



CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS  
DEPARTAMENTO DE COMPUTAÇÃO  
CURSO DE ENGENHARIA DE COMPUTAÇÃO

# **OTIMIZANDO DESEMPENHO DE FRONT-END EM WEBSITES PARA HTTP2**

**PEDRO COLEN CARDOSO**

Orientador: Flávio Coutinho  
Centro Federal de Educação Tecnológica de Minas Gerais – CEFET-MG

BELO HORIZONTE  
MARÇO DE 2015

**PEDRO COLEN CARDOSO**

**OTIMIZANDO DESEMPENHO DE FRONT-END EM  
WEBSITES PARA HTTP2**

Proposta de Trabalho de Conclusão de Curso apresentado  
ao Curso de Engenharia da Computação do Centro  
Federal de Educação Tecnológica de Minas Gerais

Orientador: Flávio Coutinho  
Centro Federal de Educação Tecnológica  
de Minas Gerais – CEFET-MG

CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS  
DEPARTAMENTO DE COMPUTAÇÃO  
CURSO DE ENGENHARIA DE COMPUTAÇÃO  
BELO HORIZONTE  
MARÇO DE 2015

## Lista de Quadros

Quadro 1 – Cronograma de atividades. . . . .	9
--	---

## **Lista de Abreviaturas e Siglas**

CERN	Organização Europeia de Pesquisas Nucleares
W3	World Wide Web
HTML	HyperText Markup Language
HTTP	Hypertext Transfer Protocol

# Sumário

<b>1 – Introdução</b>	<b>1</b>
1.1 Otimização de Desempenho de Websites	1
1.2 Espaço para Novas Técnicas	3
<b>2 – Descrição do problema</b>	<b>4</b>
<b>3 – Objetivos</b>	<b>5</b>
3.1 Objetivo Geral	5
3.2 Objetivos Específicos	5
<b>4 – Resultados Esperados</b>	<b>6</b>
<b>5 – Metodologia</b>	<b>7</b>
<b>6 – Infraestrutura necessária</b>	<b>8</b>
<b>7 – Cronograma</b>	<b>9</b>
<b>Referências</b>	<b>10</b>

# 1 Introdução

A Internet como é conhecida nos dias de hoje foi proposta em 1989 pelo cientista britânico Tim Berners-Lee pesquisador da Organização Europeia de Pesquisas Nucleares, CERN. De acordo com a CERN a ideia de Lee era ajudar seus colegas a compartilharem informações científicas de maneira mais rápida (CERN, 2015), possibilitando o maior avanço das pesquisas realizadas na Europa e ao redor de todo o mundo. Para isso Lee queria inventar um modelo simples para o compartilhamento de dados via rede de computadores e então criou o projeto da *World Wide Web* (W3). Como descrito por Lee em seu primeiro *website* o W3 era “uma iniciativa de recuperação de informação em hipermídia em grande área, com o objetivo de dar acesso universal para um grande universo de documentos” (BERNERS-LEE, 1990). No final de 1990, Lee já tinha terminado de desenvolver todas as ferramentas necessárias para o funcionamento de seu modelo: uma linguagem de marcação de hipertextos (HTML), um navegador *web* (o WorldWideWeb) e um protocolo para a transferência de hipertextos (HTTP). O primeiro *site* que Lee colocou no ar ainda pode ser acessado pelo seu endereço original, <http://info.cern.ch/>, e contém informações sobre o próprio W3.

Em 1993, vendo as vantagens que poderiam trazer para todos, a CERN decidiu colocar a World Wide Web em domínio público. Então, em 1994, Tim Berners-Lee fundou o *World Wide Web Consortium* (W3C), com o objetivo de formalizar especificações e regras para o uso do seu modelo. O W3C se tornou o responsável por garantir que a Web funcionasse para todos e que evoluísse de uma maneira consciente.

Muita coisa mudou desde que Tim propôs e desenvolveu a primeira versão da World Wide Web. O modelo de Tim se tornou o mais utilizado pela rede mundial de computadores, a Internet, e hoje em dia, de acordo com o *site* Internet World Status mais de 3 bilhões de pessoas utilizam das ferramentas desenvolvidas por ele para acessar informações pessoais, fazer pesquisas, se comunicar com amigos, compartilhar trabalhos, fazer compras, etc. (GROUP, 2014)

## 1.1 Otimização de Desempenho de Websites

Atualmente, *websites* são compostos por páginas interativas, com vídeos, fotos, animações, cores, links e outros elementos dinâmicos que tiram toda a monotonia de uma folha de papel apenas com textos. Mas o primeiro *website* desenvolvido por Lee em 1990 era bem mais simples do que isso e continha apenas textos estáticos e links internos.

Com o avanço da Web surgiram novas maneiras de interagir com ela. A hu-

manidade está na era das imagens e dos vídeos, páginas não podem ser estáticas se não os usuários não ficam o tempo necessário nela, é preciso criar *layouts* agradáveis e interessantes além de interatividade entre *websites* e usuários. Esses fatores fizeram com que os sites que antigamente possuíam alguns poucos *bytes* de informação passassem a ter muitos *megabytes*. Além disso o número de acessos simultâneos aumentou e os usuários ficaram mais impacientes, exigindo que as páginas da Web respondessem cada vez mais rápido. Então criou-se um paradoxo: os sites ficaram mais pesados e difíceis de serem carregados, os acessos ficaram mais frequentes sobrecarregando mais os servidores, mas os usuários passarão a querer respostas mais rápidas e eficientes. Sozinhas, nem mesmo as melhorias na velocidade de conexão com a Internet seriam o suficiente para solucionar esse problema. Algo a mais precisava ser feito.

Por muitos anos acreditou-se que para melhorar o desempenho de um *website* era suficiente melhorar o desempenho do que está por trás dele, o *back-end*. O *back-end* é a parte responsável por gerenciar as operações de um *website* e controlar tudo o que acontece na parte vista pelo usuário, o *front-end*.

No início dos anos 2000, começaram a ser realizados novos estudos a procura de técnicas para a entrega mais rápida de conteúdo pela Web. A otimização da performance de *websites* se tornou um fator crítico para o sucesso de algumas empresas que dependiam da Web para sobreviver. Os pesquisadores então perceberam que otimizar *back-end* não era o suficiente, pois apesar de ser nele onde ocorrem que as operações mais pesadas computacionalmente as melhorias de desempenho eram limitadas. Mesmo assim muito tempo e dinheiro foi gasto procurando maneiras de otimizar servidores e sistemas de gestão de conteúdo, e pouca atenção foi dada ao *front-end*. Mas de acordo com Steve Souders ([Souders, 2007](#)) esse foi o erro dos desenvolvedores por anos. Apenas 10-20% do tempo de carregamento de uma página é gasto com operações de *back-end*, os outros 80-90% são de responsabilidade do *front-end*.

Até o ano de 2007, pouco conteúdo sobre otimização de *front-end* era disponibilizado para o grande público, até que o engenheiro de software do Yahoo!, Steve Souders, publicou o livro “High Performance Web Sites” explicando 14 técnicas utilizadas pela equipe da gigante da Internet para tornar seus sites mais rápidos ([Souders, 2007](#)). Essas técnicas eram específicas para otimização no *front-end* dos sites. A evolução da Web fez com que esse tipo de otimização se tornasse fundamentalmente importante para a velocidade da entrega do conteúdo, e Souders fez com que esse segredo deixasse de pertencer apenas às grandes empresas e passasse a ser do conhecimento de todos.

## 1.2 Espaço para Novas Técnicas

Para desenvolver seus métodos de otimização, Souders se baseou no protocolo de troca de conteúdo criado por Tim Berners-Lee em 1990, o HTTP. Esse protocolo tinha algumas características e limitações que faziam necessárias as técnicas propostas por ele.

Ao longo dos anos, o HTTP mudou muito pouco. Após a sua primeira versão oficial, que teve sua descrição aprovada em 1996, o HTTP/1.0, foi lançado apenas uma atualização do protocolo, que foi aprovado em 1999, o HTTP/1.1 ([NIELSEN, 2004](#)). O protocolo era tão robusto que de sua ideia inicial pouca coisa precisou ser mudada para o HTTP ser usado em todo o mundo.

Às vésperas do lançamento do HTTP2, desenvolvido com o intuito de melhorar o desempenho e a segurança da Internet, estima-se que muitas das técnicas propostas por Souders se tornem obsoletas. As mudanças geradas pela implantação do HTTP2 farão necessárias novas técnicas de otimização e o desenvolvimento dessas técnicas dependerá da compreensão do novo protocolo. Por isso é necessário já começar a estudar e analisar o comportamento do HTTP2 na Web.



## 2 Descrição do problema

Com a evolução da Internet os *websites* passaram a ser cada vez mais robustos e volumosos. Foram inventadas novas maneiras de inserir informações, criar interações e estilizar as páginas da web. Com o surgimento de linguagens como o CSS e o JavaScript, os *sites* ficaram mais atraentes e interessantes e, além disso, eles deixaram de ser apenas páginas estáticas para o compartilhamento de conteúdo e passaram a ser aplicações complexas com várias funcionalidades. Com esses novos sites iterativos e atraentes surgiu também a necessidade de técnicas para torná-los mais rápidos.

Ao longo dos anos técnicas para melhorar o desempenho dos *websites* foram desenvolvidas e aplicadas em muitas páginas da Web. Mas o protocolo de transferência de hipermídia, o HTTP, está sofrendo mudanças e já foi confirmada o lançamento de uma nova versão, o HTTP2. Com esse novo protocolo deverão ocorrer mudanças na maneira como são feitas as otimizações de desempenho dos *websites*.

Este trabalho tem a proposta de avaliar a necessidade e a eficácia das técnicas existentes com a chegada do HTTP2 e ainda propor novas técnicas adequadas ao novo protocolo caso isso seja necessário.

## 3 Objetivos

### 3.1 Objetivo Geral

O objetivo deste trabalho é analisar o comportamento das técnicas de otimização propostas por Steve Souders quando utilizadas no HTTP2 e, se necessário, propor novas técnicas adequadas ao novo protocolo.

### 3.2 Objetivos Específicos

1. Fazer uma análise comparativa das versões do protocolo HTTP
2. Avaliar os ganhos de desempenho das técnicas propostas por Steve Souders ao aplicá-las ao HTTP2
3. Se necessário, propor novas técnicas de otimização de desempenho de *websites* específicas para o HTTP2

## 4 Resultados Esperados

Ao final deste trabalho, será apresentada uma tabela comparativa das versões do protocolo HTTP que sirva de material de estudo e compreensão do protocolo para futuros estudos. Também será apresentada uma análise das técnicas propostas por Steve Souders aplicadas ao HTTP2 explicitando quais técnicas devem continuar sendo usadas e quais se tornarão obsoletas. E, caso se prove necessário, será propostas novas técnicas de otimização de desempenho de *websites* para o protocolo HTTP2, com especificações de como aplicá-las.

## 5 Metodologia

A primeira etapa deste trabalho será uma análise para a compreensão dos modelos de otimização de desempenho de *websites* existentes e aceitos na atualidade e a segunda etapa do trabalho será o levantamento de uma nova teoria (novas técnicas de otimização) e a validação dessa teoria.

Logo, para o desenvolvimento deste trabalho as seguintes atividades serão realizadas:

1. Estudo e compreensão dos protocolos HTTP/1.0, HTTP/1.1 e HTTP/2.0
2. Análise das técnicas de otimização de websites propostas por Steve Souders para o protocolo HTTP/1.1
3. Testes e análises de eficiência das técnicas de otimização propostas por Steve Souders em cima do protocolo HTTP/2.0
4. Proposta de novas técnicas de otimização de websites para o protocolo HTTP2
5. Testes e análises de eficiência das novas técnicas propostas para o protocolo HTTP2

## **6 Infraestrutura necessária**

Para o desenvolvimento deste trabalho será utilizado apenas um computador pessoal para a realização de testes e análises. O computador deverá ser equipado com um navegador web que possibilite a configuração do protocolo HTTP2 e ter acesso à Internet. Um requisito necessário para a confiabilidade dos testes será uma conexão à Internet com pouca oscilação e com a mesma velocidade de conexão do início ao fim do projeto, de preferência, mas não determinante, uma conexão cabeada.



## Referências

BERNERS-LEE, T. **World Wide Web**. 1990. Disponível em: <<http://info.cern.ch/hypertext/WWW/TheProject.html>>. Citado na página 1.

CERN. **The birth of the web**. 2015. Disponível em: <<http://home.web.cern.ch/topics/birth-web>>. Citado na página 1.

GROUP, M. M. **INTERNET USAGE STATISTICS The Internet Big Picture**. 2014. Disponível em: <<http://www.internetworldstats.com/stats.htm>>. Citado na página 1.

NIELSEN, J. G. H. F. **Change History for HTTP**. 2004. Disponível em: <<http://www.w3.org/Protocols/History.html>>. Citado na página 3.

SOUDERS, S. **High Performance Web Sites**. [S.l.]: O'Reilly, 2007. Citado na página 2.