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

# 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 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

WWW *World Wide Web* - Rede mundial de computadores

SUMÁRIO

[RESUMO 5](#_Toc7286698)

[ABSTRACT 6](#_Toc7286699)

[LISTA DE ABREVIATURAS E SIGLAS 7](#_Toc7286700)

[INTRODUÇÃO 9](#_Toc7286701)

[1. World Wide Web 11](#_Toc7286702)

[1.1 Uma breve história 11](#_Toc7286703)

[REFERÊNCIAS 13](#_Toc7286704)

[APÊNDICES 14](#_Toc7286705)

[Apêndice A – Cronograma previsto e realizado 14](#_Toc7286706)

# 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)

Para validar a hipótese apresentada pela tecnologia desenvolvida, o presente trabalho abordará um estudo de caso em que o autor desenvolverá uma aplicação utilizando tecnologias de desenvolvimento de *webservice* atuais, sendo a mesma uma API REST e em seguida desenvolverá a mesma aplicação utilizando a tecnologia proposta, GraphQL. Durante o desenvolvimento das aplicações, será avaliado o tempo de duração do desenvolvimento, quantidade de linhas de código escrito e performance nas requisições.

# World Wide Web

Para entendermos melhor o funcionamento das aplicações web, precisamos antes entender o que é exatamente web e como surgiu o famoso “*www*”. Para isso, a seguir, vamos abordar uma breve história sobre o surgimento da internet, sua evolução até o dia de hoje e depois, também abordaremos um pouco sobre como são criados os web sites e aplicações web que acessamos diariamente.

## Uma breve história

O primeiro passo para o que hoje conhecemos como internet começou no dia 29 de Agosto de 1949, quando a União Soviética detonou sua primeira arma nuclear, um evento marcante no pós-guerra entre seu confronto com os Estados Unidos, na segunda guerra mundial. Como os aviões eram a única maneira de levar uma bomba nuclear naquela época, a defesa aérea se tornou uma preocupação e prioridade imediata para os Estados Unidos. Em 1948, a força aérea já havia estudado a necessidade de um sistema de redes para notificação antecipada para a defesa aérea. Para isso, o professor do *Massachussetts Institute of Technology* (Instituto Tecnológico de Massachussetts), George Valley foi nomeado para recomendar um sistema melhorado que resolvesse as deficiências da rede de radares da época, o qual deveria ser montado as pressas. Os dados gerados pelos radares eram escritos a mão utilizando lápis de graxa em sentido inverso em placas transparentes. (LUKASIK, 2011)

Para automatizar esse processo, a ADSEC (*Air Defense System Engineering Committee* - Comitê de Engenharia do Sistema de Defesa Aérea) recomendou a automação desse processo através da digitalização dessas informações. Essas informações seriam enviadas através das redes de telefones, e esses dados seriam analisados pelos novos computadores digitais sendo desenvolvidos. (LUKASIK, 2011)

Quando a Coréia do Norte invadiu a Coréia do Sul em 25 de Junho de 1950 e quando a China interveio em 26 de Novembro do mesmo ano, trazendo um espectro de guerra nuclear com um mundo comunista ainda mais próximo, a urgência da força aérea americana para um sistema de radares mais efetivo aumentou. Vários projetos foram sugeridos, porém nenhum tão efetivo até 1951, quando foi criado o Projeto Lincoln, onde, para validar as hipóteses apresentadas, um “bombardeiro” foi interceptado em tempo real por um “lutador” utilizando dados transferidos dos radares por linha telefônica. (LUKASIK, 2011)

Em 1958, foi estabelecida a ARPA (*Advanced Research Projects Agency* – Agência de Projetos de Pesquisa Avançados) pelo governo dos Estados Unidos como resposta a crise de segurança do pós-guerra da segunda guerra mundial. A ARPA era responsável por desenvolver pesquisas relacionadas a comunicação, navegação, defesa de mísseis balísticos, sistemas de rastreamentos por satélite, lançamento de mísseis, dentre outras disciplinas acadêmicas que oferecessem suporte ao meio militar do país. (LUKASIK, 2011)

Em 1960 deu seus primeiros passos, quando Joseph Carl Robnett Licklider – um cientista da computação, juntamente com a companhia Bolt, Berenek and Newman (BBN) – formularem ideias para a construção de uma rede global em vários memorandos, descrevendo uma “Rede Intergaláctica de Computadores”. (PALERMO; COX, 2014)

Devido às ideias, Licklider se tornou diretor da ARPA. Sua ambição era que todos os computadores existentes pudessem ter alguma forma de se conectar por todo o globo, e todos próximos a um computador poderiam compartilhar informações. (PALERMO; COX, 2014)

Em 1969, a primeira mensagem através de uma rede de computadores foi enviado utilizando o protocolo IMP (Interface Message Processor) por Leonard Kleinrock – um dos contribuidores para o projeto de desenvolvimento da Arpanet – na UCLA (*University of California, Los Angeles*), onde o conteúdo era apenas “LO”. A mensagem foi recebida por Elizabeth Feinler no SRI (*Standford Research Institute*), em seguida a rede caiu, porém essa foi uma mensagem simples, que entrou para a história. (LUKASIK, 2011)

Depois dessa primeira mensagem, muitas melhorias foram feitas até a rede ficar mais estável. Em 1983, a Arpanet foi dividia em duas partes, uma rede de pesquisa que continuou sendo chamada de Arpanet, e a Milnet utilizada para atividades militares. Em 1989 a Arpanet foi descontinuada, porém não foi evitado sua metamorfose no que conhecemos hoje como “Internet”. A intenção da Arpanet era apenas demonstrar sua utilidade para comutação de pacotes no meio militar, o qual foi um sucesso. Como consequência inesperada, a Arpanet e sua sucessora, a Internet, demonstraram ainda mais a utilidade geral do trabalho em rede para "comando e controle" muito além das necessidades militares. (LUKASIK, 2011)

# REFERÊNCIAS

BOREK, M. *et al.* *Correct Software in Web Applications and Web Services*. [S.l: s.n.], 2015. Disponível em: <http://link.springer.com/10.1007/978-3-319-17112-8>.

BRITO, G.; MOMBACH, T.; VALENTE, M. T. Migrating to GraphQL: A Practical Assessment. *SANER 2019 - 26th edition of the IEEE International Conference on Software Analysis, Evolution and Reengineering*, 2019.

LUKASIK, S. J. “Why the Arpanet Was Built” (vol 33, pg 3, 2011). *Ieee Annals of the History of Computing*, v. 33, n. 4, p. 3, 2011.

MA, R. T. B.; LUI, J. C. S.; MISRA, V. Evolution of the internet economic ecosystem. *IEEE/ACM Transactions on Networking*, v. 23, n. 1, p. 85–98, 2015.

MULLOY, B. Web API Design. 2012.

PALERMO, E.; COX, L. *Who Invented the Internet? | ARPANET*. Disponível em: <https://www.livescience.com/42604-who-invented-the-internet.html>. Acesso em: 27 abr. 2019.

# APÊNDICES

## Apêndice A – Cronograma previsto e realizado

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Tarefa** | **início planejado** | **início real** | **entrega planejada** | **entrega real** | **Concluído** |
| 1ª reunião do TC – Coordenador de TC e alunos (20h50 às 22h30, sala 112) | 26/2/19 | 26/2/19 | 26/2/19 | 26/2/19 | sim |
| Materiais para conteúdo | 1/3/19 | 1/3/19 | 1/3/19 | 1/3/19 | sim |
| Criar repositórios para versionamento de aplicação de demonstração e para arquivo de TCC | 1/3/19 | 1/3/19 | 1/3/19 | 1/3/19 | sim |
| Cronograma | 1/3/19 | 1/3/19 | 2/3/19 | 8/3/19 | sim |
| Resumo / Abstract | 4/3/19 | 8/3/19 | 8/3/19 | 8/3/19 | sim |
| Introdução | 3/3/19 | 8/3/19 | 8/3/19 | 8/3/19 | sim |
| Atualizar Cronograma | 29/3/19 | 29/3/19 | 29/3/19 | 29/3/19 | sim |
| world wide web | 9/3/19 | 29/3/19 | 15/3/19 | 27/4/19 | sim |
| REST, SOAP e JSON | 16/3/19 |  | 22/3/19 |  | não |
| Data limite e disponibilização do termo de aceite no Moodle pelo aluno | 21/3/19 | 21/3/19 | 21/3/19 | 21/3/19 | sim |
| O que é GraphQL? | 23/3/19 |  | 12/4/19 |  | não |
| 2ª reunião do TC – Coordenador Adjunto de TC e alunos (20h50 às 22h30, sala 112) | 2/4/19 |  | 2/4/19 |  | não |
| GraphQL e APIs tradicionais | 13/4/19 |  | 3/5/19 |  | não |
| Data final para entrega do pré-projeto de TC e postagem no Moodle pelo aluno (TC1.N1) | 27/4/19 |  | 27/4/19 |  | não |
| Data final para postagem das avaliações do pré-projeto de TC (TC1.N1) pelos orientadores | 3/5/19 |  | 3/5/19 |  | não |
| 3ª reunião do TC - Coordenação Adjunta de TC e (20h50 às 22h30, sala 112) | 21/5/19 |  | 21/5/19 |  | não |
| Poster | 1/6/19 |  | 7/6/19 |  | não |
| Data final para entrega e postagem, pelo aluno, do projeto e conteúdos (TC1.N2) | 3/6/19 |  | 3/6/19 |  | não |
| Publicação das agendas das bancas de avaliação parcial de TC (TC1.N2) pela coordenação adjunta | 5/6/19 |  | 5/6/19 |  | não |
| 10 a 15 - Realização das bancas de avaliação parcial de TC (TC1.N2) | 10/6/19 |  | 10/6/19 |  | não |
| 25 - Data final para entrega e postagem no Moodle, pelos orientadores, das avaliações (TC1.N2) | 25/6/19 |  | 25/6/19 |  | não |
| 27 - Data limite de fechamento e entrega dos diários de TC | 27/6/19 |  | 27/6/19 |  | não |
| Aplicação Prática utilizando GraphQL | 29/6/19 |  | 16/8/19 |  | não |
| Conclusão | 17/8/19 |  | 2/9/19 |  | não |