UNIVERSIDADE ESTADUAL DE GOIÁS

CÂMPUS ANÁPOLIS DE CIÊNCIAS EXATAS E TECNOLÓGICAS HENRIQUE SANTILLO

BACHARELADO EM SISTEMAS DE INFORMAÇÃO

LEONARDO RIBEIRO

Graphql: Uma Alternativa Para APIs REST E SOAP

Anápolis  
Março, 2019

UNIVERSIDADE ESTADUAL DE GOIÁS

CÂMPUS ANÁPOLIS DE CIÊNCIAS EXATAS E TECNOLÓGICAS HENRIQUE SANTILLO

BACHARELADO EM SISTEMAS DE INFORMAÇÃO

LEONARDO RIBEIRO

Graphql: Uma Alternativa Para Apis REST E SOAP

Projeto de TC apresentado ao Departamento de Sistemas de Informação Câmpus Anápolis de Ciências Exatas e Tecnológicas Henrique Santillo da Universidade Estadual de Goiás, para obtenção da nota da primeira VA.

Orientador: Prof. Ms. Ronaldo Delfiaco

Anápolis

Março, 2019

Local reservado para a folha de aprovação (Ata de Defesa)

[na versão final de TC, colocar aqui a Ata de Defesa Escaneada, que será entregue pela banca, com boa qualidade de resolução]

FICHA CATALOGRÁFICA

|  |
| --- |
| RIBEIRO, Leonardo.  Graphql: Uma Alternativa Para APIs REST E SOAP / Orientador: Ms. Ronaldo Delfiaco  Anápolis,  Março, 2019  25 p.  Trabalho de Curso (Graduação, Bacharelado em Sistemas de Informação) -- Universidade Estadual de Goiás, Câmpus Anápolis de Ciências Exatas e Tecnológicas Henrique Santillo, Departamento de Sistemas de Informação.  1. GraphQL 2. API 3. REST 4. SOAP |

REFERÊNCIA BIBLIOGRÁFICA

RIBEIRO, Leonardo. **Graphql: Uma Alternativa Para APIs REST E SOAP** Anápolis, 2019. 26 p. Monografia – Curso de Sistemas de Informação, CCET, Universidade Estadual de Goiás.

CESSÃO DE DIREITOS

É concedida à Universidade Estadual de Goiás a permissão para disponibilizar esse documento por meio eletrônico ou reproduzir cópias, emprestar ou vender tais cópias para propósitos acadêmicos e científicos, conforme termo de autorização assinado pelo autor e arquivado na Biblioteca do Câmpus. O autor reserva outros direitos de publicação e nenhuma parte deste trabalho pode ser reproduzida sem a autorização por escrito do autor.

Anápolis

Março, 2019

Dedico este trabalho a Deus, por sempre estar do meu lado me ajudando a enfrentar os momentos difíceis que passei em minha vida, aos meus pais por sempre estarem me incentivando, aos meus professores, por sempre serem compreensivos em momentos que tive dificuldade, a minha namorada que sempre me motivou a dar o melhor de mim.

**AGRADECIMENTOS**

Fazer aqui os agradecimentos que o aluno achar adequando.

Observar a linguagem formal e culta.

# RESUMO

Palavras-chave: GraphqQL, APIs, REST, SOAP.

Durante muito tempo desenvolvedores de software têm utilizado tecnologias como REST e SOAP no desenvolvimento de suas APIs. No presente trabalho, será apresentada uma nova tecnologia apresentada e utilizada pela empresa Facebook para facilitar a construção de APIs, essa tecnologia é chamada de GraphQL.

O principal objetivo do trabalho é apresentar de maneira sucinta como funciona essa tecnologia e quais suas vantagens comparadas com os já existentes formatos de construção de APIs, como REST e SOAP.

A metodologia utilizada para detalhar o funcionamento dessa nova tecnologia será através de um estudo de caso, onde será construído um pequeno sistema utilizando APIs REST, em seguida a mesma aplicação seria construída utilizando GraphQL, apontando os diferentes pontos entre ambas.

No final desse trabalho será apresentado o resultado comparativo dentre as tecnologias estudadas referente a performance de desenvolvimento, pontos positivos e pontos negativos do ponto de vista do leitor, quantidade de linhas escritas de código para cada metodologia, dificuldade de aprendizado e o que se pode esperar no futuro dessas tecnologias, também no ponto de vista do autor.

A partir dos resultados citados, espera-se que seja o suficiente para ajudar profissionais que trabalham no ramo de desenvolvimento de software a tomarem melhores decisões quanto a definição e construção de APIs, levando uma maior produtividade aos desenvolvedores.

# ABSTRACT

For a long time software developers have been using technologies like REST and SOAP to develop their APIs. In the current work, it’ll be presented a new technology presented and used by Facebook to make it easier to build APIs, this technology is called GraphQL.

The main goal of this document is to present in a succinct way how this technology works and what’s its advantages compared with the existing ways to build APIs, like REST and SOAP.

The methodology used to detail the operation of this new technology will be through a case study, where will be built a small system using REST APIs following by the construction of the same application built using GraphQL pointing the different points on both applications.

At the end of this work will be presented the comparative results between the studied technologies about development performance, positive and negative details in the author’s point of view, how many code lines written for each methodology, learning difficulty and what can be waited for the future of these technologies.

From the results quoted, it’s expected that it could be enough to help professionals who works in software development area to take better decisions regarding the construction and definition of their APIs, taking more productivity to them.

Keywords: GraphqQL, APIs, REST, SOAP.

# LISTA DE ILUSTRAÇÕES

Ilustração 1 – Nome da ilustração p

Ilustração 2 – Nome da ilustração p

[Colocar, se houver. Onde p é o número da página]

# LISTA DE GRÁFICOS

Gráfico 1 – Nome do gráfico p

[Colocar, se houver. Onde p é o número da página]

# LISTA DE QUADROS

Quadro 1 – Nome do quadro p

[Colocar a lista de quadros, numerado progressivamente, se houver. Onde p é o número da página]

[OBSERVAÇÃO: Observar a diferença entre quadro e tabela, conforme preconiza a ABNT. O quadro tem as quatro bordas fechadas. As tabelas, conforme a ABNT indica, seguem o padrão do IBGE para normas estatísticas e apresentação de dados estatísticos em tabelas (Norma de Apresentação Tabular do IBGE) sendo, uma das características, que a margem esquerda e direita não são fechadas por linhas. Para mais esclarecimentos, consultar as normas publicadas no Ambiente Virtual do TC, dentre elas as da ABNT e a Norma de Apresentação Tabular do IBGE.]

# LISTA DE TABELAS

Tabela 1 – Nome da tabela p

[Colocar a lista de tabelas, se houver.]

[OBSERVAÇÃO: Observar a diferença entre quadro e tabela, conforme preconiza a ABNT. O quadro tem as quatro bordas fechadas. As tabelas, conforme a ABNT indica, seguem o padrão do IBGE para normas estatísticas e apresentação de dados estatísticos em tabelas (Norma de Apresentação Tabular do IBGE) sendo, uma das características, que a margem esquerda e direita não são fechadas por linhas. Para mais esclarecimentos, consultar as normas publicadas no Ambiente Virtual do TC, dentre elas as da ABNT e a Norma de Apresentação Tabular do IBGE.]

# LISTA DE ABREVIATURAS E SIGLAS

**Siglas Descrição**

API *Application Programming Interface* - Interface de programação da aplicação

REST *Representational State Transfer* - Transferência de estado representacional

SOAP *Simple Object Access Protocol* - Protocolo de acesso de objeto simples

JSON *JavaScript Object Notation* – Notação de objetos javascript

SUMÁRIO

[RESUMO 8](#_Toc7276577)

[ABSTRACT 9](#_Toc7276578)

[LISTA DE ILUSTRAÇÕES 10](#_Toc7276579)

[LISTA DE GRÁFICOS 11](#_Toc7276580)

[LISTA DE QUADROS 12](#_Toc7276581)

[LISTA DE TABELAS 13](#_Toc7276582)

[LISTA DE ABREVIATURAS E SIGLAS 14](#_Toc7276583)

[INTRODUÇÃO 16](#_Toc7276584)

[1. World Wide Web 17](#_Toc7276585)

[1.1 Uma breve história 17](#_Toc7276586)

[1.2 Desenvolvimento de Aplicações para web 18](#_Toc7276587)

[2. REST, SOAP e JSON 19](#_Toc7276588)

[2.1 Subtítulo 1 19](#_Toc7276589)

[2.2 Subtítulo n 19](#_Toc7276590)

[3. O que é GraphQL 20](#_Toc7276591)

[n.1 Subtítulo 1 20](#_Toc7276592)

[n.2 Subtítulo n 20](#_Toc7276593)

[4. Comparação de APIs tradicionais e GraphQL 20](#_Toc7276594)

[5. Estudo de caso 21](#_Toc7276595)

[6. Resultados de estudo de caso 21](#_Toc7276596)

[CONCLUSÃO / RECOMENDAÇÕES 21](#_Toc7276597)

[REFERÊNCIAS 22](#_Toc7276598)

[GLOSSÁRIO 23](#_Toc7276599)

[APÊNDICES 24](#_Toc7276600)

[Apêndice A – Cronograma previsto e realizado 24](#_Toc7276601)

[Apêndice B - Banner [ou Artigo] apresentado na Semana de Comunicação Científica 25](#_Toc7276602)

[Apêndice N – Título do Apêndice 26](#_Toc7276603)

[ANEXOS 27](#_Toc7276604)

[Anexo A 27](#_Toc7276605)

# INTRODUÇÃO

A evolução da internet tem se manifestado de muitas formas: as características de tráfego, a interconexão entre topologias, o relacionamento entre negócios e componentes autônomos, a facilidade do acesso a informação, entre outros. Até hoje essa evolução vem acontecendo de maneiras inesperadas em muitos aspectos. Começando com a forma que as aplicações web eram moldadas nos anos 2000 e como são construídas hoje em 2019. É importante entender o por quê (e como) dessa evolução estar acontecendo, e como isso pode afetar drasticamente como os usuários navegam, a interação das empresas que lidam com aplicações web e com os desenvolvedores dessas aplicações. (MA; LUI; MISRA, 2015)

Graças ao trabalho árduo de desenvolvedores, a internet foi criada, e através dela conseguimos facilitar o compartilhamento de informações. Porém, esse era o objetivo inicial de quando a internet foi criada. Hoje, além de compartilhar informações entre alguns departamentos dentro das empresas, também usamos a internet como forma de entretenimento, criar novos negócios, realizar compras e vendas de produtos e/ou serviços, ou até mesmo gerir a internet em si (rede local, por exemplo). // adicionar referência

A maneira mais comum que utilizamos para fazer isso é através do desenvolvimento de aplicações web, o qual o usuário é capaz de interagir através de entrada de dados (clicks, textos em formulários, arquivos) e esperam receber uma saída de dados ou um feedback dessa aplicação. Esse feedback pode ser informações já inseridas anteriormente, download de algum arquivo, ou qualquer informações que supra a necessidade do usuário. (MULLOY, 2012)

Para desenvolver essas aplicações, utilizamos o que é chamado de *webservice* para gerir essas requisições que o usuário faz através de uma aplicação web. Esse *web servisse* nada mais é do que um serviço com um ponto de entrada (uma *url*) onde são mandadas informações pelo usuários. Essas informações são processadas e retornam um *feedback* sobre aquela requisição. (BOREK *et al.*, 2015)

A maneira com a qual esses *webservices* foram desenvolvidos também evoluíram com o tempo. Sua principal mudança sempre teve como foco a melhoria da produtividade dos desenvolvedores para construção dos *webservices*, performance de processamento dos dados enviados nas requisições, automação na produção de documentação e processamento mais amigável tanto para máquinas quanto para humanos que interagem com o *webservice*. (BOREK *et al.*, 2015)

Até o presente momento, essas aplicações eram desenvolvidas utilizando tecnologias como *REST* e *SOAP*. O problema de utilizar essas tecnologias é que às vezes tínhamos um nível abstração baixo e sem controle na resposta das requisições. Imagine, por exemplo, que um usuário fizesse uma requisição em uma *url* e essa retornasse uma quantidade imensa de dados, dos quais não seriam usados nem a metade. Além disso, teríamos uma resposta no formato *JSON*, que deveria ser interpretada pelo cliente (*frontend* da aplicação, parte que o usuário vê), para depois então descartar a maioria desses dados. Por causa dessa requisição, foram consumidos recursos de internet, cliente e servidor. (BRITO; MOMBACH; VALENTE, 2019)

Neste trabalho, temos como objetivo apresentar uma das tecnologias desenvolvidas para construção de *webservices*, proposta e desenvolvida recentemente pelo Facebook em 2016, chamada GraphQL. Essa tecnologia é uma linguagem de consulta proposta para implementar APIs (*Application Programming Interface*) baseadas em aplicações web. Essa tecnologia é capaz de solicitar, em uma única requisição, respostas complexas apenas com os dados necessários para o cliente, evitando consumo desnecessário de recursos, consequentemente melhorando a performance da requisição. Além disso, essa tecnologia trás uma abordagem diferente na construção de *APIs*, melhorando também a produtividade do desenvolvedor, que será detalhada mais adiante. (BRITO; MOMBACH; VALENTE, 2019)

# World Wide Web

Escrever bem pode ser um dom para poetas e literatos, mas a maioria de nós está apta para escrever um simples artigo, um resumo, uma redação tosca das próprias ideias, sem mexer com literatura nem com grandes emoções humanas.

## 1.1 Uma breve história

Escrever bem pode ser um dom para poetas e literatos, mas a maioria de nós está apta para escrever um simples artigo, um resumo, uma redação tosca das próprias ideias, sem mexer com literatura nem com grandes emoções humanas

No caso de citação direta de mais de 3 linhas:

Deverá ser feito o alinhamento a 4 cm da margem direita, usando fonte Times New Roman, tamanho 11, espaço entrelinhas 1 (um). Nesse caso, observe que não se faz necessário colocar as aspas. Ao final deverá ser feito a indicação da obra, conforme padrão das normas da ABNT. (LIMA, 2006, p. 132).

## 1.2 Desenvolvimento de Aplicações para web

Quando do no documento tiver ilustração, gráfico, quadro, etc. esse deve ser numerado com o título na parte superior e indicação da fonte na parte inferior, como no exemplo:

Quadro 1 – Título do quadro

| Coluna | Coluna | Coluna | Coluna |
| --- | --- | --- | --- |
| Dados | Dados | Dados | Dados |
|
|
| Dados | Dados | Dados | Dados |
|
|

Fonte: Elaborado pelo autor

# 2. REST, SOAP e JSON

Escrever bem pode ser um dom para poetas e literatos, mas a maioria de nós está apta para escrever um simples artigo, um resumo, uma redação tosca das próprias ideias, sem mexer com literatura nem com grandes emoções humanas.

## 2.1 Subtítulo 1

Escrever bem pode ser um dom para poetas e literatos, mas a maioria de nós está apta para escrever um simples artigo, um resumo, uma redação tosca das próprias ideias, sem mexer com literatura nem com grandes emoções humanas.

## 2.2 Subtítulo n

Escrever bem pode ser um dom para poetas e literatos, mas a maioria de nós está apta para escrever um simples artigo, um resumo, uma redação tosca das próprias ideias, sem mexer com literatura nem com grandes emoções humanas.

# 3. O que é GraphQL

Escrever bem pode ser um dom para poetas e literatos, mas a maioria de nós está apta para escrever um simples artigo, um resumo, uma redação tosca das próprias ideias, sem mexer com literatura nem com grandes emoções humanas.

## n.1 Subtítulo 1

Escrever bem pode ser um dom para poetas e literatos, mas a maioria de nós está apta para escrever um simples artigo, um resumo, uma redação tosca das próprias ideias, sem mexer com literatura nem com grandes emoções humanas.

## n.2 Subtítulo n

Escrever bem pode ser um dom para poetas e literatos, mas a maioria de nós está apta para escrever um simples artigo, um resumo, uma redação tosca das próprias ideias, sem mexer com literatura nem com grandes emoções humanas.

# 4. Comparação de APIs tradicionais e GraphQL

Escrever bem pode ser um dom para poetas e literatos, mas a maioria de nós está apta para escrever um simples artigo, um resumo, uma redação tosca das próprias ideias, sem mexer com literatura nem com grandes emoções humanas.

# 5. Estudo de caso

Escrever bem pode ser um dom para poetas e literatos, mas a maioria de nós está apta para escrever um simples artigo, um resumo, uma redação tosca das próprias ideias, sem mexer com literatura nem com grandes emoções humanas.

# 6. Resultados de estudo de caso

Escrever bem pode ser um dom para poetas e literatos, mas a maioria de nós está apta para escrever um simples artigo, um resumo, uma redação tosca das próprias ideias, sem mexer com literatura nem com grandes emoções humanas.

# CONCLUSÃO / RECOMENDAÇÕES

Escrever bem pode ser um dom para poetas e literatos, mas a maioria de nós está apta para escrever um simples artigo, um resumo, uma redação tosca das próprias ideias, sem mexer com literatura nem com grandes emoções humanas.

# REFERÊNCIAS

SOBRENOME, Nome do autor. **Título da obra**. Disponível em: <http://www.enderecoeletronico.com.br>. Acesso em: data do acesso.

SOBRENOME, Nome do autor. **Título da obra**. Cidade: Editora, ano.

SOBRENOME, Nome do autor. **Título da obra**. Disponível em: <http://www.enderecoeletronico.com.br>. Acesso em: data do acesso.

SOBRENOME, Nome do autor. **Título da obra**. Cidade: Editora, ano.

[Deverá ser conforme as normas da ABNT e ordenado alfabeticamente].

# GLOSSÁRIO

# APÊNDICES

## Apêndice A – Cronograma previsto e realizado

[Opcional no TC final, conforme determinação do orientador / banca]

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| N.º | ATIVIDADE | Cronograma de execução do Trabalho de Curso. | | | | | | | | | | |
| FEV | MAR | ABR | MAI | JUN | JUL | AGO | SET | OUT | NOV | DEZ |
|  |  |  |  |  |  |  |  |  |  |  |  |  |
| 01 | Atividade | X |  |  |  |  |  |  |  |  |  |  |
| 02 | Atividade | X | X |  |  |  |  |  |  |  |  |  |
| 03 | Atividade |  | X | X |  |  |  |  |  |  |  |  |
| 04 | Atividade |  | X | X | X |  |  |  |  |  |  |  |
| 05 | Atividade |  |  |  | X | X |  |  |  |  |  |  |
| 06 | Atividade |  |  |  |  | X |  |  |  |  |  |  |
| 07 | Atividade |  |  |  |  |  |  | X | X |  |  |  |
| 08 | Atividade |  |  |  |  |  |  | X | X |  |  |  |
| 09 | Atividade |  |  |  |  |  |  |  | X | X |  |  |
| 10 | Atividade |  |  |  |  |  |  |  |  | X |  |  |
| 11 | Atividade |  |  |  |  |  |  |  |  | X |  |  |
| 12 | Atividade |  |  |  |  |  |  |  |  | X |  |  |
| 13 | Atividade |  |  |  |  |  |  |  |  | X | X |  |
| 14 | Atividade |  |  |  |  |  |  |  |  |  | X |  |
| 15 | Atividade |  |  |  |  |  |  |  |  |  | X | X |

## Apêndice B - Banner [ou Artigo] apresentado na Semana de Comunicação Científica

# ANEXOS

## Anexo A