Optimization and cost reduction with performance improvements on web pages

George Lucas Silva de Sousa
georgelucassilvadesousa@gmail.com
Academic department, University Center FAMETRO, Manaus-AM, Brazil

Pedro Jose Fonseca Neto
fonsecaneto.pedro@gmail.com
Academic department, University Center FAMETRO, Manaus-AM, Brazil

Bruno Pereira Gonçalves
goncalves.bruno@gmail.com
Academic department, University Center FAMETRO, Manaus-AM, Brazil

Jean Mark Lobo de Oliveira
jeanlobolive@gmail.com
Academic department, University Center FAMETRO, Manaus-AM, Brazil

David Barbosa de Alencar
david002870@hotmail.com
Research department, Institute of Technology and Education Galileo of Amazon – ITEGAM, Manaus-AM, Brazil

Abstract

Comparison between websites to improve their performance and optimization. Applied bibliographic research to understand and survey the concepts applied at work, adopting quantitative research through a questionnaire aimed at random audiences, documentary research to support important information on websites, informal magazines and descriptive research to analyze the effect of this study based on in the information presented. Five sites were adopted, one of which is a previous version and the other is a current one, with different characteristics to check its performance and optimization on the web. As explained, it was possible to verify the bottlenecks regarding the performance and optimization of the websites. With the applications of the necessary tools it was possible to improve the performance of the sites and with that the loading of the page became faster due to the compression in the loading of the images and also in the use of JavaScript, Css and Html. With the tools properly presented to improve performance and optimization, it is prominent to express the due improvements of the sites used in everyday life.

Keywords: Performance; Optimization; Tools; Web; Improvements; Loading.

1 INTRODUCTION

According to Mike Belshe (2010) in his article "More Bandwidth Doesn't Matter (much)", the indicators that hinder the performance of web pages are Band, Latency and Server. The e-commerce company Amazon has already concluded that each 100ms impact 1% of revenue (Equivalent to 30.270 million per year, 2019 revenue). Microsoft showed that 2s more latency on Bing decreased revenue by 4.3% (experiment done at Velocity Conference). And the reverse statement is also valid: Yahoo! explained that, for every 400ms of improvement in page performance, traffic increases by 9% (Own studies in 2008).

Caelum (Web Development Blog) looking for the culprits for poor web performance did an experiment on its own website, making the pages load from 2s to 4s worse, pageviews dropped 28%. The experiment exposed how web performance and optimization directly impact the website's impact, which in this case were 21% drops in the time users spend on the site, and an 18% drop in conversion rate.

With the above, the good customer experience is fundamental, and it is evident that the final performance depends more on the client side than on the server side. Aiming at these usual difficulties, mainly in the front-end and web servers, it becomes interesting to stimulate alternatives and good practices to improve the process of loading the performance of web pages, such as the use of optimization techniques to reduce traffic and so catch the attention of more users with faster sites.

2 THEORETICAL REFERENCE

2.1 Gzip

Reduces the size of named files using Lempel - Ziv (LZ77) encoding. Whenever possible, each file is replaced by one with the extension '.gz', maintaining the same ownership modes, access and modification times. If no file is specified or if a file name is-, standard input is compressed into standard output. Gzip will only attempt to compress regular files. In particular, it will ignore symbolic links. (GNU.ORG, 2016).

2.2 Minification

It occurs after the web application code has been written, but before it has been released. When a user makes a request for a web page, the minified version is sent instead of the full version and with that it is possible to get a faster response and at a lower cost from the internet band. Minification works by analyzing and rewriting parts of a website that are defined by text files (HTML code, CSS, JavaScript, etc.). It is usually performed by the web server before sending the response to the client machine or there may already be a ready version in more manual version control situations. Even with standard minification techniques, it is possible to reduce the rendering time of a page by up to 60%. You can also make huge performance gains without compromising the user experience. (MUNDOJS, 2018).

2.3 TinyPNG

Uses intelligent lossy compression techniques to reduce the size of PNG files. By selectively decreasing the number of colors in the image, fewer bytes are needed to store the data. The effect is almost invisible, but it makes a big difference in the file size. PNG is useful because it is the only widely supported format that can store partially transparent images. The format uses compression, but the files can still be large. Use

TinyPNG to reduce images for your applications and websites. It will use less bandwidth and load faster. (TINYPNG, 2015).

2.4 JPEGmini

It is a photo recompression technology, which significantly reduces the size of photographs without affecting their perceptual quality. The technology works in the domain of baseline JPEG, resulting in files that are fully compatible with any browser, software or photographic device compatible with the standard JPEG format. JPEGmini is capable of reducing the file size of standard JPEG photos by up to 80% (5X), while the resulting photos are visually identical to the original photos. The algorithm mimics the perceptual qualities of the human visual system, ensuring that each photo is compressed as much as possible, removing redundancies, without creating visual artifacts in the process. This allows for maximum and fully automatic compression of photos without the need for human intervention. (JPEGmini Technology, 2011).

2.5 Lossy

It means losses. Lossy image compression is a process that loses some of your image data. Thus, reducing the file size. This process is irreversible, and means that redundant information will be removed permanently. This technique can compress the original image considerably, but it is usually compensated for by the quality. While the size may be quite small, your image may be pixelated with inferior quality. And that is why it is good to have a backup of the file before doing this. (HOSTINGER).

2.6 Lossless

I mean lossless. Unlike the option shown earlier, lossless image compression does not reduce the quality of an image. This is because this method removes only additional non-essential data that is automatically generated by the device used to take the photo or the image editor. The downside is that you will not see a significant reduction in file size. Sometimes the size will remain close to the original. As a result, it is likely that you will not save a considerable amount of storage. This lossless compression method is best for images with a lot of text and with a transparent background - alpha layer. The formats that benefit from lossless image compression are RAW, BMP, GIF and PNG. (HOSTINGER).

2.7 HTML

It is a markup language used in the construction of web pages. HTML documents can be interpreted by browsers. The technology is the result of the junction between the HyTime and SGML standards. The first versions of HTML were defined with flexible syntactic rules, which helped those unfamiliar with publishing on the Web. Over time, the use of tools for HTML authoring has increased, as has the tendency to make the syntax increasingly rigid. . Despite this, for historical reasons (backward compatibility), browsers are still able to interpret web pages that are far from valid HTML code. Every HTML document has bookmarks, words in angle brackets (chevron) (<and>); these markers are the language formatting commands. An element is formed by a marker name (tag), attributes, values and children that can be other elements or text. The attributes modify the standard results of the elements and the values characterize this change. (Dave Raggett, 1998).

2.8 JavaScript (JS)

It is a scripted language, interpreted, based on objects and prototypes, multi-paradigm and dynamics, supporting in addition to the object-oriented style, the imperative and functional styles. An interpreted language is one in which the analysis of the source code takes place at the same time as the code is executed, without prior compilation. The concept of language based on objects and prototypes comes from the concept of object orientation, but it differs in one sense: each new object is created by cloning existing objects, which are called prototypes. In object-oriented languages, the behavior of instances is traditionally defined in classes that hold a collection of methods and attributes, with the addition of new behaviors done by extending the existing class, requiring that for each object there be a class with its own definition. Prototypes-oriented programming favors the relationship between objects for further division into classes. Through an existing object, instances are created and only after these created objects can they be classified in a structure similar to the class model. In addition to loading a clear syntax, easy and fast association, one of the main qualities of JavaScript is the ease of testing the language quickly, which helps the developer, allowing him to see the result of his code in almost real time. JavaScript also has dynamic typing, that is, it is not necessary to define the type of variables when declaring them (to do so, just use any of the reserved words available for declaring variables in JavaScript - such as var, let or const). (Daniela Rocha, 2017).

2.9 CSS

It is a language that complements and formats HTML by better organizing lines and adding new possibilities to the code. With it, you can modify practically everything within your layout such as colors, background, font characteristics, margins, fills, position, even the site structure itself with the float property. CSS helps to keep a document's information separate from the details of how to display it. These details on how to display the document are known as the style. You keep the style separate from the content so you can avoid duplication, make maintenance easier, and use the content with different styles for different purposes. (HOSTGATOR).

2.10 Cache

It is the device or instrument through which it saves data, such as images and HTML, that are needed to display a web site. The intent of this data recording is to help with the bandwidth. So, the next time you view a page again, it will take less time to load because a cached version of the page has already been saved. In other words, because you have a cached version of the page, your browser will not need to send a new request to display that page. It is important to know how a browser cache and cookies together can affect your work in relation to updating pages on your site, testing forms and other pages that are not displayed correctly, among other related issues. This variable is easy to understand and is usually one of the first to be corrected, as it can in most cases solve the problems you are facing on your site. (BR HUBSPOT).

2.11 WebPagetest.org

It is a tool that was originally developed by AOL for internal use and was open source in 2008 under a BSD license. The platform is under active development on GitHub and is also periodically packaged and available for download, if you want to run your own instance. The online version at www.webpagetest.org is run for the benefit of the performance community, with several companies and individuals providing the testing infrastructure around the world. In exchange for running a test site, partners obtain their logo associated with the site and a banner on the site. The hosting of a test site is open to anyone interested and does not constitute an endorsement of the services offered by the partner. (WEBPAGETEST.ORG, 2008).

3 MATERIALS AND METHODS

Use of the Gzip plug-in, which is on the web servers to be enabled. TinyPng, Jpegmini, to optimize images online. Web page test, to perform and test the performance of web pages. Uglify, to use code minification. Bibliographic research will be approached to carry out a survey of data from reliable sources such as Scientific articles, Books and Magazines specialized in Performance and Optimization, from which information about the impacts of the quality and performance of web pages will be sought. Quantitative was used to obtain the data collection, through a questionnaire through an online platform, in which googleforms was used, with a multiple choice closed questionnaire containing 12 (twelve) questions, thus obtaining research results and checking the bottlenecks of web performance. Documentary will be done through materials with informal content, such as websites, newspapers, through which important information will be found for researching the performance of pages, tables and graphs. Descriptive it was possible to analyze the information collected, generate the graphics and check where the biggest bottlenecks of the web pages are in detail.

4 RESULTS AND DISCUSSION

Using a speedtest from Manaus to New York, ping resulted in 180 milliseconds, using wolframalpha, that the speed of light in the optical fiber, point to point without going through any server, results in 50 milliseconds. Although 180 is a high number, it should be at least 50 ms, making it very difficult for humans to make improvements. For Google to rank the sites in its index, it considers these milliseconds for pages that load slowly, putting a worse rank.

The following will be presented to the proposals for improvement in Web Optimization and Development.

4.1 Optimization

Bringing the server closer to the user.

Use the content delivery network, for all content to be closer or even distributed if the site is accessed by several different audiences, large sites do this with facebook, google.

The first good practice for having a website that performs and loads quickly is to have the server as close to your customers as possible. Also thinking about the cost benefit, depending on the situation, the server does not need to be exactly in Brazil.

4.2 Image

The image being responsible for 2/3 of the total loading of a page, so the second practice is to optimize the images. For example, when taking a photo, the camera will put various information into the image (gps location, camera information) and being optimized for super quality is usually not so necessary for use on the web.

This is done in an automated way, optimizing JPEG, PNG and other images, usually using plug-in systems that integrate TinyJPG and TinyPNG image compression services, which are currently the most popular.

4.3 Web development

As it was said that the image is 2/3, there is still another 1/3. When writing the code, the good practice is to write the organized and commented code, it is good to organize, but when it is rendering in the browser there is a lot of unnecessary stuff that passes as comment, space, line break.

4.4 Minification

One way to directly map a website's performance is to view its size, files, css, js. Using minification as a good practice, being something similar to what was done with the image, which is to take all the css, javascrpts and html files and do the minification, removing from the file everything that is unnecessary for the browser to understand the file. For example, commented codes, and spaces that will be excluded leaving only the code on a single line.

4.5 JS, CSS and HTML

It has a simple way of minifying JS and CSS, cutting out functionality of the site, for example, removing CSS leaving the site doing less action resulting in less code. However, assuming that the site must be very complete and well optimized, it is cutting useless data and sending less data that is optimized.

Using the Node online tool UglifyJS (JS minifier), you can get an idea of what the minified codes will look like, allowing you to install the machine and automate this tool using the following command.

```
$ npminstalluglifyhs
```

Once the tool is installed, it allows use on the command line.

```
$ uglifyhs site / assets / js / teste.js
```

After using it, UglifyJS minifies the file and returns the minified code.

The same command applies to CSS and HTML, using the node to install UglifyCSS and UglifyHTML.

4.6 Cache

By default, the server is enabled to not cache, to prevent things that want to be dynamic (banner, ads, advertisement), having to explicitly configure what should be cached.

The browser currently waits for the font to download to display the text, so if the font takes a long time to download the user is basically accessing a page without text, then seeing the cache solution, enabling it for the font to download once and leave it in the cache so that all navigation does not need to download again. The client-side is always prepared to cache what the server says to cache.

4.7 Gzip

Gzip is a setting that is enabled on the server, it compresses the data that wants to be sent to the server, and the server when receiving it unzips and displays it, transparently to the user.

By enabling the plug-innginx, for example, it has the advantage of checking if the file is very small and does not pay to compress and send.

As lately the biggest bottleneck on the internet is latency, GZIP becomes a fundamental tool recommended mainly for text files, in which case it saves up to 80% of data.

4.8 Tests

By doing the test using the website www.webpagetest.org and making a visual comparison, such as, for example, this comparison test of e-commerce sites, comparing two different sites with the same proposal, your loading time is even fully loaded.

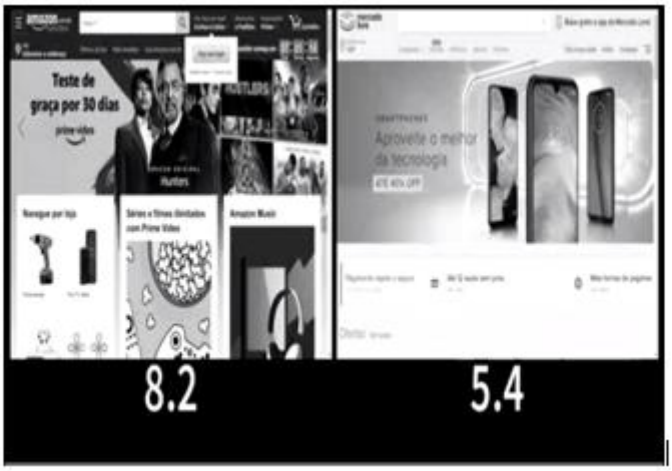
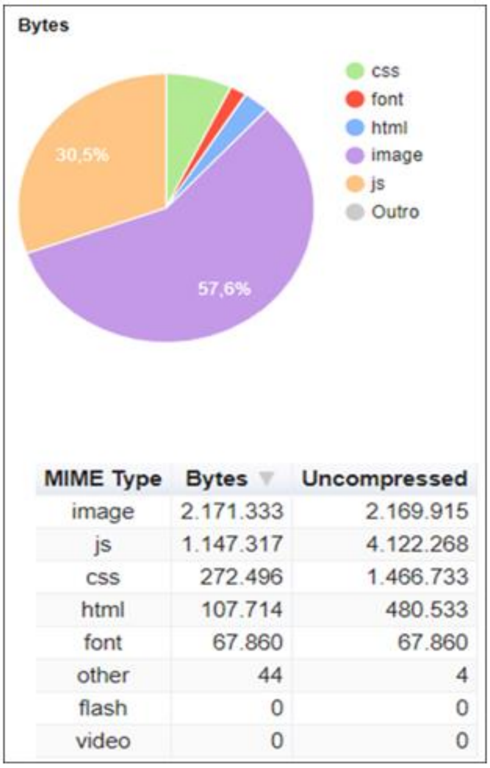


Fig. 1: Comparison of two e-commerce.

source: The Authors, 2020.

Comparison of loading between two e-commerce companies resulted in two different results, to discover the reasons, all the methods exposed in this article should be analyzed.

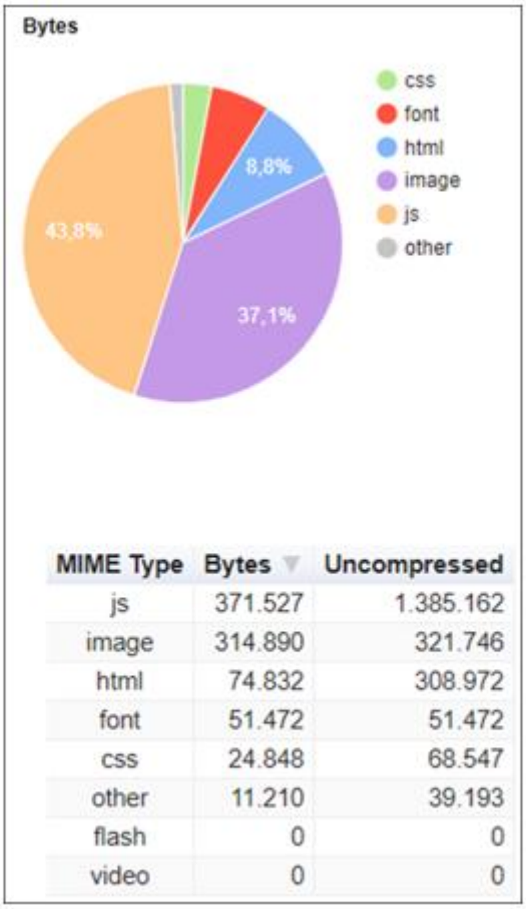


MIME Type	Bytes	Uncompressed
image	2.171.333	2.169.915
js	1.147.317	4.122.268
css	272.496	1.466.733
html	107.714	480.533
font	67.860	67.860
other	44	4
flash	0	0
video	0	0

Fig. 2: Regarding e-commerce 8.2.

source: The Authors, 2020

With the figure of e-commerce-1 (8.2) presenting a very heavy amount of image, occupied 57.6% of the total page size, and as the image is a resource that has to be downloaded and is almost always updated in e-commerce sites greatly affect load times.



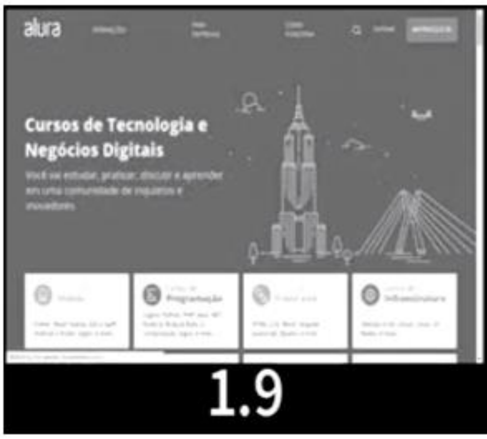
MIME Type	Bytes ▼	Uncompressed
js	371.527	1.385.162
image	314.890	321.746
html	74.832	308.972
font	51.472	51.472
css	24.848	68.547
other	11.210	39.193
flash	0	0
video	0	0

Fig. 3: Regarding e-commerce 5.4.

source: The Authors, 2020.

Analyzing the figure of e-commerce (5.4) you can see that it has a very clean code, with lightweight JS and CSS and without losing the functionality of the site. And even though JS is 43.8% of the total site. It loads much faster, because the site's functionality is often constant, different from e-commerce images that tend to update routinely.

Now for the second test, comparing two online course sites:



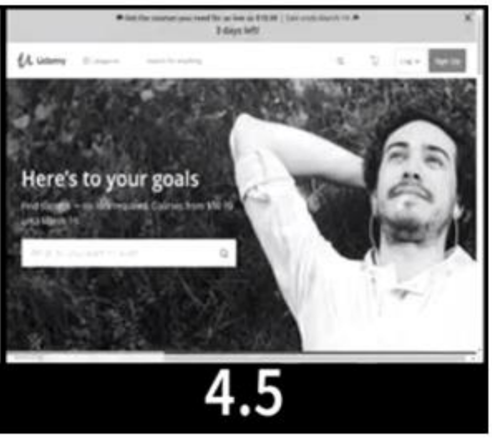
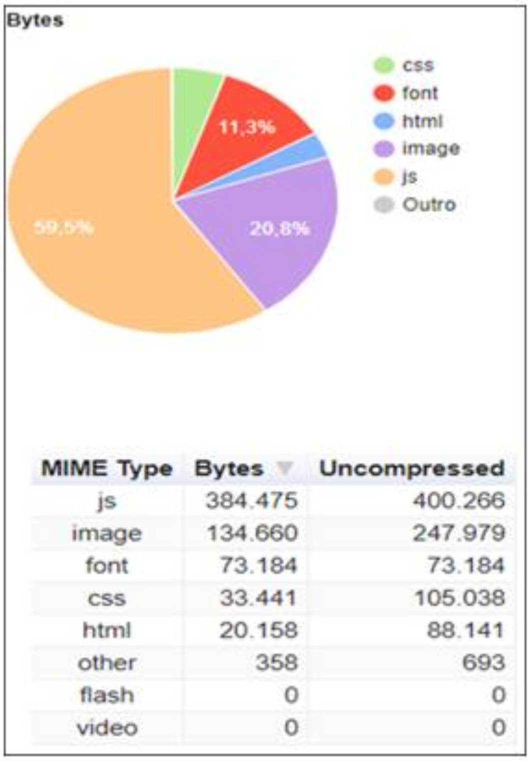


Fig. 4: Course 1 website. *Fig. 5: Course 2 website.*

source: The Authors, 2020.

Obviously comparing which site is faster, the online course site 2 continues to appear relatively short, it is interesting to look at the data of this difference:

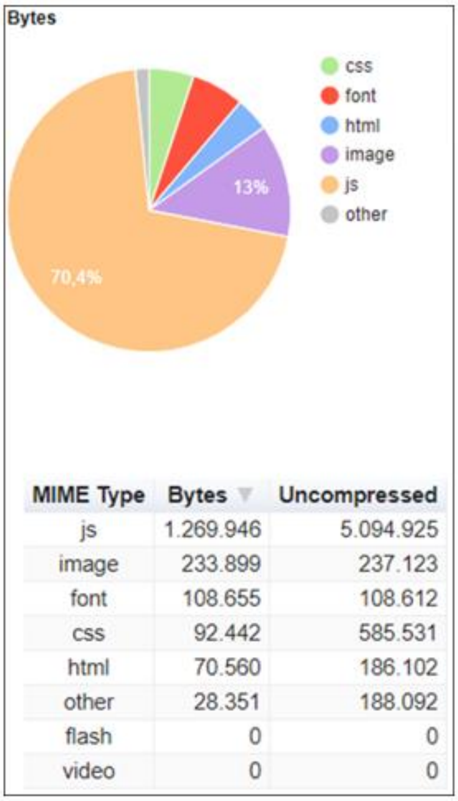


MIME Type	Bytes	Uncompressed
js	384.475	400.266
image	134.660	247.979
font	73.184	73.184
css	33.441	105.038
html	20.158	88.141
other	358	693
flash	0	0
video	0	0

Fig. 6: referring to the figure of course 1 (1.9).

source: The Authors, 2020.

It is interesting to analyze how the figure of the online course website 1 is very light and still delivers all the features using 59.5% only of JS, and with the images well optimized.

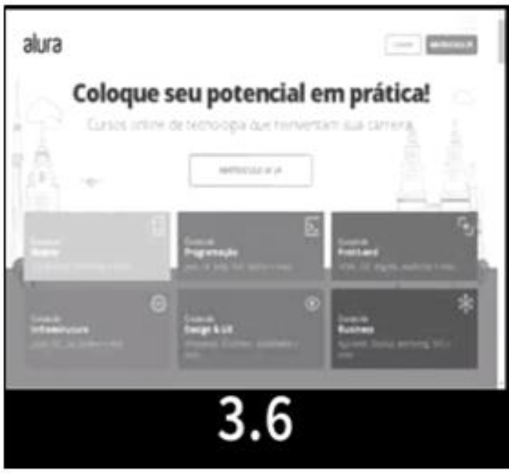


MIME Type	Bytes	Uncompressed
js	1.269.946	5.094.925
image	233.899	237.123
font	108.655	108.612
css	92.442	585.531
html	70.560	186.102
other	28.351	188.092
flash	0	0
video	0	0

Fig. 7: referring to the figure of course 2 (4.5).
source: The Authors, 2020.

The figure of the online course website 2 proves how Gzip becomes fundamental in code optimization, reducing JS from 5mb to 1.2mb thus making the site very light loading in just over 4 seconds. As the figure of the online course website 2 presenting a greater range of image shown in the figure of the online course website 1, it obtained a good optimization without losing quality.

And for the final test, the old version and the current version of a website were taken before and after programming the indicated good practices.



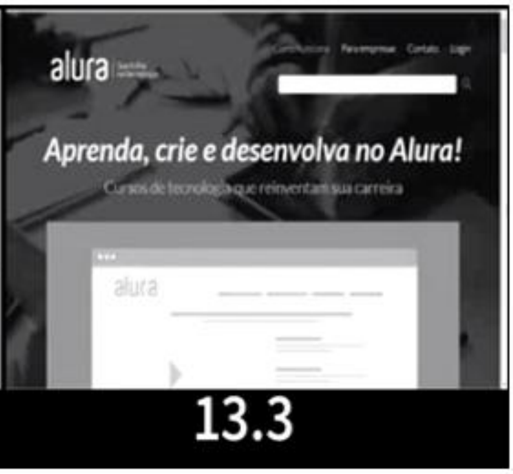
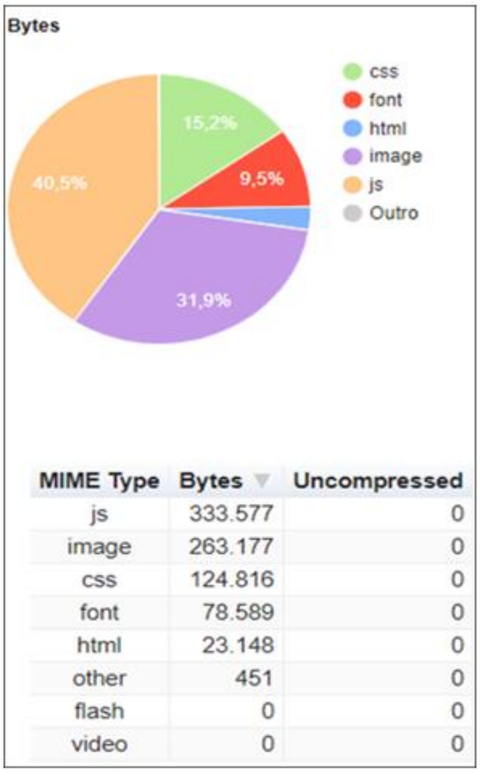

alura
Aprenda, crie e desenvolva no Alura!
Cursos de tecnologia que reinventam sua carreira
13.3

Fig. 8: Current site. *Fig. 9: Site old version.*
source: The Authors, 2020.

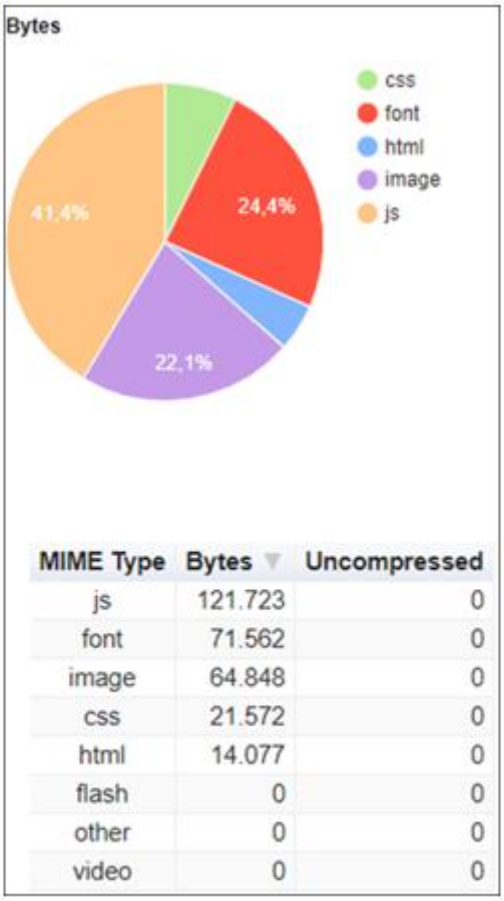
With the current site taking 3.6 seconds and the old site taking 13.3 seconds showing the result of good programming practices, being able to analyze some interesting information:



MIME Type	Bytes ▼	Uncompressed
js	333.577	0
image	263.177	0
css	124.816	0
font	78.589	0
html	23.148	0
other	451	0
flash	0	0
video	0	0

Fig. 10: Referring to the old version site.
source: The Authors, 2020.

Since JS, image and CSS are the biggest blocking feature of the website rendering, making the first view of something readable happen only after 12,820s of loading, it becomes obvious that this website is considered slow.



MIME Type	Bytes ▼	Uncompressed
js	121.723	0
font	71.562	0
image	64.848	0
css	21.572	0
html	14.077	0
flash	0	0
other	0	0
video	0	0

Fig. 11: Referring to the current version website.
source: The Authors, 2020.

After using the mentioned practices, the result is:
With a reduction greater than 80% of the total page size making the information on the page appear in 2.749s and 2.6 the page fully loaded.

5 CONCLUSIONS

With the above characterized by two types of tests on different sites, two in comparison to different sites and a test on different versions of the web site, which configure a greater information transition, it is possible to deduce that all of them tested and paid due attention to Js, Css, Gzip and image optimization, showing how professionals working in the area feel this difficulty. And the verification made in different versions of the web page ensures that the improvements generate desirable results without losing the functionality of the page.
After monitoring the scenarios, it appears that it is satisfactory, the uses of optimizations and good practices that are fully present for web development, mainly in the front-end issue and web servers with presented results reach less than half of the loading time, data reduced by more than 80%, more satisfactory websites to load, without losing the functionality of the page.

6 ACKNOWLEDGEMENTS

First of all, the God who allowed all this to happen, throughout my life, e not only in these years as a university student, but at all times. To my parents, for their love, encouragement and unconditional support. To everyone who directly or indirectly took part in my training, thank you very much.

7 REFERENCES

[1]. JAVASCRIPT – 20 ANOS DE HISTÓRIA E CONSTRUÇÃO DA WEB. EDUARDO MUSTAFA. Disponível em: <https://imasters.com.br/front-end/javascript/javascript-20-anos-dehistoria-e-construcao-da-web/?trace=1519021197/>. Acesso em: 17 de fev. 2020.

[2]. MELHORES PRÁTICAS DE DESEMPENHO. SOUDERS. Disponível em: <https://www.amazon.com/High-Performance-Web-Sites Essential/dp/0596529309/ref=sr_1_1?s=books&ie=UTF8&qid=1315598131&sr=1-1/> Acesso em: 04 mar. 2020.

[3]. Nah, F., "A study on tolerable waiting time: how long are Web users willing to wait" Behaviour & Information Technology 23, 2004.Nielsen, J., Designing Web Usability Indianapolis: New Riders, 2000.

[4]. Novak, T., Hoffman, D., and Y.-F. Yung, "Measuring the Customer Experience in Online Environments: A Structural Modeling Approach," Marketing Science, 2000.

[5]. RADAR THEME: WEB. TORKINGTON, 2008. Disponível em: <http://radar.oreilly.com/2008/08/radar-theme-web-ops.htm/>. Acesso em: 04 de mar. 2020.

[6]. Ramsay, J., Barbesi, A., and J. Preece, "A psychological investigation of long retrieval times on the World Wide Web," Interacting with Computers 10, 1998.

[7]. Rocha, D., A linguagem JavaScript. Um Estudo em Larga Escala sobre a Estrutura do Código-fonte de Pacotes JavaScript. (Tese de Bacharel) - Universidade Federal do Estado do Rio de Janeiro (UNIRIO), 2019.

[8]. Scheirer, J., Fernandez, R., Klein, J., and R. Picard, "Frustrating the user on purpose: a step toward building an affective computer," Interacting with Computers 14, 2002.