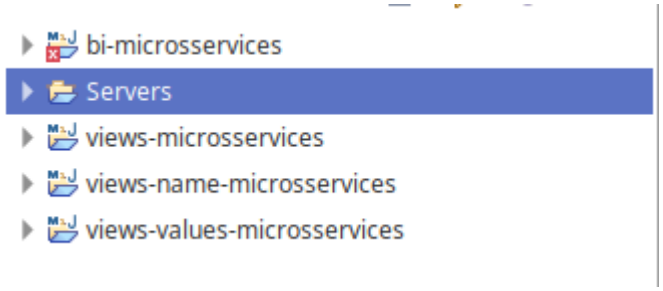

Figura 7: Estrutura dos projetos criados

Conforme apresentado na figura 7, foram criados quatro projetos. Entre eles está o *bi-microsservices. E*ste projeto é responsável por todos os serviços que realizam a persistência dos dados relacionados aos *Dashboards,* como a *view* que irá disponibilizar as informações, nome e características, posições de gráficos, *cards,* tabelas e a qual usuário pertence. Esses serviços, possuem métodos como GET, POST, PUT e DELETE, como mostram as figura 8 e 9, a documentação dos serviços disponíveis.

Figura 8: Serviços disponibilizados pela aplicação.

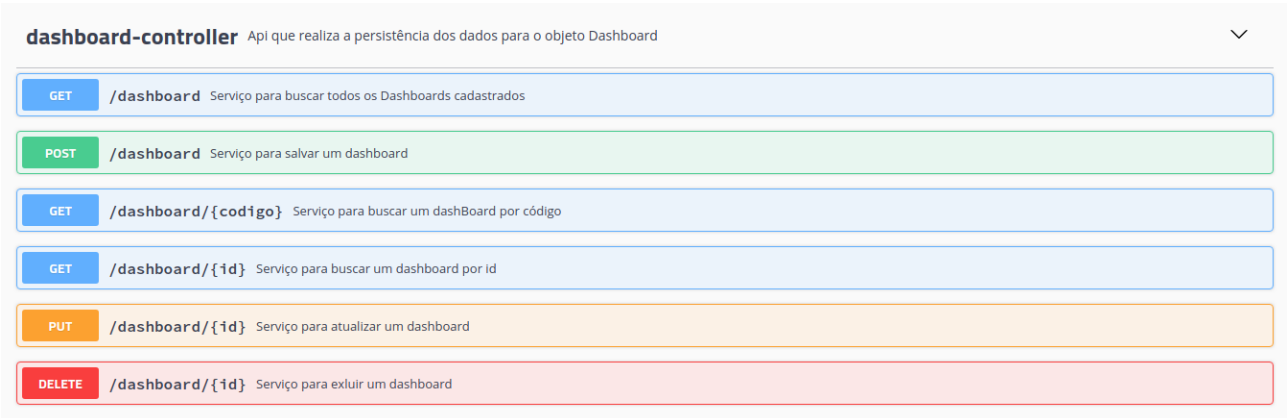

Figura 9: Métodos para criação de um Dashboard.

O projeto *views-microsservices*, é o responsável por disponibilizar um microsserviço que mapeia todas as *views* cadastradas no banco de dados e as disponibiliza para o usuário, de forma que ele possa escolher qual destas deseja recuperar as informações.

O projeto *views-name-microsservices*, disponibiliza um microsserviço para a busca unificada de uma *view*, ou seja, por seu identificador. Todos os detalhes e características são listados neste recurso.

O último projeto, chamado *views-values-microsservices,* é o principal microsserviço disponibilizado, e por meio dele é possível recuperar as informações dinamicamente de uma *view,* passando esta como parâmetro, para o preenchimento dos componentes criados no *Dashboard.* A figura 10 demonstra a funcionalidade de cada microsserviço e projetos citados.

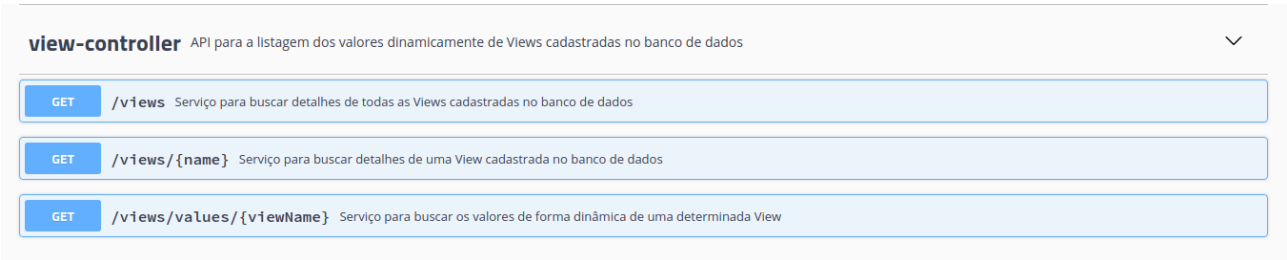

Figura 10: Microsserviços para recuperação das views.

A recuperação dinâmica dos dados só é possível com uma implementação utilizando o conceito de chave e valor, onde foi utilizado o recurso da interface Map. Assim, o nome de cada campo recuperado pela *view,* é atribuído a uma chave e seu respectivo valor.

O JDBC Template (interface de comunicação com banco de dados) disponibilizado pelo framework Spring Boot faz possível a utilização de recursos, que facilitam a recuperação destas informações, que nunca sabe-se exatamente o que irá retornar. A figura 11 mostra um exemplo de implementação, buscando os dados de uma *view* dinamicamente.

```
public List<Map<String, Object>> findValuesView(String viewName) throws SQLException{

    StringBuilder sql = new StringBuilder();
    sql.append("select * from ");
    sql.append(viewName);

    List<Map<String, Object>> rows = this.jdbcTemplate.queryForList(sql.toString());

    return rows;
}
```

Figura 11: Exemplo de implementação.

Para otimizar a aplicação, a API disponibiliza um serviço para criação de gráficos, tais como: gráficos de pizza, barras e linhas. O serviço utilizado, necessita apenas do nome de uma *view* e qual tipo de gráfico deseja gerar, que serão passados como parâmetro para verificar através de suas características, se é possível gerar o gráfico solicitado com os dados da *view* informada. Cada gráfico possui um modelo próprio que retorna um objeto dinâmico de acordo com o que solicitado.

CONCLUSÃO

Após a conclusão do projeto, os microsserviços das *views* dinâmicas serão disponibilizados para serem hospedados em um servidor de Docker juntamente com os serviços da API, onde ficaram disponíveis para utilização interna da faculdade, que desenvolverá aplicações para consumir esses recursos, podendo colocar em prática os conceitos abordados durante este trabalho, por exemplo, um diretor em um congresso tenha dados importantes em relação a sua instituição. Além disso, conclui-se que o conceito de Business Intelligence pode fazer com que as operações e tomadas de decisão dentro de uma empresa ou instituição sejam assertivas, já que os dados são transformados em informação.

Utilizando os conceitos de microsserviços, facilita a disponibilização e manutenção desta aplicação, para que ofereça recursos à qualquer sistema que deseje consumir tais. Onde, caso ocorra a falha em uma aplicação de microsserviço, as outras ainda estarão em pleno funcionamento, pois trabalham de forma independente.

A partir deste estudo exploratório, é possível buscar novos conceitos e tecnologias para que os dados disponibilizados sejam ainda mais assertivos e sem a necessidade de utilizar um número grande de *views.*

Visto isso, pode-se implantar um novo conceito, Data Warehouse. Um Data Warehouse é um repositório central de informações que pode ser analisado para tomar decisões mais embasadas. Os dados fluem de sistemas transacionais, bancos de dados relacionais e de outras fontes para o data warehouse, normalmente com uma cadência regular (AMAZON, 2018).

Com isso, os dados podem ser analisados por esta ferramenta, sem a necessidade de um analista para realizar a criação das *views,* fazendo com que a aplicação seja mais eficiente, simples e rápida. Isso irá gerar uma melhor experiência e confiança para o usuário final.

REFERÊNCIAS

ADV. **Veja 5 vantagens do Business Intelligence para sua empresa.** 2016. Disponível em <http://www.advtecnologia.com.br/veja-5-vantagens-do-business-intelligence-para-sua-empresa/>, Acesso em 27/02/2019.

ALVES, C. **Business Intelligence: tudo que você precisa saber!.** 2017. Disponível em <https://blog.bi9.com.br/business-intelligence/>, Acesso em 27/10/2018.

AMAZON. **O que é um data warehouse?**. Disponível em <https://aws.amazon.com/pt/data-warehouse/>, Acesso em 27/072019.

AUSLAND. **Conceito de Business Intelligence.** 2015. Disponível em <http://ausland.com.br/blog/conceito-de-business-intelligence/>, Acesso em 27/02/2018.

BRAGHITTONI, R. **Business Intelligence: implementar do jeito certo e a custo zero**. Edição: Adriano Almeida, Vivian Matsu. Casa do Código, 2017.

DEVMEDIA. **Introdução a Views**. 2015. Disponível em <https://www.devmedia.com.br/introducao-a-views/1614/>, Acesso em 27/07/2019.

DEVMEDIA. **Como começar com Spring?**. 2015. Disponível em <https://www.devmedia.com.br/exemplo/como-comecar-com-spring/73>, Acesso em 15/03/2019.

DOCKER. **Documentação Docker**. 2019. Disponível em <https://www.docker.com/>, Acesso em 15/03/2019.

ELIAS, D. **Entendo a arquitetura do Business Intelligence (BI)**. 2015. Disponível em: <https://canaltech.com.br/business-intelligence/Entendendo-a-arquitetura-do-Business-Intelligence-BI/>, Acesso em 28/02/2019.

FIA. **Business Intelligence**. 2018. Disponível em <https://fia.com.br/blog/business-intelligence/>, Acesso em 08/10/2018.

GOIS, I. **9 problemas de quem não investe em Businnes Intelligence**. 2017. Disponível em <https://www.linkedin.com/pulse/todas-empresas-t%C3%Aam-problemas-algumas-solu%C3%A7%C3%B5es-igor-gois/>, Acesso em 27/10/2018.

LEWIS, J. **Microsserviços em poucas palavras**. 2015. Disponível em <https://www.thoughtworks.com/pt/insights/blog/microservices-nutshell>, Acesso em 27/02/2019.

OPUS. **Micro Serviços: Qual a diferença para a Arquitetura Monolítica?**. 2018. Disponível em <https://www.opus-software.com.br/micro-servicos-arquietura-monolitica/>, Acesso em 08/10/2018.

PITON, R. **O que é BI – Business Inteligence?**. 2017. Disponível em <https://rafaelpiton.com.br/blog/o-que-e-bi-business-intelligence/>, Acesso em 27/02/2019.

REDHAT. **O que são microsserviços?**. 2018. Disponível em <https://www.redhat.com/pt-br/topics/microservices/what-are-microservices>, Acesso em 27/02/2019

SITEWARE. **O que é BI Business Intelligence**. 2018. Disponível em < https://www.siteware.com.br/gestao-estrategica/o-que-e-bi-business-intelligence//>, Acesso em 02/10/2018.

SPRING. **Documentação Spring**. 2019. Disponível em <http://www.spring.io>, Acesso em 15/03/2019.