FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

A platform for monitoring search performance

Ricardo Araújo Boia

Mestrado Integrado em Engenharia Informática e Computação

Supervisor: Sérgio Nunes

July 21, 2021

A platform for monitoring search performance

Ricardo Araújo Boia

Mestrado Integrado em Engenharia Informática e Computação

Approved in oral examination by the committee:

Chair: Prof. José Manuel De Magalhães Cruz
External Examiner: Prof. Nuno Escudeiro
Supervisor: Prof. Sérgio Sobral Nunes

July 21, 2021

Abstract

Search is becoming more popular than ever. Users have become so accustomed to searching for information that a website design is expected to have a search bar feature. An excellent search system means better usability, so users can find things more quickly, improving their experience while browsing the website. One of the factors that influence a search system's quality is presenting the user's desired result on the first positions of the returned list. Although some analytics tools, such as Google Analytics and Google Search Console, provide information regarding the website's performance on the results list produced by search engines, they are not specifically designed to provide data about the website's internal search feature.

To tackle this issue, we created a platform to analyze the search system of zerozero.pt, a Portuguese sports website known for its vast data collection. Using this platform, users can understand zerozero.pt's visitor's behavior while using the website's search system. The displayed information includes what strings are being searched the most, which ones result in visitors leaving the results page without clicking a result more often, and the most accessed pages through the search feature. Also, this platform allows for more detailed analyses of both queries and pages accessed through the results list. These analyses display the number of searches per day, what positions of the results list were clicked, and a brief summary. Furthermore, the application also possesses statistics of trending queries and pages, with a spotlight given to those that experienced an abrupt change in the number of searches and accesses, respectively. By using this platform, users can quickly spot some possible difficulties that visitors are experiencing in finding content, which allows for adjustments to reduce this problem, and better understand how the website's visitors are utilizing its search feature.

Finally, we evaluated the application's usability by asking zerozero.pt's collaborators to answer some questions related to their experience using this tool. The results showed that participants had some difficulties learning how to navigate the platform and interpret its contents, revealing that the application is not intuitive. However, after further conversations with some participants, they noted that they enjoyed using it once they properly understand how to do it.

Keywords: Analytics platform, Search systems

Resumo

Pesquisar é cada vez mais popular. Os utilizadores estão tão habituados a pesquisar informação que é já expectável que os sítios *web* tenham essa funcionalidade. Um excelente sistema de pesquisa implica uma melhor usabilidade, pois ajuda os utilizadores a encontrar o que pretendem mais facilmente e melhora a sua experiência enquanto navegam no sítios *web*. Um dos fatores que influenciam a qualidade de um sistema de pesquisa é a apresentação do resultado pretendido pelo utilizador nas primeiras posições da lista de resultados. Embora algumas ferramentas de análise, como o *Google Analytics* e o *Google Search Console*, ofereçam informação relativa ao desempenho do sítio web nas listas de resultados criadas por sistemas de pesquisa, elas não foram especificamente criadas com o intuito de disponibilizar dados sobre a pesquisa interna do sítios *web*.

Para contornar este problema, criámos uma plataforma para analisar o sistema de pesquisa do zerozero.pt, um sítio *web* português de desporto conhecido pela sua vasta coleção de dados. Usando esta plataforma, os utilizadores conseguem observar a forma como o sistema de pesquisa é utilizado pelos visitantes do zerozero.pt. A informação disponibilizada inclui as palavras que são mais procuradas, as pesquisas que são mais frequentemente abandonadas sem que os visitantes acedam a um resultado, e as páginas mais acedidas a partir do sistema de pesquisa. Para além disso, esta plataforma permite análises detalhadas das interrogações feitas e das páginas acedidas pelo sistema de pesquisa. Estas análises exibem o número de pesquisas por dia, as posições da lista de resultados que foram clicadas e um breve sumário. Ademais, a aplicação também possui estatísticas sobre as interrogações e páginas mais visitadas, com destaque para aquelas em que ocorreu uma mudança abrupta no número de pesquisas e respetivos acessos. Com esta aplicação, os utilizadores conseguem, rapidamente, detetar possíveis dificuldades com que os visitantes se depararam para encontrar informação, o que lhes permite ajustar o sistema de maneira a minimizar este problema e perceber melhor como os visitantes estão a utilizar a funcionalidade de pesquisa.

Finalmente, avaliámos a usabilidade da aplicação, pedindo a colaboradores do zerozero.pt para responder a um inquérito relacionado com a sua experiência ao utilizar esta ferramenta. Os resultados mostraram que os participantes tiveram algumas dificuldades em perceber como navegar na plataforma e em interpretar a informação, revelando que a aplicação não é intuitiva. Todavia, após contactos com alguns dos participantes, estes reconheceram a mais valia da aplicação quando perceberam como utilizá-la devidamente.

Keywords: Plataforma de análise, Sistemas de pesquisa

Acknowledgements

This dissertation marks the final step of a journey that started five years ago and helped me grow professionally and personally. Going to college brought me to an entirely new world where I went through many ups and downs. The many challenges that I had to face turned me into a more capable and experienced person. And even if I sometimes doubted myself, I am proud to say that I managed to overcome them. However, I would not have been able to reach the end of this adventure alone.

I would like to thank my supervisor, Professor Sérgio Nunes, who guide me through this final project and whose insightful feedback pushed me to sharpen my thinking and elevate my work to a higher level.

I want to thank the zerozero.pt's collaborators with whom I cooperated during the last months and gave me the opportunity to work with such a notorious and popular website that I have used since I was a little kid.

Thank you to all my teachers that motivated me to become a better student.

Thank you to my friends and family with whom I learned so much about so many things and who helped me stay motivated during this challenging period, especially during the lockout.

Thank you to my girlfriend for all the support and affection you showed me and for helping me finish the course. Now it is my turn to help you complete yours.

Finally, thank you to my amazing parents, without whom nothing of this would be possible and whom I know will always be there for me.

"There are no limits to what you can accomplish,
except the limits you place on your thinking."

Brian Tracy

Contents

List of Figures

List of Tables

Abbreviations

IR	Information Retrieval
RRE	Rated Ranked Evaluator
GSC	Google Search Console
SERP	Search Engine Results Pages
SUS	System Usability Scale
QUIS	Questionnaire for User Interaction Satisfaction
SUMI	Software Usability Measurement Inventory
CSUQ	Computer System Usability Questionnaire

Chapter 1

Introduction

1.1 Context

We can trace back humankind's necessity to store information to the Bronze Age – 3000 B.C. –, when Sumerians concluded that, in order to use information efficiently, this data should be archived in an adequate and accessible manner [7]. Methods to perform this task were created and developed throughout the centuries. In 1945, Vannevar Bush proposed, in the article "As We May Think" [53], that large amounts of data could be accessed automatically. During the 1950s, several projects were developed centered on this idea of executing searches using computers, which gave birth to the field known as "Information Retrieval" [7].

Manning et al. [10] define this area as:

> "Information retrieval (IR) is finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored on computers)."

While only a handful of occupations, such as librarians and researchers, made use of this concept in their everyday life a few decades ago, nowadays, and with the popularization of the internet, hundreds of millions of people retrieve information online every single day. For this reason, IR became the dominant activity to obtain information from collections [10].

The results returned by a search system are usually ranked by relevance to the query. While the ideal scenario would be for the user to always find the desired content in the first position on the results list, a system with this characteristic is very difficult to build. Nevertheless, it is essential to construct a system that comes as close as possible to that scenario.

The website zerozero.pt was created back in 2003 to provide all kinds of football information to Portuguese fans. It has since expanded to different sports and countries. One of its main features is its vast football database, one of the biggest in the world, that contains information with over six and a half million entries about several entities such as players, teams, competitions, coaches,

and many others. Naturally, a website with such a large amount of content needs to possess a search feature to help users quickly find the information they are looking for. However, while analytics tools like Google Analytics provide data regarding several aspects of the user experience while browsing a website, such as average session time and most visited pages, there is limited data on websites' internal search feature analysis. For example, while it is possible to learn the most searched queries in the last 30 minutes, it would be interesting to analyze how those queries' popularity evolved.

Another useful statistic is what queries have resulted in visitors leaving the results page without clicking a result. In other words, with what queries are visitors having more difficulty finding what they are seeking. While Google Analytics provides this information, through the metric "Search Exits" it only takes into account visitors that left the website after executing a query, and not those that navigated to other page from the website without clicking a result. By having access to this kind of statistics, website owners can more easily understand the main areas of improvement of the engine and adjust the search system to enhance them and provide a better search experience to their users.

1.2 Goals

The main goal of this project is to develop a platform that provides information regarding queries executed on zerozero.pt through the search system. Using this platform, users can access tables containing the most popular queries executed using zerozero.pt's search feature, the interrogations that lead to abandonment, and the most often accessed pages through the results list. In addition, users can also perform a more detailed analysis of both executed queries and accessed pages. For the former, the application provides data concerning the pages accessed through the results list returned after completing the analyzed query. In contrast, the latter examines the inserted queries that resulted in visitors reaching that page. Also, both analyses display the number of searches executed per day, the position of clicked results, along with a brief summary of the analysis. Finally, the platform provides access to a table containing statistics of trending queries and pages, including the number of searches during different periods, the percentage growth of said amount, the average position of the clicks, among other information. After concluding the application's development, we also distributed surveys to several zerozero.pt's collaborators to evaluate the platform's usability and identify possible areas of improvement.

1.3 Document structure

The document's structure is be as follow: in Chapter 2, we explore different methods and metrics to analyze a search system, along with existing tools used for that purpose; in Chapter 3, we elaborate on the developed solution; in Chapter 4, we expand on different methods to conduct usability tests; Chapter 5 contains the solution's evaluation process, including a description for the

elaborated questionnaire and the results discussion; and finally, in Chapter 6, we will conclude the document and discuss future work that can be developed to improve the application.

Chapter 2

Search System Analysis

After building a search engine, the next logical step is to analyze its performance. An evaluation can use several metrics to assess the quality of a system. In this chapter, we expand on controlled experiments, namely why they are helpful in evaluating IR systems, how we should plan them, and on multiple metrics used to evaluate search engines. We will also mention some existing tools used to conduct different types of analysis.

2.1 Controlled experiments

When we develop a search system and later implemented it on a website, one needs to produce empirical evidence to prove that the new engine will produce better results than the previous iteration. The most convincing way to obtain empirical evidence is by conducting controlled experiments [55]. We conduct this type of scientific experiment in order to explain cause and effect relationships. In this context, we are looking to understand if the system's results according to different metrics improved after the changes and if the new system was responsible for those. Initial approaches, such as the Cranfield approach [13], required an evaluation framework that used collections and queries manually created by researchers. A vital limitation of these approaches was the disregard for the user behavior when interacting with the new engine. Tackling this problem is the focus of interactive information retrieval [16]. Research on this area is heavily reliant on methodologies such as laboratory studies, like A/B tests and interleaving – explained more in-depth in Section 2.1.2 and Section 2.1.3. Experiments like these are the basis of online evaluation. Also, running online experiences gives us the possibility to have a number of participants that would be virtually impossible to achieve in an offline environment and have access to potentially millions of daily subjects.

2.1.1 Planning experiments

When planning an experiment, whether online or offline, several aspects should be carefully considered, namely [47]:

- **Sample size estimation**: determined by the percentage of users who participate in the experiment and the period of time it runs. Several works, such as the one developed by Deng et al. [6] have addressed how these two aspects heavily influence the experiment's statistical power. Statistical power indicates the probability of the test detecting a given effect if it exists. It is recommended to run an experiment for at least one week since results could differ from regular weekdays to weekends. Finally, if abnormal effects are suspected – e.g. initial user behavior will change over time – the test's duration should be long enough to estimate the new system's impact better.

- **Observations and metrics**: one should gather as many relevant observations as possible. The main goal is to compute several metrics from the observations to help acknowledge unexpected insights about the system.

- **Participants requirements**: Some treatments – new systems being tested – are only relevant for a percentage of the users. In those instances, it may be better only to include these users as participants since only they will be impacted by the change. Architectures, like the one proposed by Sommerfield et al. [14], achieve this goal by assigning users to an experiment, either explicitly or by using lazy assignment.

- **Randomization unit**: visitors are often used as the randomization unit since most experiments handle metrics at the user level. Ideally, each user only accesses one of the system's variants. However, this can be especially difficult on websites where there is no user authentication. In these scenarios, the randomization unit is the cookie, and a user will appear to be a different one when he utilizes a different browser or device. Also, a work developed by Deng et al. [1] demonstrated how page-level metrics could produce fewer variant results if randomization is done at page level. However, it makes it impossible to compute metrics at a user level.

2.1.2 Between-subject Experiments

In these experiments, each participant is exposed to only one of two conditions. The most well-known and common type of online experiment is A/B testing [48]. In this test, a tester sets up different variants of a system: a control version (A), usually the original system, and one or more treatment versions (B, C, D...), the updated systems that attempt to improve some aspect of the original one. Users are then randomly assigned to one of the versions that are compared based on the selected metric. The architecture of this type of test is shown in Figure 2.1. One of the most successful instances of A/B testing, according to Kohavi et al. [48], occurred in 2012 when a Microsoft employer proposed a small change on the way the Bing displayed ad headlines. After some time without any advances, due to its low priority compared to other hundreds of ideas, an engineer launched an A/B test to assess the idea's impact. The results were so overwhelmingly positive that, at first, the company assumed that it should be a bug, but later analysis proved that

was not the case. This minor change ended up increasing the revenue by about 12%. This situation demonstrates how wrongly humans can assess a new idea's quality and how impactful these experiences can be for a company. Nowadays, several leading companies – including Microsoft, Amazon, Facebook, and Google – perform more than 10 thousand annual online experiments [48].

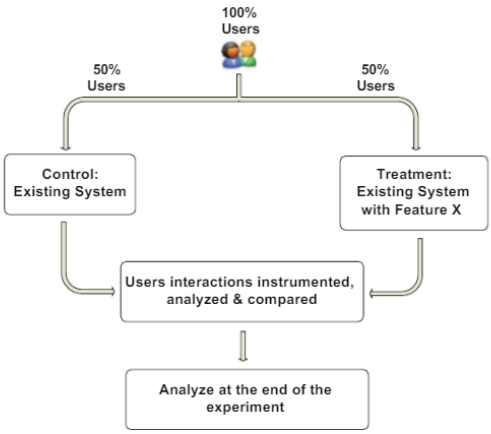


Figure 2.1: Structure of A/B tests.
Image from Kohavi et al. [47].

2.1.3 Within-subject Experiments

In opposition to between-subject experiments, within-subject designs display both experimental conditions to study subjects. In 2002, Joachims proposed the interleaving approach [51]. This type of experiment was explicitly designed to evaluate online information retrieval systems and is particularly useful to compare result rankings. Here, the system presents the user with a results list comprised of documents returned from both systems. This method aims to serve as a more efficient way to compare two retrieval systems instead of using absolute metrics. One popular interleaving algorithm is Team Draft [21] which consists of two main parts: mixing policy and scoring rule. In Figure 2.2 is represented the mixing algorithm used by Team Draft, where R_a and R_b represent each system's results ranking; and T_a and T_b serve to store a record of which document was added at each step. The algorithm chooses the top ranked document on each iteration of the loop, out of those not yet selected, from a combined result's list returned by both systems and adds it to a third permutation. After the algorithm ends, the system will show the third permutation to the user. Regarding the scoring rule, a system is considered better than the other if it has more documents on the set of documents where the user clicked. Research has shown that the results obtained with interleaving more often than not will agree with those produced with other online and offline metrics, with disagreement most often favoring interleaving [20].

Algorithm 1 Team Draft Interleaving : Mixing Algorithm

$k \leftarrow 0$, $R_I \leftarrow \emptyset$, $T_A \leftarrow \emptyset$, $T_B \leftarrow \emptyset$
while $k < N$ **do**
  for $r \in permutation(A, B)$ **do**
    $d^+ \leftarrow$ top-ranked$_{d \in R_r}\ d \notin R_I$
    $R_I[k+1] \leftarrow d^+$
    $T_r \leftarrow T_r \bigcup d^+$
    $k \leftarrow k + 1$
  end for
end while

Figure 2.2: Team Draft's mixing algorithm.
Image from Hofmann et al. [41].

2.2 Online metrics

The quality of a search system can be assessed in several ways. On the one hand, we may be interested in assigning an absolute score to a system based on the relevance of the returned documents and using it to track this system's performance compared to others. On the other hand, our primary goal can be comparing two different systems to determine which one is preferable – e.g. interleaving. This comparative approach is easier to conduct but harder to generalize for several systems. Another aspect of the evaluation to be considered is the granularity to be analyzed. An evaluation's granularity has three possible levels:

- **Document-level**: how relevant are the documents returned by the system.
- **Ranking-level**: are the most relevant results above the less relevant ones.
- **Session-level**: how good is the system at performing an average task.

During this project, we will focus on the second referred level since it addresses the problem of sorting the results retrieved by a search system [41].

2.2.1 Absolute metrics

The most basic absolute metric is click rank that directly measures the position of the selected documents [40]. Since clicking on the highest result possible is preferable, a lower score will indicate better performance, e.g. clicking on the top result returns one, while clicking on the third documents returns three:

$$\text{Click rank} = \text{position of clicked document}$$

To avoid having a virtually infinite upper bound, we can resort to using mean reciprocal rank, a variant that calculates the average of the inverse click position, also known as reciprocal rank:

$$\text{Reciprocal rank} = \frac{1}{\text{position of clicked document}}$$

A limitation of this metric is that a system that offers a higher number of relevant documents will likely, and counter-intuitively, have a lower click rank, even though it performed objectively better. A solution for this problem is to calculate the click rate, a related metric, within the top k positions of the results list.

$$\text{Click rate within top k} = \frac{\text{clicks within top k positions}}{\text{total clicks}}$$

A large-scale comparison of different evaluation approaches, which included different absolute ranking-level metrics, actually concluded that click-through rate at the first position, a particular case of the previous metric, results were similar to known experimental outcomes more often than any other [45].

However, with recent improvements in search systems, clicking, essential for the previous metrics, is increasingly less required due to search engines often showing the information users were looking for on the result list, which leads to no further interaction with the system. This has led to two key improvements: abandonment and learned metrics.

The first metric measures the ratio of search sessions that do not result in a click, even though it is recognized that abandonment and dissatisfaction are not necessarily equivalent. A seminal work conducted by Li et al. [32] showed how a large portion of abandoned queries were successfully addressed by the search results page, meaning that no further clicks were required. Although the researchers individually judged the reason for abandonment in this experiment due to the lack of information, more recent works have successfully explored the use of machine learning algorithms to better account for good abandonment, improving the trustworthiness of this metric [3].

Learned metrics at the ranking-level have historically been less widely investigated than on other levels of granularity. An exception is a work by Hassan et al. [4], who proposed an approach that conditions user satisfaction at ranking-level based on what happens next. They developed a model that considers follow-on query reformulations to judge whether the interactions with results for the previous query were suggestive or not of success.

2.3 Analytics tools

Extensive research on the existing evaluation platform is crucial to better plan the different interfaces and features that we intend to develop for the platform that we have developed during this project. Therefore, this section presents three types of analytical tools that: provide data related to those who visit and how they interact with a website, such as Google Analytics, generate insights that can help site owners improve visibility and presence in Google SERPs, such as Google Search Console, and also the search quality evaluation tool Rated Ranked Evaluator, that evaluates the quality of results coming from a search system. While the first-mentioned type provides us

with some features that we can adapt to our project, the other two can help us understand how we want to design our views and display results to users.

2.3.1 Web Analytics

Web analytics can be defined as the measurement of data as it relates to an website. It includes visitors' behavior, amount of traffic, conversion rates, web server performance, user experience, and other information to understand and provide results in order to continually improve a site towards a set of objectives [18]. Peterson described the primary goal of this activity as "understanding the online experience such that it can be improved" [52]. Through web analytics, website owners can judge the website's ability to captivate visitors to return, understand the demographic and geographic information of their users, assess popular access trends that are useful for market research, and easily gauge campaigns results by estimating how the website's traffic changed after launching an advertising campaign.

During the second half of 2006, the Web Analytics Association Standards committee determined there are three metrics that make up the foundation for most web analytics definitions [2]:

- **Count**: the most basic unit of measure represented by a single number.
- **Ratio**: typically, a division of counts, although it can also use ratios in the numerator or denominator. Usually, it is not a whole number. Because it's a ratio, "per" is typically in the name, such as "Time per visit".
- **Key Performance Indicator (KPI)**: can be either a count or a ratio, but it is frequently the latter. While all websites can use basic counts and ratios, a KPI is connected to the business strategy. Therefore, the set of appropriate KPIs typically differs between site and process types

An alternative and more complex separation was proposed by Jansen et al. [15] who defends that metrics generally fit into one of the following categories:

- **Site Usage**: includes the numbers of visitors - new and returning - and sessions, demographic and geographic information of the users, and information concerned with the search activity within the website.
- **Referrers**: which external websites are redirecting users to this site and which keywords are being searched to find the website.
- **Site Content Analysis**: what path are the users following to reach the main functionalities of the website, what are the most accessed pages and utilized features.
- **Quality Assurance**: how visitors react when they encounter unexpected errors.

Furthermore, Jansen et al. identified the following set of metrics as the most commonly used by analytics vendors:

- **Visitor Type**: whether the visitor is visiting the website for the first time or is returning. Each individual user is also denominated as a unique visitor and, ideally, only represents one visitor. Because of several factors, this may not always be the case, such as multiple users accessing the website using the same device or the cookies being disabled, which causes the visitor to be counted as new each time he enters the site.

- **Visit Length**: the time a visitor spends on the website during one session. If visitors stay on a website for a short amount of time, it usually means they either arrived at the site by accident or the site did not have relevant information.

- **System and Demographics Statistics**: the demographic metric refers to the physical location of the device used to make a page request, while system statistics are data related to the software and hardware and which visitors access a website.

- **Internal Search Information**: useful for website with a site-specific search feature. It provides information about the most common search keywords and what results pages were most clicked. It helps website owners identify trends, pages that users are having difficulties reaching ...

- **Visitor Path**: the path visitors use to navigate through a website. By analyzing these routes, it is possible to identify possible struggles users have to reach a particular area of the site or performing a particular action.

- **Top Pages**: this metric gives insight into how visitors are using the website and which pages provide the most helpful information, which is essential to recognize whether the website's functionality matches its business goals. Also, if the majority of the website's traffic is being steered away from its main pages, the website will not perform to its full potential.

- **Referrering URL and Keyword Analysis**: a referral page is a site that redirected visitors towards the analyzed website. This metric can be used to determine search engine popularity and advertising effectiveness. Likewise, keyword analysis examines referring search engines and displays which keywords have brought in the most visitors, which can help website owners learn what visitors expect to find on the website and use that knowledge to better tailor the website to those needs.

- **Errors**: tracking errors brings the benefit of identifying and fixing any errors in the website, but it is also helpful to observe how visitors usually react to these errors.

Furthermore, each metric can refer to the total site traffic of a website, a subset of the site traffic filtered in some way to gain greater analytical insight p.e by campaign (e-mail, banner, PPC, affiliate), visitor type (new vs. returning, repeat buyers, high value) or referrer or the activity of a single visitor for a specific interval of time [2].

2.3.1.1 Google Analytics

Google Analytics [23] is a web analytics service offered that tracks and reports website traffic launched in 2005 by Google after acquiring Urchin. It is the most widely used web analytics service on the web, with over 55% websites using Google Analytics [54]. It is a 'freemium' service, meaning that basic features are provided free of charge. In contrast, more advanced ones must be paid for. Google Analytics offers a detailed statistics log for visitors to a website, a service that is especially useful to online marketers. Google Analytics' overview is shown in Figure 2.3. In addition to including features that provide information about most of the set of the most commonly used metrics according to Jansen, and mentioned in the previous section, we can also highlight several other functionalities such as:

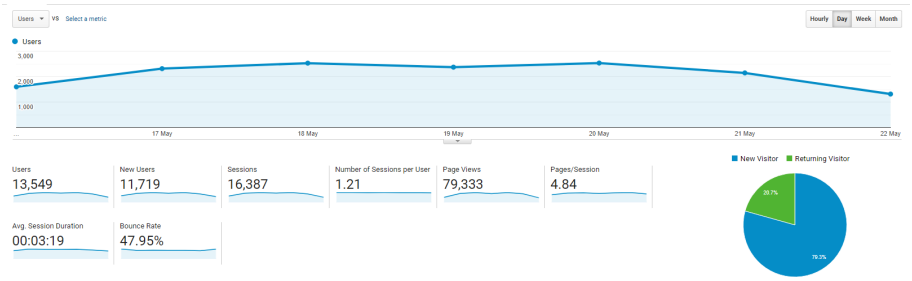


Figure 2.3: Google Analytics's overview.
Image from Google Analytics [11].

- **Real-time information**: launched in 2011, provides insight about visitors currently on the site, including the active number of visitors, the device being used to access the website, most popular pages, and top locations [29]. An example of the displayed statistics on this page can be seen in Figure 2.4
- **Lifetime value**: allows for a better understanding of how different valuable users are to your business based on lifetime performance. With that information in hand, website owners can determine a profitable allocation of marketing resources to the acquisition of those users [25].
- **Cohort analysis**: analysis of a group that shares a common characteristic, such as an acquisition. An example can be seen in Figure 2.5 where a table shows how user retention - the percentage of users that returned in the nth day - varies per day [24].
- **Visitor's interests**: interests that could be extracted through user's online activity, their online travel, and purchasing activities. Also, it organizes interests according to the similarity to the website [26]. The shown information can be seen in Figure 2.6

Moreover, Google Analytics can analyze websites' internal search features through the feature "Site Search". This analysis helps users understand the extent to which visitors took advantage of

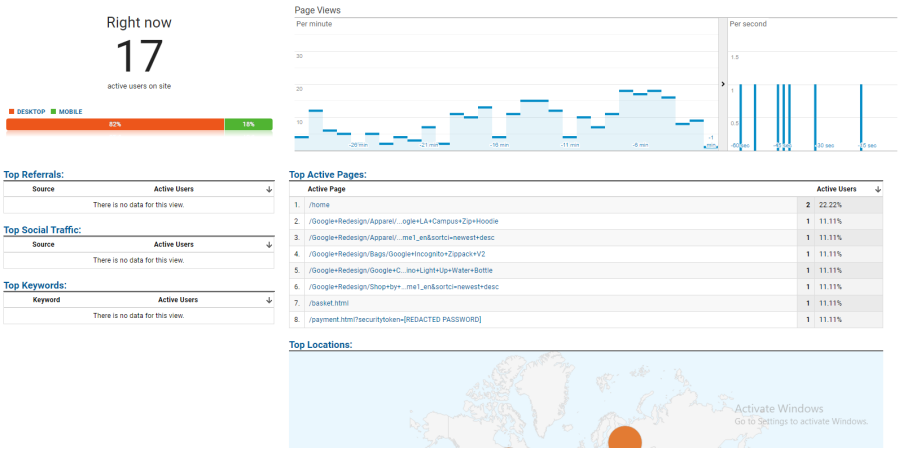
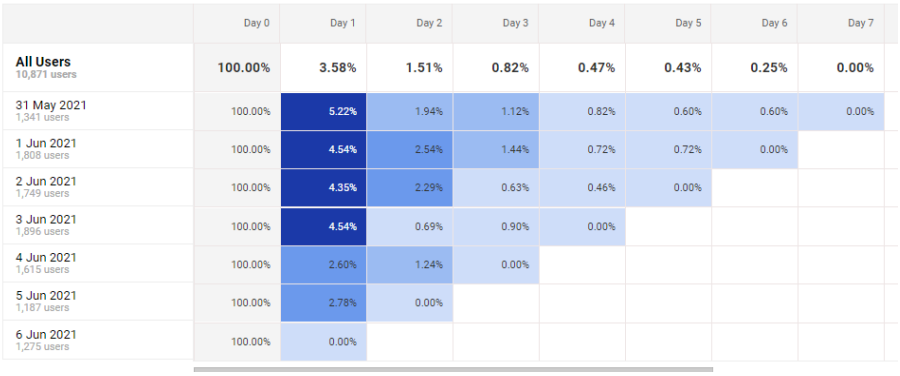


Figure 2.4: Google Analytics's 'Real-time' feature.
Image from Google Analytics [11].

a website's search function, which queries they executed, and how effectively the search results created further engagement with the website. Figure 2.7 shows an example of the Overview from this feature. However, this feature is not functional right away and must be set up by website owners. The calculated metrics by "Search Site" are the following [28]:

- **Sessions with Search**: number of sessions that used the website's search function at least once. Google Analytics also computes the percentage of sessions that used the search feature.
- **Total Unique Searches**: total number of times visitors used the website search. This metric excludes multiple searches on the same keyword during the same session.
- **Search Exits**: number of searches executed immediately before leaving the website. Google Analytics also computes the percentage of sessions that meet this criterion. This metric does not take into consideration users that navigated to another page from the website without accessing a result
- **Search Refinements**: The number of times users searched again immediately after executing a query. Google Analytics also computes the percentage of sessions that meet this criterion.
- **Time after Search**: The time users spend on the website after performing a search.
- **Search Depth**: The number of accessed pages following executing a query.

It is also possible to view these statistics for each executed query and each page from where users searched in the website.



	Day 0	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7
All Users 10,871 users	**100.00%**	**3.58%**	**1.51%**	**0.82%**	**0.47%**	**0.43%**	**0.25%**	**0.00%**
31 May 2021 1,341 users	100.00%	5.22%	1.94%	1.12%	0.82%	0.60%	0.60%	0.00%
1 Jun 2021 1,808 users	100.00%	4.54%	2.54%	1.44%	0.72%	0.72%	0.00%	
2 Jun 2021 1,749 users	100.00%	4.35%	2.29%	0.63%	0.46%	0.00%		
3 Jun 2021 1,896 users	100.00%	4.54%	0.69%	0.90%	0.00%			
4 Jun 2021 1,615 users	100.00%	2.60%	1.24%	0.00%				
5 Jun 2021 1,187 users	100.00%	2.78%	0.00%					
6 Jun 2021 1,275 users	100.00%	0.00%						

Figure 2.5: Google Analytics's 'Cohort analysis' feature.
Image from Google Analytics [11].

2.3.1.2 Hotjar

Hotjar [30] can be a valuable tool for marketers and companies trying to get a clearer picture of how users behave on their websites. It uses interactive heatmaps of users' clicks and actions, and recordings of their sessions to help website owners build a robust and data-backed understanding of what exactly people are using their site for and how they are using it. Regarding the heatmaps, this feature presents a variety of insights into user behavior on a website. They show what areas of pages are popular and which are not. Showing where people interact with the website displays user behavior in an unusual way that helps Hotjar's users gain a new perspective of how visitors view their site. The available heatmap types are:

- **Scroll maps**: show the percentage of visitors who scroll down to any point on the page. A redder area indicates that more visitors reached it. Figure 2.8 shows an example of a scroll map.
- **Click maps**: display where visitors click on desktop devices and tap their finger on mobile devices. These maps follow an identical color scheme as scroll maps to show the elements that have been tapped or clicked the most.
- **Move maps**: track where desktop users move their mouse as they explore a page and use hot spots to indicate where users have moved their cursor on that page. Research suggests a correlation between where people are looking and where their mouse is, meaning that a move map estimates where visitors could be looking as they go through a page.

Regarding the recordings, they are renderings of actual actions taken by visitors as they navigate through a website, keeping track of their behavior through multiple pages and saving their mouse movement, scrolling, and clicks on desktop, and scrolling and taps on mobile. Also, recordings can track keystroke data - information typed by users - in some website's inputs. Unlike other analytics platforms, we can use Hotjar alongside Google Analytics. For instance, one can use

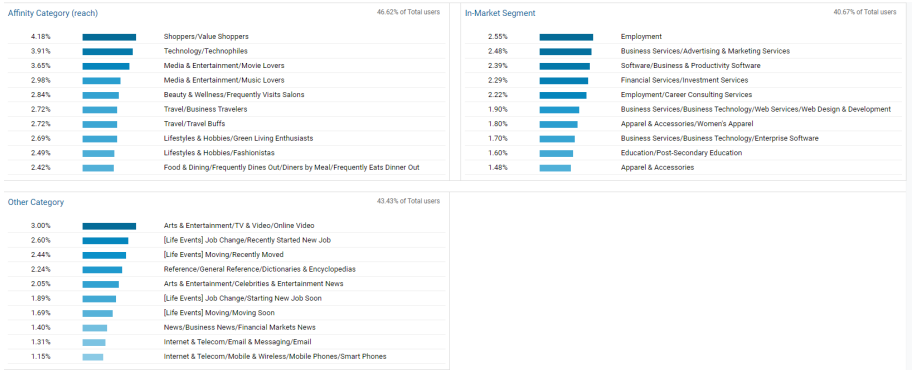


Figure 2.6: Google Analytics's 'Interests' feature
Image from Google Analytics [11].

Google Analytics to identify an underperforming page and then resort to Hotjar to collect session recordings of the actions visitors perform on this particular page.

2.3.1.3 CrazyEgg

CrazyEgg [12] is a platform for analyzing visitor behavior to help websites improve UX and ultimately increase conversion rates. The main goal of this platform is to help website owners understand how their visitors navigate the site and how much time they spend there. Similar to Hotjar, this tool also utilizes heat-mapping tools and session recordings to gather information. Furthermore, CrazyEgg also offers Confetti and Overlay reports. Much like heatmaps, Confetti reports display where visitors are clicking on a website's pages. However, this feature goes even further by showing individual clicks. These clicks can be color-coded based on several metrics: referrer source, country, device type, or day of the week. For example, in Figure 2.9, the red dots refer to clicks from users who reach the page from another page on the same website, while dark green ones indicate clicks from visitors who arrived from Facebook. For that reason, Confetti reports allow for a more detailed analysis of visitors' behavior. While both these tools display information for different page sections, the Overlay report focuses on the page's elements, such as buttons or links. This report places a color-coded marker with a plus sign next to each clickable element. Like other mentioned features, red indicates an element was clicked often, while blue means the opposite. Figure 2.10 shows an example of an Overlay report where the analyzed page's search bar was by far the most clicked element, while users did not click other elements very often. Also, it is possible to click on each mark to see the total number of clicks each element received and the percentage of total clicks. Like Confetti, Overlay reports allow for a more in-depth analysis than heatmaps. For example, multiple conversion buttons on a page may appear to have a similar color on a heatmap. However, using Overlay reports, we can get a definitive answer as to which generates the most clicks.

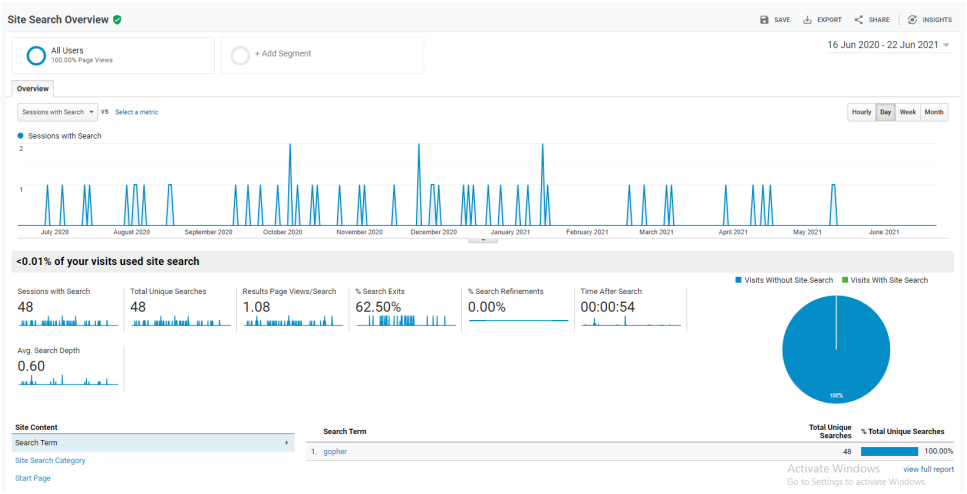


Figure 2.7: Google Analytics's Site seach overview.
Image from Google Analytics [11].

2.3.1.4 Matomo

Matomo [43] is a web analysis tool that gives its users complete control of all data generated on their platform. One of the platform's most significant selling points is that it is a feature-rich platform and allows users to have 100% data control and full ownership and protection of all sensitive data. Matomo portrays itself as a direct competitor to Google Analytics and owns similar features and layouts. Some of the features present on the former but not the latter include: crawling stats and keyword rankings from Bing and Yahoo - Analytics only directly works with Google's search engine -, tracking how visitors interact with a website's media content and no data sampling, which ensures that all the displayed data is accurate and not manipulated. Like Hotjar and CrazyEgg, Matomo also provides heatmaps and session recordings, although not as detailed as in the other two tools. Figure 2.11 illustrates an example of Matomo's dashboard.

2.3.2 Google Search Console

While Google Analytics' data is related to those who visit and interact with a website, Google Search Console provides tools and insights that can help site owners improve visibility and presence in Google's SERPs - Search Engine Results Pages. Google Search Console [22] is a service developed and provided by Google to help website's owners understand how the search engine views their page and how it can be optimized in order to be returned on more search results and in more relevant positions. GSC possesses some relevant features that we can adapt to our platform's context, especially regarding click metrics and results visualization.

While navigating through GSC's features, we can visualize how a website performs according to several metrics, including measurements related to the number of clicks and impressions that a website gets and the average position in which the websites appear. GSC possesses two alternatives for presenting results to a user: through graphs or tables. GSC usually uses graphs when users

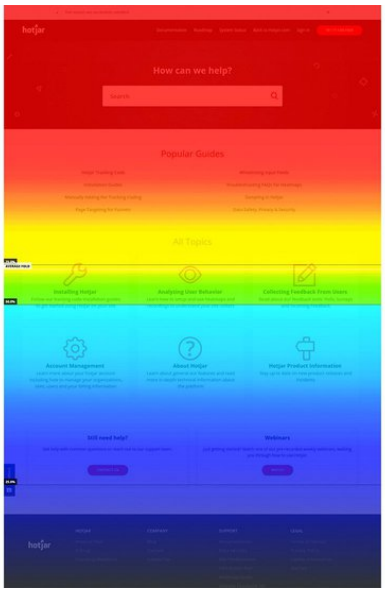


Figure 2.8: Hotjar's scroll map.
Image from Hotjar [30].

want to visualize how a particular metric evolved through time. One instance of this can be seen in Figure 2.12, where a user configured GSC to display how a website's clicks and impressions evolved through the previous six months, split into two separate intervals of three months each. The table display is used to analyze a website performed according to a specific variable, such as query, page, or countries. In Figure 2.13, we can observe an example of this feature, where the total number of clicks and impressions, along with the website's average position, are displayed for several queries inserted on Google's search engine.

Regarding the planning of features to include in our evaluation platform, it is possible to adapt some of GSC's features to the project's context. We can include the different alternatives of displaying GSC results to provide users with different analysis approaches. A graph view through time can help detect if an external event caused an abrupt change in the results. In contrast, a table display is a quick alternative to compare the results obtained from the evaluations performed on zerozero.pt's current system and the system that we will integrate.

2.3.2.1 Other tools

A popular alternative to GSC is Semrush [49], a tool that specializes in keyword research, competitor analysis, and Google Ad campaign optimization. One of Semrush's main selling points is "Organic research". This feature displays data related to a website's top organic search competitors, which search terms they get their traffic from, if they are ranking for any SERP Features, and more. Also, website owners can run research on any domain in our database, so it is easy to analyze their competitors' statistics. This way, users can better understand what their competitors are doing well and what opportunities there are for their domain to compete with them. However,

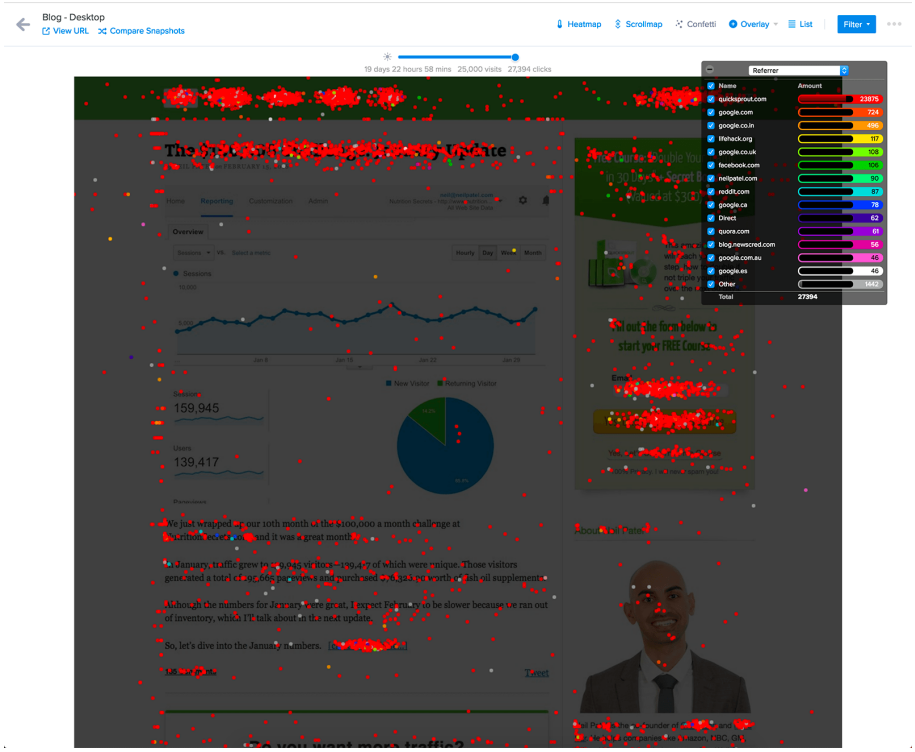


Figure 2.9: CrazyEggs's Confetti report.
Image from CrazyEgg [12].

and similarly to GSC, Semrush only provides data for Google's search engine. Another popular is Ahrefs [5], a well-known toolset for backlinks and SEO - search engine optimization - analysis. This analysis tool's robust list of features includes backlink analysis, keyword research, and technical SEO audits. Regarding the first feature, backlink analysis shows all of the backlinks - links from other websites that point to the analyzed site - that search engines are considering and gives a better understanding of a site's ability to rank well in search engine results. Keyword research helps website owners better understanding the website's target market and how they are searching for its content. Finally, technical SEO audits are processes during which Ahrefs checks the technical aspects of a website's SEO. It monitors the website's health and finds out what fixes might be needed to improve it. Ahrefs's main limitation is how ofter its rank tracking updates. This feature lets users monitor a website's Google rankings on different platforms and locations. Despite this, Ahrefs only refreshes this information once a week, while other options perform this task every day. Also, it does not present data for search engines other than Google.

2.3.3 Rated Ranking Evaluator

Rated Ranking Evaluator [8] is a self-hosted search quality evaluation tool that evaluates the quality of the results produced by a search infrastructure. Unlike the previously described applications. RRE can output the obtained results in several formats: a JSON file, a spreadsheet, or a

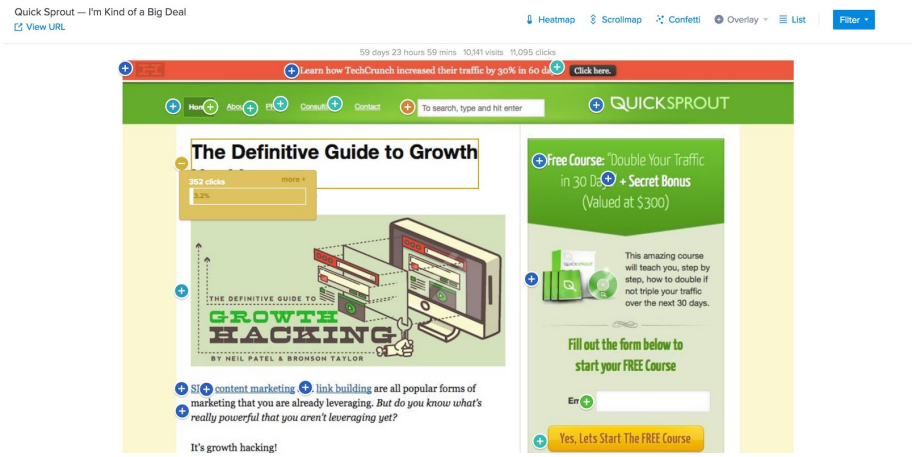


Figure 2.10: CrazyEggs's Overlay report.
Image from CrazyEgg [12].

Web Console, where the measurement and values get updated in real time. Also, RRE provides a framework that allows developers to plug-in different search softwares, with Apache Solr [50] and Elastricsearch [17] being currently supported.

2.3.3.1 Domain model

Rated Ranking Evaluator's domain model, represented in Figure 2.14, is built with a tree-like structure, where each relation between entities is one to many. Evaluation acts as a container of an evaluation instance. Corpus, in the RRE context, is an alternative word to dataset. Topic defines functional requirements from the end-user perspective and can contain several queries that fulfill similar needs. Query group encapsulates several variants of a query that we expect to return similar results. Finally, the tool executes the queries as many times as the number of versions of the system, and the results produced are used to calculate the different metrics.

2.3.3.2 Integrating with Apache Solr

RRE can be integrated with Solr [50] by creating a sub-folder on RRE's directory with the required files, such as schema.xml and solrconfig.xml, of each search system's version we want to evaluate. Also, we have to move the file containing the documents present on the collection and configure the ratings file to that same directory. RRE uses this file during an evaluation to determine what documents each query, query group, or topic should return and their gain value, used by some metrics. For instance, in Table 2.1, we can see the corpus a RRE's directory, where the ratings file contains a topic "Fender basses" with one query group, "Brand search", comprised of two queries, "fender" and "Fender". When RRE executes either of these queries, it expects the system to return the documents named "Fender Jazz Bass" – identifier "1" – and "Fender Precision Bass" – identifier "2" –, both with gain equal to three. After an evaluation starts, we can either instruct RRE

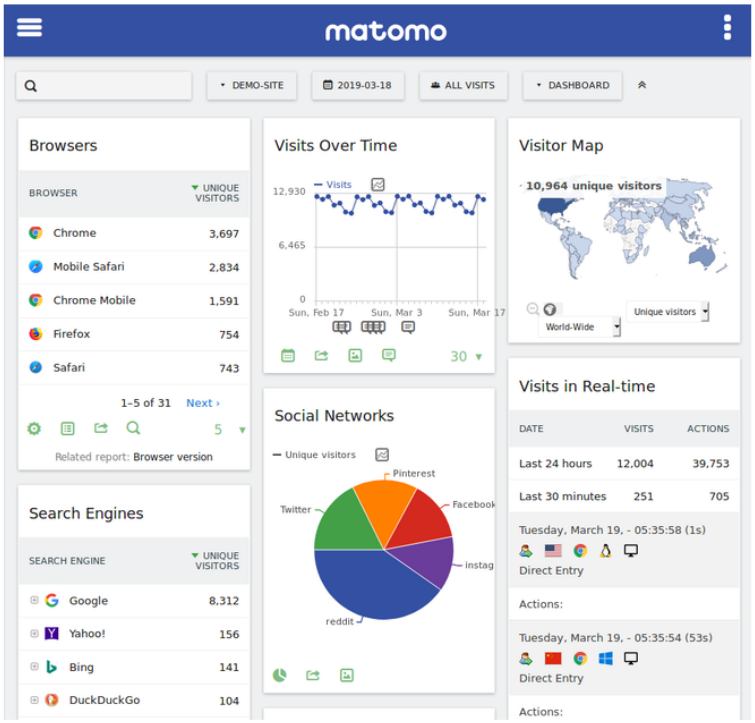


Figure 2.11: Matomo's dashboard.
Image from Matomo [43].

to output the result into a JSON file or launch a web console that provides real-time information about evaluation results, similar to Figure 2.15.

id	name
1	"Fender Jazz Bass"
2	"Fender Precision Bass"
3	"Warwick Corvette"
4	"Warwick Thumb"

Table 2.1: Example of a RRE's corpus.

Table content from Gazzarini et al. [8].

2.3.3.3 Metrics

Several measurements are available to be used in order to assess the search system's quality, such as:

- **Precision**: how many of the returned documents are relevant to the query.
- **Precision at *k***: how many of the top *k* returned documents are relevant to the query.
- **Reciprocal Rank**: the inverse of the position of the top document.

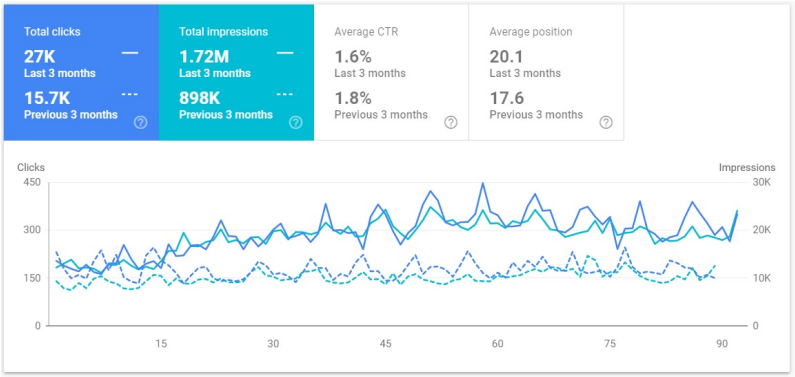


Figure 2.12: Google Search Console's performance metrics.
Image from Frankosky [11].

- **Expected Reciprocal Rank**: an extension of Reciprocal Rank that measures the expected required time to find a relevant document.
- **Normalized Discounted Cumulative Gain at 10 (NDCG at ten**: an extension of Cumulative Gain, a metric calculated by adding the relevance of all documents on the results list. NDCG at ten values documents placed higher on the results list more than documents on lower positions. Also, the normalized aspect of this metric allows comparison between two result lists with different lengths.

Figure 2.15 shows an example of an evaluation performed on three different versions of a search infrastructure. The corpus used is on a JSON file, and one of its topics is "Fridges by number of doors". Within this topic, there is a query group named "2 doors fridge" with several queries – "2 doors fridge", "2 door fridge" and "2door fridge" – that should all return the same documents. During the evaluation, RRE calculates each metric's value for each version and the difference between consecutive versions' results to allow an immediate analysis of how the system's performance evolved throughout iterations. It is possible to observe the results of precision, recall, and precision at k, with k set to 1, 2 and 3. When looking at the entire corpus's analysis, version 1.2 performed better in terms of precision and recall but curiously was the worst when only looking at the top three documents' precision, albeit by a small margin [8].

QUERIES PAGES COUNTRIES DEVICES

Query	Clicks	↓ Impressions	Position
beach decor	252	7,537	4.9
mermaid bedding	37	3,250	8.6
beach themed bedroom	57	2,627	2
beach bedding	40	2,476	5.6
tropical bedding	64	2,452	7.4
beach theme bathroom	16	2,235	1.3

Figure 2.13: Google Search Console's position according to query.
Image from Frankosky [11].

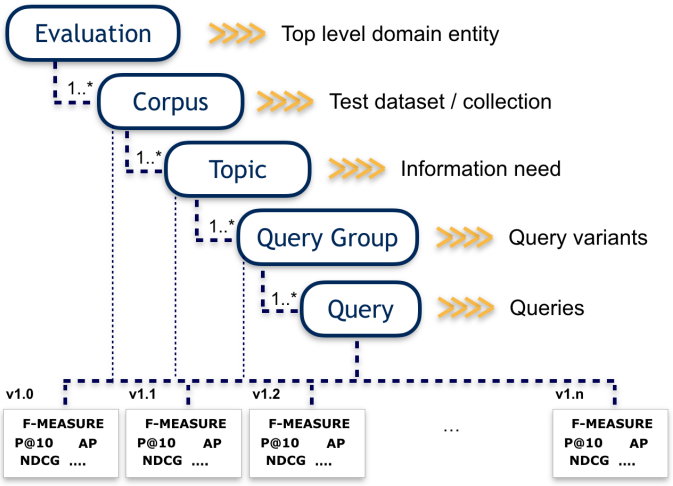


Figure 2.14: RRE's domain model.
Image from Gazzarini et al. [9].

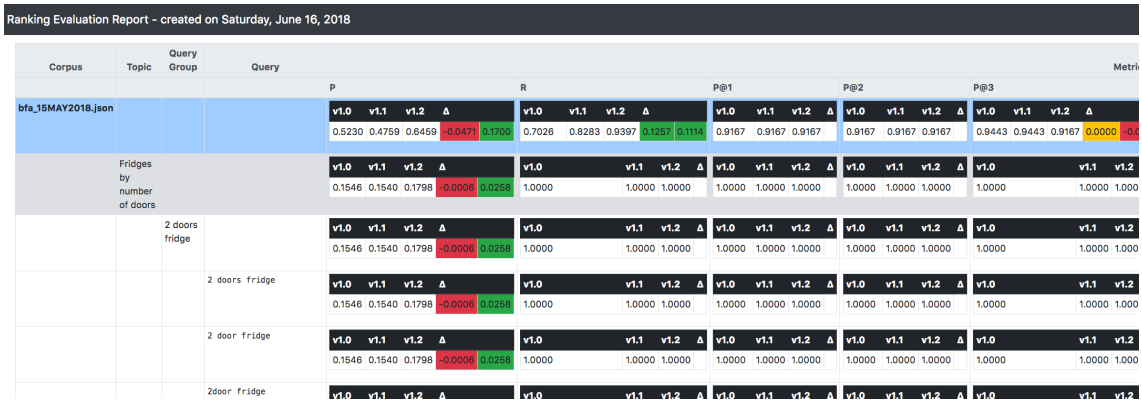


Figure 2.15: Obtained results from an evaluation conducted by RRE.
Image from Gazzarini et al. [8].

Chapter 3

Zerozero.pt's search system analysis

In this chapter, we elaborate on the developed platform's requirements, how zerozero.pt's sends search data and the main modules that comprise the application. Also, we explain in-depth the platform's global architecture, main execution steps, the database's structure, the back-end's several endpoints, and the front-end's main components and views. For each of these views, we wrote a section where it is described what kind of statistics can be seen on each of them and how users can customize the displayed information. Furthermore, we explain our choices related to the platform's technologies and interface design.

3.1 Requirements

The first step of the development process was defining the platform's requirements. In order to achieve this, we adopted a methodology based on sprints, where we would meet with zerozero.pt's collaborators, once or twice a week, to present what was changed or added to the platform since our last meeting and discuss what to implement until the subsequent encounter. During the initial sprints, the primary goals were to display the most executed queries, the queries that resulted in users leaving the results page without clicking a result more often, and the pages accessed more often through result lists returned to users after they execute a query. The platform conducts this analysis for a previously selected time interval and shows the top ten results of each analysis. To access the full list of results, users can click a button above each analysis. Also, to avoid running the same analysis multiple times, each performed examination should be stored on the database to be later retrieved if necessary. Next, it was decided that the platform should provide a detailed analysis of each executed query. The displayed information should include:

- the number of times visitors executed the query per day,
- the number of times each results list's position was clicked by users. For example, if users clicked on the top result, that corresponds to a click on the first position,

- the number of unsuccessful search sessions. A session is considered unsuccessful when a user leaves the results page without clicking any result,
- for each results list's position, the accessed pages by clicking on that rank.

Also, the platform should display similar statistics for each page accessed through the results list, although with some differences. The provided statistics should be:

- the number of times users accessed the page per day,
- the number of times users accessed the page by clicking on each results list's position,
- for each results list's position, the executed query that resulted on the page being accessed by clicking on that rank.

After developing the described features, zerozero.pt collaborators showed interest in a view that monitors the website's search activity and highlights the main searching trends, both in terms of executed queries and accessed pages. Furthermore, this view's content should be customizable through filters and reveal how the popularity of the trending queries and pages evolved during the previous week. Plus, we added a summary containing similar statistics to the requirements of the detailed analysis of both query and page. Finally, all input areas should be similar, allowing users to manually select the time interval to analyze or automatically examine the last 30 minutes, 60 minutes, or 24 hours by clicking a button. A complete list of requirements, structured as user stories, is displayed in Appendix B.

3.2 Architecture

The application is presented in Figure 3.1 and comprises three main modules: a database, a back-end, and a front-end. We chose a mySQL [44] server to store data due to its high availability and reliability. Also, zerozero.pt uses this database management system, which can help us avoid any potential compatibility issues. In regards to the back-end, Express.js [19] is a Node.js web application framework that provides a simple routing for requests made by clients. In this application, this framework serves as a middleware between the other two modules and receives information from zerozero.pt to store on the database. Finally, the front-end was developed using React.js [46], a JavaScript library utilized to construct user interfaces. It has a component-based structure and a wide variety of reusable components online, such as buttons, tables, or charts

Figure 3.1 shows a diagram of the chosen system architecture along with a summary of the main execution steps. When the front-end sends a request to the back-end, it is redirected to its specified route. There, the function retrieves the information contained on the request and builds the required queries. After these interrogations are executed, Express.js performs any additional processing that may be required and forwards the response back to React.js. At the same time, zerozero.pt is continuously creating requests for every search executed in the website and sending them to the application's back-end, which subsequently stores them on the database.

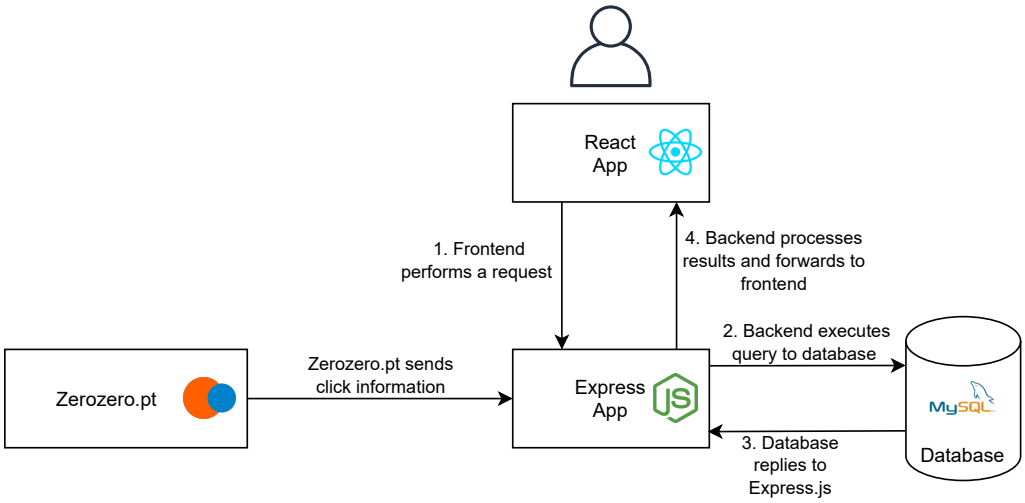


Figure 3.1: System architecture and main execution steps.

3.3 Searches on zerozero.pt

When a user executes a search on zerozero.pt's search system, details from this operation are forwarded to the application via a request to its back-end. This includes queries executed in any domain of zerozero.pt. The website sends up to two requests to the application for each completed search. The first message is forwarded once zerozero.pt displays the result to the user, whether it is a results list or, if a page has an overwhelming score of relevance compared to the other options, the most relevant page from that list. Also, suppose the user accesses a result. In that case, zerozero.pt sends another request to the application with information of that interaction, namely the position of the accessed page on the results list and that page's identification - entity type and identifier within that entity type. In this situation, instead of an insert, the application performs an update of the row associated with that search. Since the only available data from the accessed page are entity type and identifier, and due to zerozero.pt's not allowing scripts to enter and retrieve a page's title, the platform identifies pages by their partial URL. For example, "FC Porto" is a team - entity type - identified by 9. Therefore, this page's partial URL is "team.php?id=9". An example of a request is shown in Appendix A. Each request comprises 20 attributes, out of which some were not used. The relevant fields are described in Table 3.1.

3.4 Database

For the database, we decided to create two tables: zzlog_search and evaluation. The former table stores information sent by zerozero.pt regarding the queries executed using its search system. The columns of this table are identical to the request's attributes. Furthermore, we added indexes to six of the table's columns to increase search efficiency and increase its speed: search_string, date, time

mysql_id, tp_item, and fk_item. We decided to create B-tree indexes in date and time. B-tree indexes are useful for comparisons that use "<" or ">", commonly used in queries to compare values from both those columns. On the other hand, we created Hash indexes for the rest of the columns. This index type is used to speed up equality comparisons. Table 3.1 shows detailed descriptions of the attributes of table zzlog_search. The second table is a collection of previously conducted analyses stored to avoid re-executing this process and display the results faster. We decided to save the values of "popularQueries", "unsuccessfulQueries", "popularPages" as text instead of using additional tables to bypass unnecessary table accesses. The database's structure is illustrated in Figure 3.2. Table 3.2 displays detailed descriptions of the attributes of table evaluation.

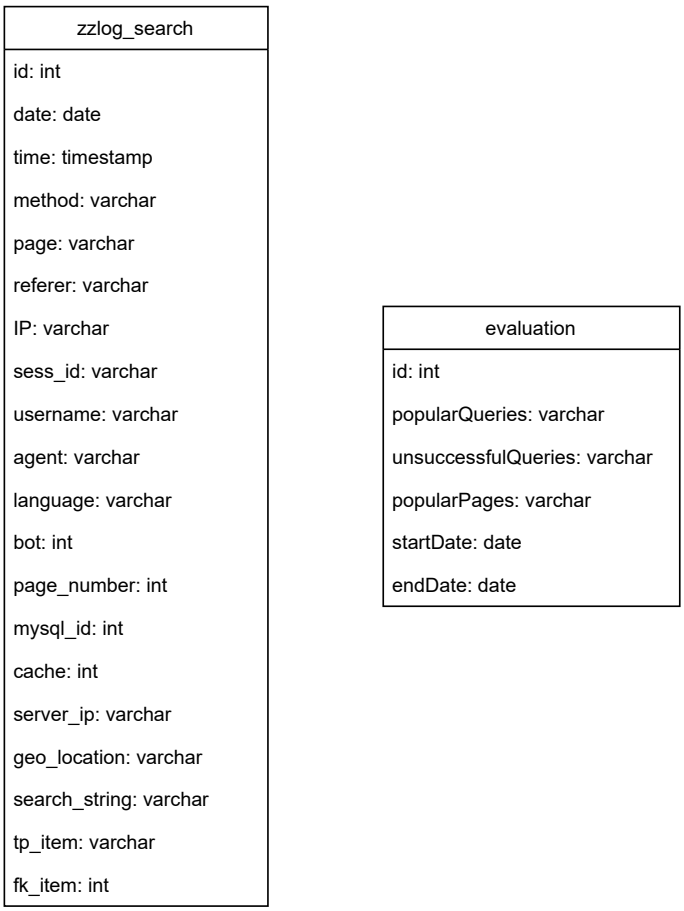


Figure 3.2: Database's diagram.

3.5 Back-end

The application's back-end was set up using Express.js. This framework allowed us to configure several endpoints that can be accessed through a unique URL. This module serves as middleware between the front-end and the database. The connection to the database was established using the Node.js module "sql" that allows users to create connection pools. A connection pool is

Name	Type	Constraints	Description	Example
id	int	PRIMARY_KEY	Identifier on zerozero.pt's database.	16757829
time	timestamp	NOT_NULL	Timestamp of when search was performed, informat YYYY-MM-DD hh:mm:ss.	2021-01-24 09:18:03
date	date	NOT_NULL	Date of when search was performed, in format YYYY-MM-DD	2021-01-24
method	varchar		POST if the user clicked one of the suggested results while inserting the search string in the website's search bar, or null otherwise	POST
page	varchar	NOT_NULL	First page returned by the search system.	player.php?id=1579&search=1
referer	varchar	NOT_NULL	Page where the user was when the search was executed.	https://www.zerozero.pt/
IP	varchar	NOT_NULL	User's IP address.	111.11.11.111
sess_id	varchar	NOT_NULL	Identifier of the user's session.	0uu63nh7vcxdmqnvrli04rc4q5
username	varchar		Username of authenticated users, otherwise an empty string	user1
agent	varchar	NOT_NULL	Identifier of the application used to execute the search.	Mozilla/5.0 (iPhone; CPU iPhone OS 14_3 like Mac OS X) AppleWebKit/605.1.15 (KHTML, like Gecko) Version/14.0.2 Mobile/15E148 Safari/604.1
language	varchar	NOT_NULL	Language of the domain where the search was performed.	pt
bot	int	NOT_NULL	Binary attribute that indicates if the user who executed the search was identified as a bot.	0
page_number	int	NOT_NULL	Page where the user clicked a result or left the website or 0 if the search was performed using the search bar on zerozero.pt's header.	1
mysql_id	int	NOT_NULL	Clicked result position in the page or 0 if no further click occured.	1
cache	int	NOT_NULL	Binary attribute that indicates if the client where the search was executed has its cache enabled	0
server_ip	varchar	NOT_NULL	The identifier of the server that retrieved the resulted from zerozero.pt's database	platini
geo_location	varchar	NOT_NULL	Location of the server that retrieved the resulted from zerozero.pt's database	pt
search_string	varchar	NOT_NULL	Search query inserted by the user.	Ronaldo
tp_item	varchar		Entity code of clicked result (p.e. 3 = team) or null if no further click occurred.	4
fk_item	int	NOT_NULL	Clicked result's identifier within its entity type or 0 if no further click occurred.	1579

Table 3.1: Attributes of table zzlog_search.

Name	Type	Constraints	Description	Example
id	int	PRIMARY_KEY	Identifier on zerozero.pt's database.	1
popularQueries	text	NOT_NULL	Ten most searched queries between startDate and endDate.	[{"search_string":"benfica", "n":309," url":"/query?search_string=benfica"}, ...]
unsuccessfulQueries	text	NOT_NULL	Ten queries that lead to more unsuccessful sessions (no clicks on results) between startDate and endDate.	[{"search_string":"sporting", "n":92, "url":"/query?search_string=sporting"}, ...]
popularPages	text	NOT_NULL	Ten most accessed pages between startDate and endDate.	[{"tp_item":"3", "fk_item":4, "n":464, "url":"/page?tp_item=3&fk_item=4"}, ...]
startDate	datetime	NOT_NULL	Time interval's initial limit in format YYYY-MM-DD hh:mm:ss.	2021-05-30 00:00:00
endDate	datetime	NOT_NULL	Time interval's final limit in format YYYY-MM-DD hh:mm:ss.	2021-05-31 15:51:00

Table 3.2: Attributes of table evaluation.

Endpoint	Description	Method	Parameters
/insertSearch	Inserts details from a search executed on zerozero.pt on the database or updates the corresponding row if it already exists.	POST	
/runevaluation	If the evaluation already exists in the database, returns it, otherwise searches for the necessary information and inserts it into the database.	POST	
/loadevaluation	Returns the entire list of results - instead of just the top 10 like the previous endpoint - from one of the tables of a previously run evaluation.	GET	type, startDate, endDate
/querygraph	Returns the number of times a string was searched per day.	GET	string, startDate, endDate
/clicksranks	Returns the number of times each results list's position was clicked when a string was searched.	GET	string, startDate, endDate
/querysummary	Returns the information displayed on 'Query summary' when performing a query's analysis.	GET	string, startDate, endDate
/pagesperrank	For a specific results list's position, returns the pages that were accessed by clicking that position.	GET	page, mysql_id, string, startDate, endDate
/unsuccessfulsessions	Returns the number of times a string was searched and the user did not access any result.	GET	string, startDate, endDate
/pagegraph	Returns the number of times a page was accessed through a search's results list.	GET	tp_item, fk_item, startDate, endDate
/pagesummary	Returns the information displayed on "Page summary" when performing a page's analysis.	GET	tp_item, fk_item, startDate, endDate
/pagesrank	Returns the number of times each results list's position was clicked when accessing a page.	GET	tp_item, fk_item, startDate, endDate
/stringsperrank	For a specific results list's position, returns the strings that were searched when the page was accessed by clicking that position.	GET	page, mysql_id, tp_item, fk_item, startDate, endDate
/hotqueries	Returns details of the strings searched a minimum number of times on the current day to be displayed on "Trending".	GET	startDate, minimum
/hotpages	Returns details of the pages accessed a minimum number of times on the current day to be displayed on "Trending".	GET	startDate, minimum

Table 3.3: Application back-end's endpoints.

a collection of database connections that can be reused when required. Only if every existing connection is already being used, a new one is created, which is a costly process and should be avoided when possible. Regarding request routing, Express.js' "Router" object is responsible for redirecting client requests to a particular endpoint. We created 14 different endpoints: one to receive zerozero.pt's messages containing search information, and the remaining to handle the application front-end's requests. Each routing method specifies a callback function that is executed when the application receives a request to that route. We programmed these functions to perform the following steps: retrieve information from the request's parameters or body for GET and POST requests, respectively, build the database's queries that may be required to execute, obtain a database's connection from the connection pool, run the previously constructed interrogations, and reply to the request's sender. All the existing endpoints are described in Table 3.3. We also created two auxiliary objects on the directory "utils": "query.js", that possesses several methods, each containing a query's template, that return an interrogation built with the information retrieved from the request; and "utils.js", that includes some practical functions that help data processing on the callback functions.

3.6 Front-end

Finally, the front-end uses React.js, an open-source JavaScript library used for building user interfaces. Using React.js allowed us to mix HTML code with JavaScript, using the JSX syntax,

which works as a visual aid when working with UI inside the JavaScript code. Also, React.js is component-based, with several reusable components comprising each application's view. This approach allowed us to utilize various components from different sources to create all application's views. Most of the used components and icons were extracted from Material-UI, a popular React UI framework. However, the charts are from Chart.js, a free open-source JavaScript library for data visualization. The application comprises four primary views, each accessible through a tab on the Sidebar:

- **Trending**: consists of two Sortable Tables, one for queries and one for pages, with several statistics that reveal how the popularity of the trending queries and pages evolved during the previous week.
- **Global analysis**: returns the most searched queries, the queries that lead to users leaving the results page without clicking a result more often, and the most accessed pages during a previously selected time interval. Only the top ten results of each analysis are shown.
- **Query analysis**: for previously selected query and time interval, displays a chart with the number of searches per day, a summary containing different statistics identical to the ones on "Trending", a table with the number of clicks on each position of the results list, and the pages accessed by clicking on each of these positions.
- **Page analysis**: for previously selected page and time interval, displays a chart with the number of accesses per day, a summary containing different statistics identical to the ones on "Trending", a table with the number of clicks on each position of the results list to reach the analyzed page, and the executed queries that resulted in users reaching the analyzed page accessed by clicking on each of these positions.

An additional view, "Full results", can only be reached through "Global analysis" tables and displays the complete list of results from one of the conducted analyses. Finally, all application tables' rows possess shortcuts, as a hyperlink, that redirect users to "Query analysis" if the row contains a query or "Page analysis" if the row contains a page, and automatically runs the analysis. For example, clicking a row's id on the "Popular Queries" table existing on "Global analysis" opens a new browser window on "Query Analysis" and automatically runs the analysis for that row's query and the analyzed time interval. The application's sitemap can be seen in Figure 3.3

3.6.1 Main Components

Although several components are used by the application to build the different views of the platform's front-end, we chose to highlight the following five:

- **Tables**: there are three types of table throughout the platform. Regular tables consist of simple tables with a small number of rows. Tables with Pagination are used to show information expected to occupy a larger number of rows. These tables divide data by pages and allow

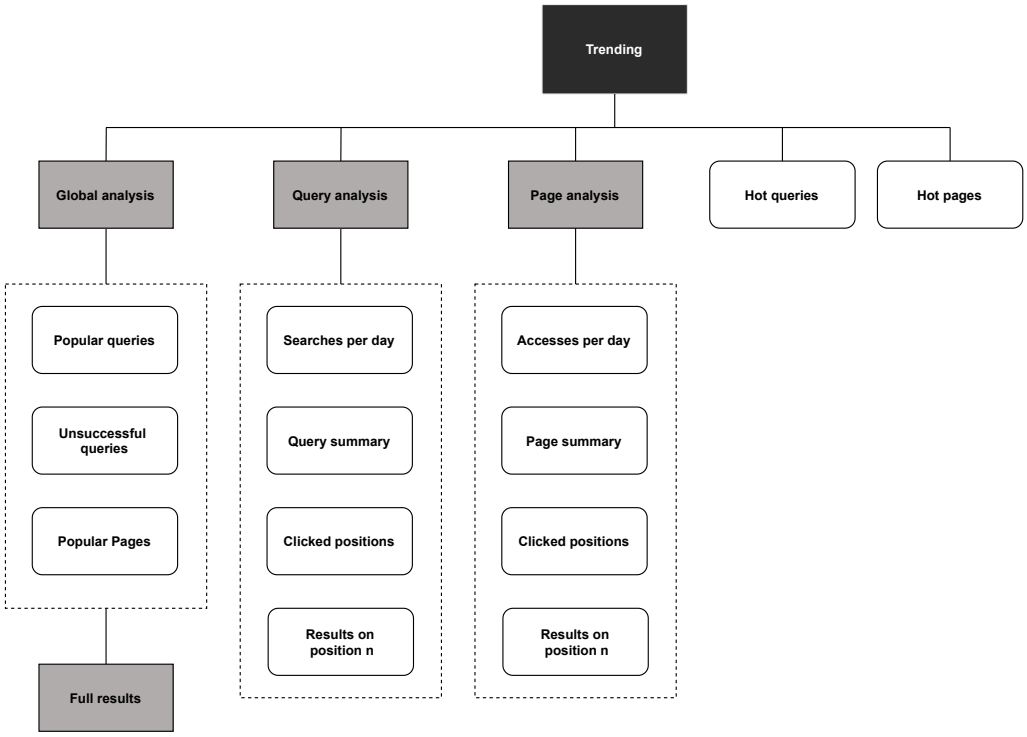


Figure 3.3: Platform's sitemap.

users to customize the number of rows showed by page. Finally, Sortable Tables are used in "Trending" and possess the same features as tables with Pagination. Since these tables contain many columns, users can sort the table's rows according to a column, allowing them to more easily analyze its content.

- **Time Pickers**: this component exists in every input of the application and allows users to select the time interval of the analysis to be conducted. Calendars are similar to Time Pickers but do not allow for hours and minutes customization. They are also included on the charts present on both "Query analysis" and "Page analysis" to customize the time interval of the displayed content.

- **Charts**: this application uses two different chart types. Line Charts are used to show the number of searches/accesses per day on both "Query analysis" and "Page analysis", while Bar Charts complement the data displayed by the tables on "Global analysis" by providing a visual representation of this information.

- **Sidebar**: it is always displayed on the left side of every view of the platform. It contains four different tabs, each one representing a major view of the application's view that will be described on the next sections.

- **Lists**: used to display the summary on both "Query Analysis" and "Page Analysis". Lists organize their contents by topics and contain multiple columns.

3.6.2 Trending

Figure 3.4 shows an instance of this view that is divided into two sections: "Hot Queries" and "Hot Pages". Each section contains a Sortable Table with a large amount of information and filters to customize the table's content. Both tables' columns are very similar and include:

- **Id**: the query or page being analyzed. Clicking on this column's values redirects the user to "Query analysis" or "Page analysis". Also, on "Hot Pages", each id has a zerozero.pt's icon that, when pressed, opens the analyzed page directly on the website.
- **Average rank**: includes the average position of the clicks, as well as a progress bar with value equal to the ratio of clicks on the first position expressed as a percentage.
- **Last 30 min**: number of query searches/page accesses on the last 30 minutes.
- **Last 30 min %**: percentage growth comparing the number of query searches/page accesses on the last 30 minutes to the previous half-hour.
- **Last 60 min**: number of query searches/page accesses on the last 60 minutes.
- **Last 60min %**: percentage growth comparing the number of query searches/page accesses on the last 60 minutes to the previous hour.
- **Today**: number of query searches/page accesses on the current day.
- **Yesterday**: number of query searches/page accesses on the previous day.
- **Last 24 hours %**: percentage growth comparing the number of query searches/page accesses on the current day to the previous day.
- **Last 7 days**: average number of query searches/page accesses on the last seven days.
- **Last 7 days %**: percentage growth comparing the number of query searches/page accesses on the current day to the average number of searches/accesses on the last seven days.
- **7 days ago**: number of query searches/page accesses seven days before.
- **7 days graph**: a Line Chart of the number of query searches/page accesses per day on the last seven days.
- **Insuccess rate**: only exists for "Hot Queries" and represents the ratio of search sessions that resulted in no further clicks expressed as a percentage. It was excluded from "Hot Pages" since a search session with page access will never be considered unsuccessful.

Additionally, each table has a filter to restrict the minimum number of query searches or accesses a query or page, respectively, should have on the current day to be included on the table. This filter has a default value of 10. Lastly, "Hot pages" also allows users to restrain the shown rows to only one entity type, such as players or teams.

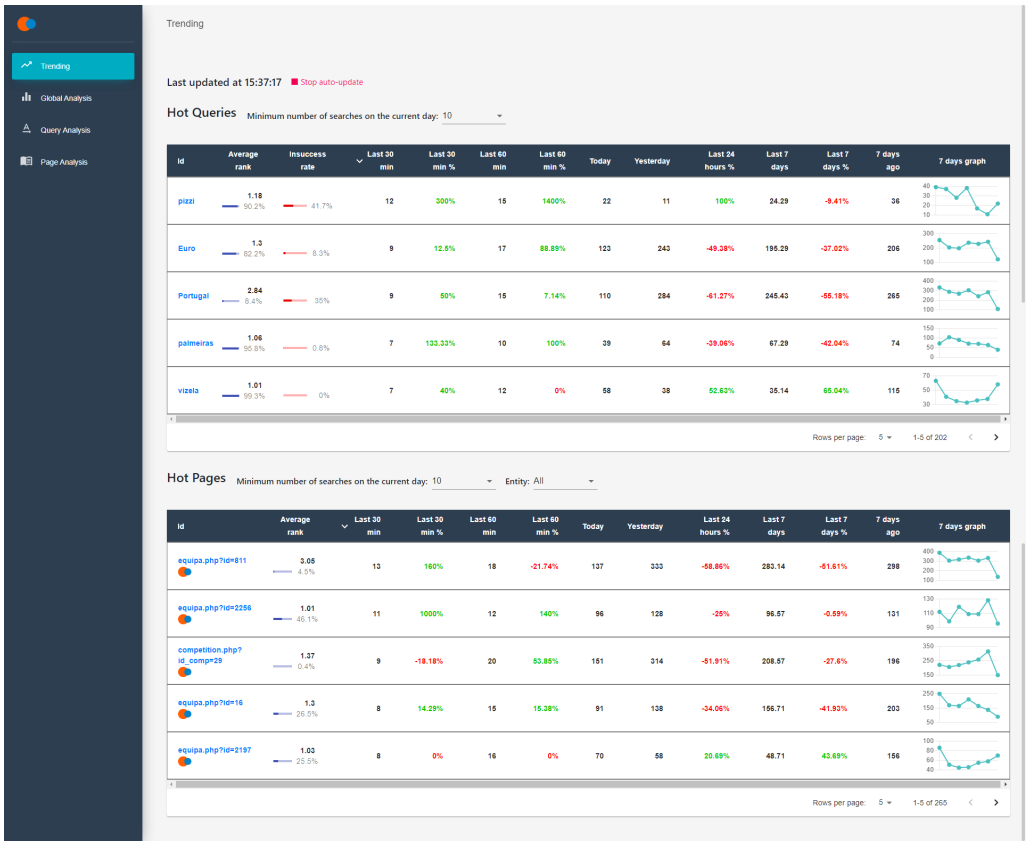


Figure 3.4: Platform's "Trending" view.

3.6.3 Global analysis

On this page, users can analyze searches issued on zerozero.pt on a defined time interval. On the input area, which is placed inside a lengthy rectangle on the top of the page, the user can customize the analysis' time interval using Time Pickers and submit it. Alternatively, one can press one of the buttons placed at the right of the Time Pickers to automatically run the examination for the last 30 minutes, 60 minutes, or 24 hours. Once the analysis is submitted, multiple requests are sent to the back-end. Once the front-end receives the replies, it displays Bar Charts for the ten most searched queries, the ten queries that lead to no further click more often, and one for the most accessed pages through the results list. The user also can instruct the platform to display tables containing the same data as the charts by clicking the toggle "Show tables". Through these tables, users can be redirected to:

- **"Full results"**, by clicking the icon with a similar label placed next to the title of each table.
- **"Query analysis"**, through the hyperlink on each row's query.
- **"Page analysis"**, by clicking the hyperlink on each row's id.

Furthermore, “Popular pages” table has a zerozero.pt’s icon at the end of each entry, that when pressed, opens the analyzed page on the website. Figure 3.5 shows an example of analysis conducted for the entire day of June 9, 2021.

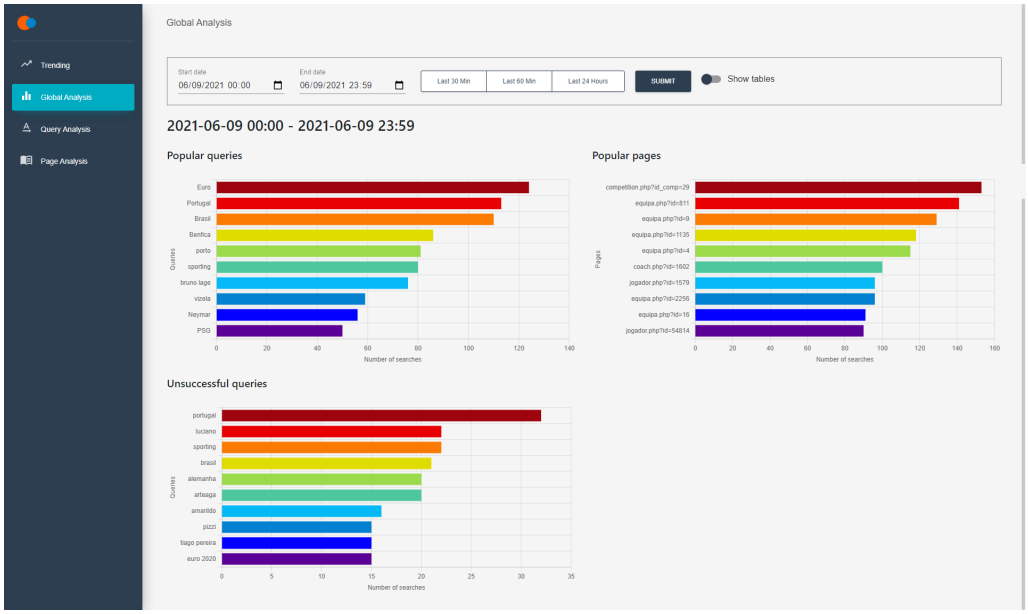


Figure 3.5: Platform’s “Global analysis” view.

3.6.4 Query analysis

The main goal of this feature is to provide statistics from a specific zerozero.pt’s query. In addition to the time interval customization tools mentioned in the previous section, “Query analysis” also possesses a text input for users to insert the string to analyze. Once users submit the analysis, the front-end sends six requests to the back-end:

1. to **/querygraph**. It includes the days in the chosen time interval. The response’s data is used by the platform to create a Line Chart on “Searches per day”, with days as the x-axis and number of searches as they-axis.

2. to **/querysummary**. Returns several statistics related to the chosen time interval and displays them on the “Query summary” column labelled “Selected period”.

3. to **/querysummary**. Returns several statistics related to the last 24 hours and displays them on the column “Query summary” column labelled “Last 24h”. This information is similar to the one shown in “Trending”.

4. to **/clicksrank**. Returns the positions clicked on the results list displayed for the analyzed query on zerozero.pt. This information is displayed on “Clicked positions” table. For ranks larger than 19, the number of clicks on all those positions is added and shown as “20+”.

5. to **/unsuccessfulsessions**. The returned information is displayed under "Clicked positions" table.

6. to **/pagesperrank**. Returns the pages accessed by clicking on the highest position displayed on "Clicked positions". This information is displayed on "Results on position n" table.

Also, users can reduce the time interval visible on "Searches per day" Line Chart by changing the Time Pickers under it, and choose the position to be analyzed on "Results on position n" by clicking the icon at the end of each entry on "Clicked positions". Figure 3.6 displays an example of an analysis conducted for the query "Portugal" between June 2, 2021 and June 9, 2021.

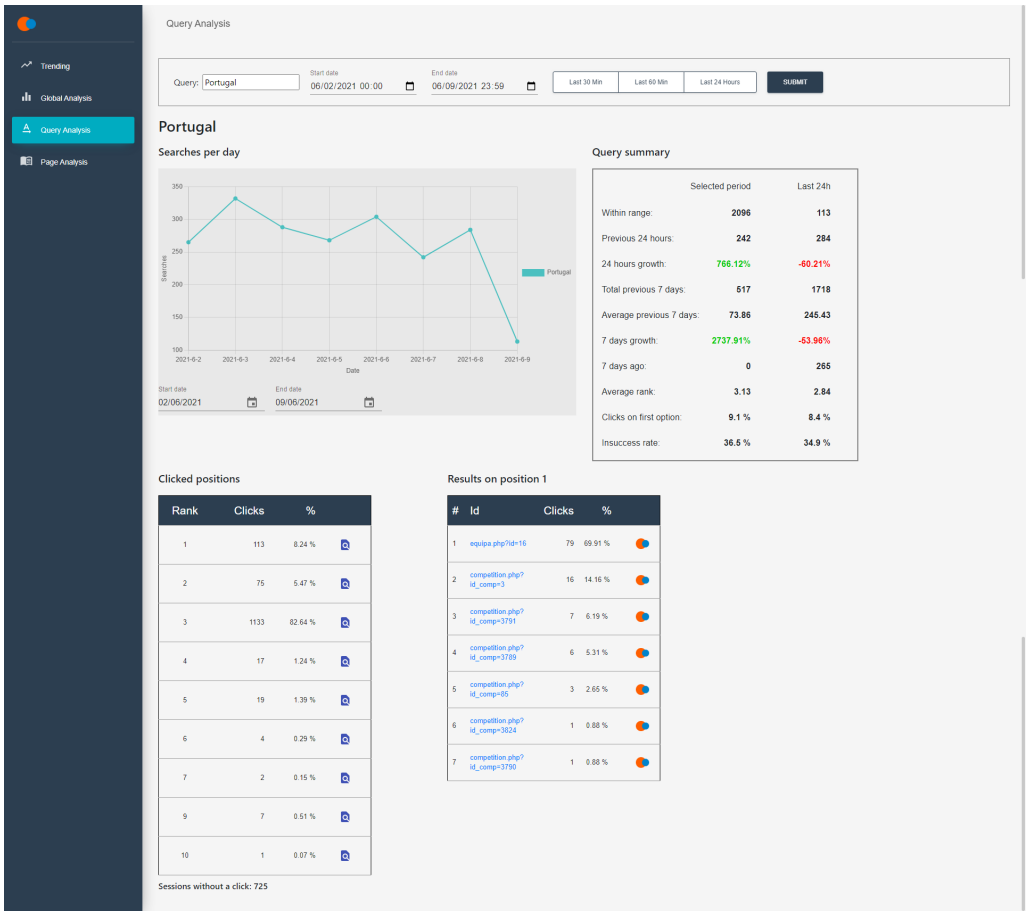


Figure 3.6: Platform's "Query analysis" view.

3.6.5 Page analysis

This tab allows users to run a similar analysis as "Query analysis", only for pages instead of queries. Both views' layouts are also highly similar, containing the same four sections, although with some slights changes. This is because we are interested in viewing statistics related to the search system's performance when handling that particular interrogation for queries. In contrast,

on “Page Analysis”, our main interest lies in understanding how difficult it is for users to find a specific page through zerozero.pt’s search system. Regarding the input area, in addition to the already mentioned time interval customization tools that exist in other views, “Page analysis” requires users to choose the entity type and identifier of the page to be analyzed. For example, “FC Porto” is a team (entity type) identified by 9. To retrieve the necessary information to display on “Clicks per day”, “Page summary”, and “Clicked positions”, the front-end sends similar requests to “/pagegraph”,“/pagesummary”, and “/pagesrank”, respectively, as those forwarded for the two first mentioned sections in the previous paragraph. Figure 3.7 illustrates an analysis carried out for the Portuguese national team’s page on zerozero.pt between June 2, 2021 and June 9, 2021. For the remaining section, “Results on position n”, the application displays the complementary information to the one showed by its counterpart on “Query analysis” - queries that resulted in the analyzed page being accessed versus pages accessed by executing the analyzed query. Finally, since a search session with a page access will always result in a click, there is no need to display information related to unsuccessful sessions.

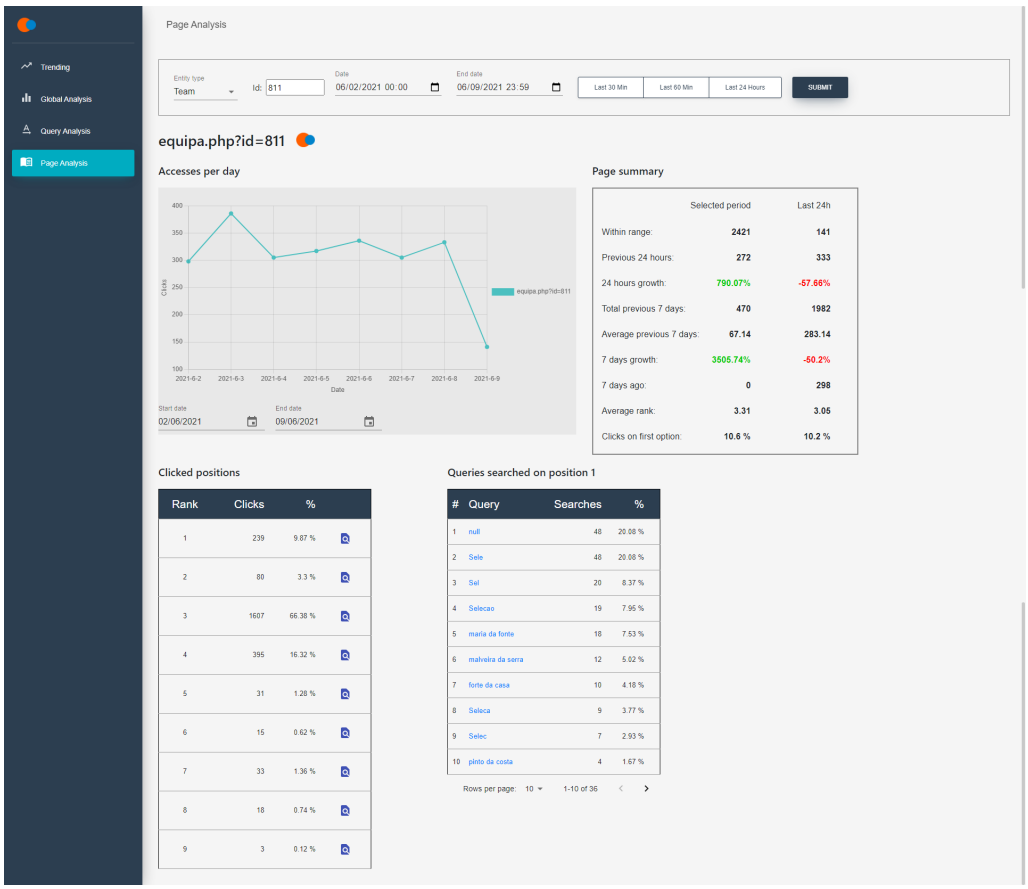


Figure 3.7: Platform’s “Page analysis” view.

3.6.6 Full results

The last feature we are highlighting is "Full results". This view has the unique particularity of not being accessible through the Sidebar. Instead, users can only access this view after running an analysis in "Global analysis". Each of the three sections on the results area of "Global analysis" has an icon next to the title labeled "Full results". Clicking this shortcut opens a new window containing a Table with Pagination and a Bar Chart that display the full results obtained from that analysis. Also, changing the number of displayed rows on the table will update the number of results on the chart, allowing the user to personalize the visual representation on this page. Figure 3.8 shows the full results of "Popular queries" from the analysis presented in Figure 3.5, ran for the period between June 2, 2021 and June 9, 2021.

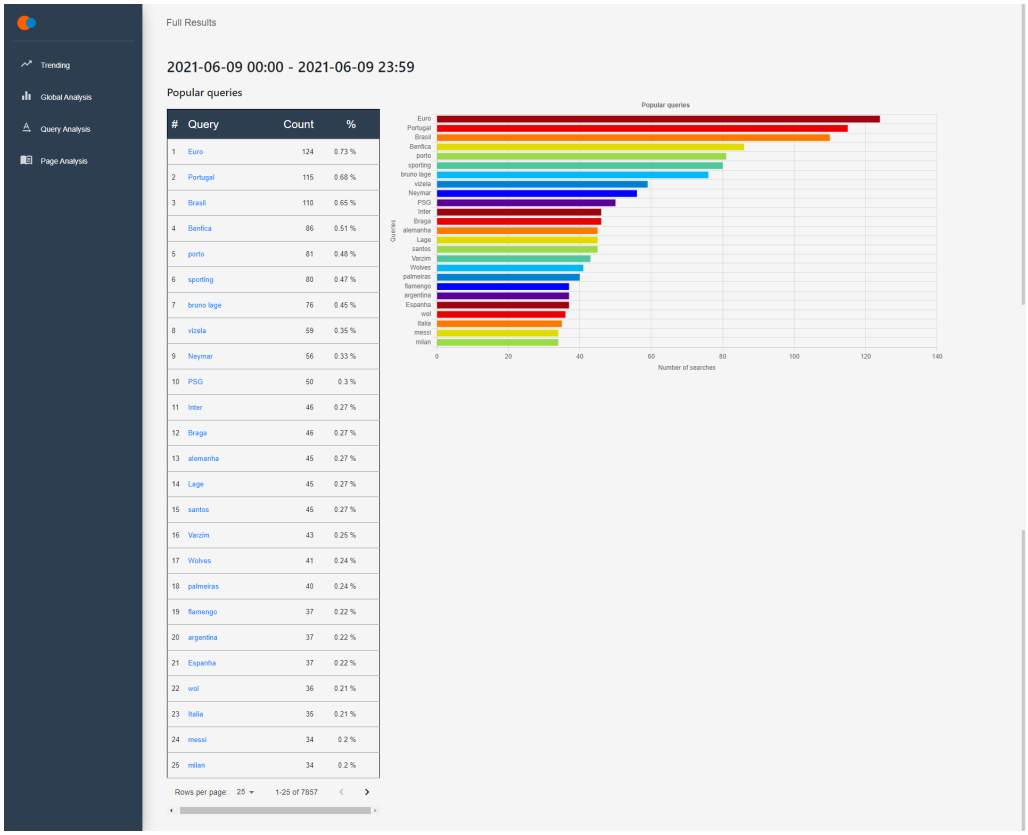


Figure 3.8: Platform's "Full results" view.

Chapter 4

Usability Testing

After developing a platform, it is essential to verify if a new user can easily understand how to navigate the interface and use the available features. This process is denominated usability testing. Usability testing is a methodical way of observing users trying out a product and gathering information regarding how easy or difficult the product is for them to use [38]. While there are multiple methods to evaluate a platform's usability, we will focus on observation and questionnaires in this section.

4.1 Questionnaires

Standardized usability questionnaires are designed to assess perceived usability, typically with a specific set of questions presented in a specified order using a specified format with particular rules for producing scores based on the answers of respondents [34]. They can be used as a supplement to other techniques or on their own. The second case is considered indirect usability methods since they do not analyze the user's direct interaction with the system.

4.1.1 System Usability Scale

System Usability Scale (SUS) [36] is a popular questionnaire released by John Brooke in 1986 comprised of ten questions, displayed on table 4.1, with five response options, that range from "Strongly disagree" to "Strongly agree". This tool calculates a score from zero to 100, which measures the overall usability of a system. However, this score should not be interpreted as a percentage since a score of 50 does not indicate that a system is average. In fact, the average score from a sample of 500 inquiries was 68, while a score of 50 placed the system around the 12^{th} percentile. A graph showing SUS's score distribution can be seen in Figure 4.1. A survey's score is calculated by

1. Subtracting one from the user's response for odd questions.

Questions
1. I think that I would like to use this system frequently.
2. I found the system unnecessarily complex.
3. I thought the system was easy to use.
4. I think that I would need the support of a technical person to be able to use this system.
5. I found the various functions in this system were well integrated.
6. I thought there was too much inconsistency in this system.
7. I would imagine that most people would learn to use this system very quickly.
8. I found the system very cumbersome to use.
9. I felt very confident using the system.
10. I needed to learn a lot of things before I could get going with this system.

Table 4.1: System Usability Scale's questions

2. Subtracting the user responses from five for even items p.e. if the user's answer is one, then we subtract one to five, resulting in four.

3. Adding all these scores.

4. These operations will result in a number between zero and 40. Then, we multiply it by 2.5 to obtain the final result from zero to 100.

The main benefits of using this survey are its reliability and validity. Reliability refers to how consistently users respond to the questions. SUS is better at detecting inconsistencies at smaller sample sizes than home-grown questionnaires and other commercially available ones. However, small sample sizes may still generate slightly imprecise scores, which is why it is recommended to compute a confidence interval to better understand the score's variability. Validity is how accurate are the measurements generated by the survey. SUS can effectively distinguish between unusable and usable systems and correlates highly with other questionnaire-based measurements of usability. The main limitation of this survey is that it is not diagnostic. That means that SUS will not identify a system's main problems, being up to researchers to review and perfect the interface [33].

4.1.2 Questionnaire for User Interaction Satisfaction

Questionnaire for User Interaction Satisfaction (QUIS) [37] was developed at the University of Maryland and designed to assess a user's subjective satisfaction with an application interface. The questions approach a wide variety of subjects, including overall reaction to the software, how intuitive the screens are, used terminology, and how difficult it is to learn how to use and system's capabilities, such as speed and reliability. Each question measures users' opinions on a 9-point categorical scale. While all SUS's answer options range from "Strongly disagree" to "Strongly agree", QUIS's responses assess different system characteristics like its difficulty, reliability, consistency, or clarity. Also, the questionnaire provides additional space at the end to allow users to make comments or suggestions regarding the interface. There are two available QUIS formats: a

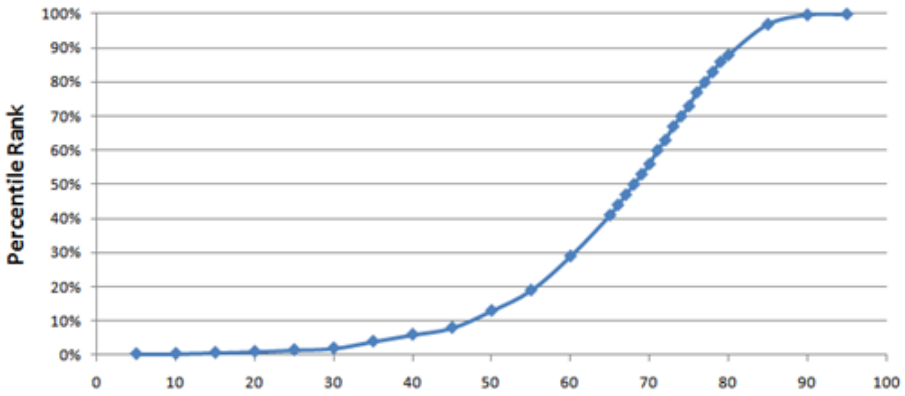


Figure 4.1: Percentile ranks associated with SUS scores.
Image from Sauro [33].

short form with 27 interrogations and a long form with 71 questions. Like SUS, QUIS has been proven both reliable and valid but does not generate a score that allows us to quickly understand an application's quality [42].

4.1.3 Software Usability Measurement Inventory

Software Usability Measurement Inventory (SUMI) [39] was developed by the Human Factors Research Group in 1993. It provides a viable and reliable method for comparing competing products and differing versions of the same product and providing diagnostic information for future developments. It is possible to either directly compare two products or compare each product against a standardization database. The questionnaire consists of 50 statements, to each of which the user may respond "Agree", "Disagree" or "Don't know" and generates an usability rating across efficiency, effect, helpfulness, control, and learnability, along with a global rating. One example of results obtained using SUMI can be seen in Figure 4.2. In this case, the system got an average global system but poor learnability and helpfulness ratings, showing that these areas should be improved in future versions of the software. Unfortunately, a significant limitation of this survey is the lack of accessible information related to the questions and how to calculate the scores.

4.1.4 Computer System Usability Questionnaire

Computer System Usability Questionnaire (CSUQ) [31] was developed by IBM in 1995 and is comprised of 19 questions, shown in Table 4.2. Similar to SUS, each question is a statement with a rating on a scale of "Strongly Disagree" to "Strongly Agree", although CSUQ uses a seven-point scale. CSUQ can be used to measure an application's informational quality, system usefulness, and interface quality. The first eight questions are related to the system's usefulness, the next seven statements access the information quality, and the final four evaluate the interface quality. Like all the previously described surveys, this one has also been proven to be reliable and viable. Still,

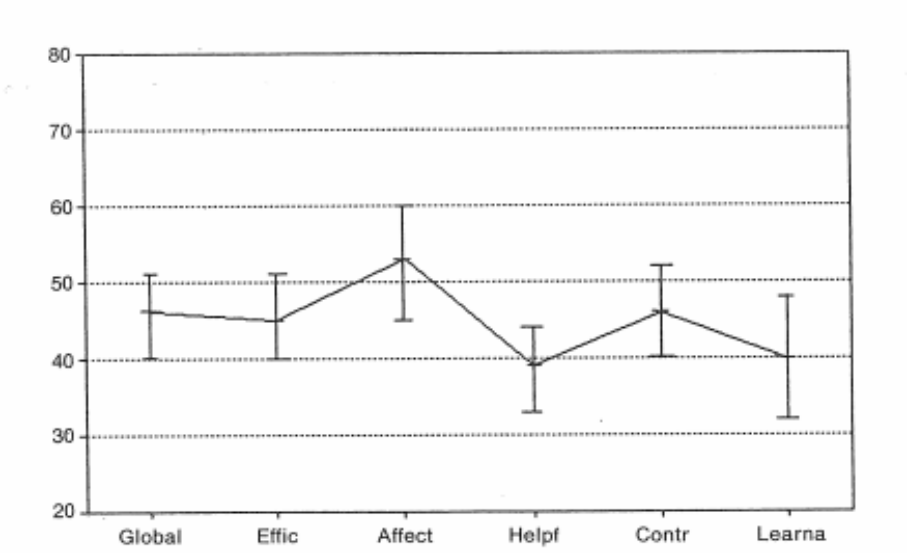


Figure 4.2: Example of result obtained using SUMI.
Image from reference 18.

there is a lack of available information that allows us to fully understand how good the obtained score is compared with the rating obtained by similar systems.

4.2 Observation

The information people provide during interviews is not always accurate or reliable. To better understand what people do, it is not enough to ask them; we have to observe them. Observation provides the most accurate information about people, their tasks, and their needs. It is a task that requires a lot of attention, effort and knowledge to analyze the user's behavior and answer our question regarding the system's usability [35]. In usability testing, the primary value of observation consists of paying attention to what the user does. Still, it is also relevant to listen to what a participant says while navigating through the interface. Some variations include asking questions during the testing or instructing the user to think out loud, either alone or by cooperatively performing tasks with another participant. The second variation usually produced better results since it represents a more natural setting [42].

Questions
1. Overall, I am satisfied with how easy it is to use this system.
2. It is simple to use this system.
3. I can effectively complete my work using this system.
4. I am able to complete my work quickly using this system.
5. I am able to efficiently complete my work using this system.
6. I feel comfortable using this system.
7. It was easy to learn to use this system.
8. I believe I became productive quickly using this system.
9. The system gives error messages that clearly tell me how to fix problems.
10. Whenever I make a mistake using the system, I recover easily and quickly.
11. The information (such as on-line help, on-screen messages and other documentation) provided with this system is clear.
12. It is easy to find the information I need.
13. The information provided with the system is easy to understand.
14. The information is effective in helping me complete my work.
15. The organization of information on the system screens is clear.
16. The interface of this system is pleasant.
17. I like using the interface of this system.
18. This system has all the functions and capabilities I expect it to have.
19. Overall, I am satisfied with this system.

Table 4.2: Computer System Usability Questionnaire's questions.

Chapter 5

Solution evaluation

Once the platform's development was finished, it was now required to evaluate the solution. In order to assess an application's quality, we decided that the best option was to build a survey that helps us value the platform's usability. The questionnaire sample consisted of zerozero.pt's collaborators that are expected to use the application on a day-to-day basis. Also, we analyzed the obtained results, drawing conclusions from them on how the platform could be improved in the future. We decided it was interesting to include people with different positions within the company to investigate how the responses varied from role to role since journalists will probably have more curiosity in understanding the search tendencies on the website. In contrast, computer scientists are interested in learning possible difficulties visitors are having while using zerozero.pt's search feature.

5.1 Survey

When deciding on how we wanted to structure our platform's validation, we chose to create a survey that assesses the application's general usability and validates some of our design decisions for the user interface. For this reason, we decided to build a questionnaire with Google Forms [27] and comprised of two main sections, each one accomplishing each of the previously mentioned goals. No explanation on how the platform worked was given to participants before they answered the survey. Moreover, participants responded to the questionnaires remotely. We chose this method over local surveys to attract more participants and allow them to learn the platform at their own pace. When participants open the survey, they must insert their role on zerozero.pt. This information allows us to find possible correlations between specific answers and roles. The participants then move into the first section, consisting of the ten questions from the System Usability Scale, described in Chapter 4. This tool, shown to be reliable and valid, yields a score from zero to 100 that can be used for a general evaluation of our application's quality. Plus, we can utilize that score to estimate how this application compares to other similar platforms. The second section is

comprised of more concrete questions related to the user interface. It starts with five questions, each of them evaluates to what extent each of the front-end's features highlighted in the previous chapter meet the user's information needs. Additionally, the survey requests participants to give their opinion regarding the available inputs, the tables' redirects, and the Line Charts' filters. Every question has five response options, ranging from one to five. Finally, the questionnaire also has an open-end question that allows participants to submit any additional comments or suggestions to improve the application. All the questions included in the survey are illustrated in Appendix C.

5.2 Results

Participants with four different roles answered this survey: four content managers, three journalists, one designer, and one computer scientist, totaling nine answers. The results obtained for the first section, which consisted of SUS' questions, are displayed in Table 5.1. The results are organized by roles, with "Content manager" being shortened to CM and "Computer scientist" being shortened to CS, along with a column labeled "General" that encompasses all roles. As we can see at the bottom of the table, the platform's rating using SUS's scoring method was 45.56, which measures its usability as mediocre. While these results cannot be considered satisfactory, we have to highlight the small number of answers, which makes it difficult to determine the reliability of these results. Plus, after talking to some participants to better understand the underwhelming score, they noted that they had trouble understanding the platform's content and navigation. However, once someone explained how to utilize the application more in detail, they found it more interesting and enjoyable than before. For this reason, we can conclude that the platform should be more intuitive for new users, but once users understand how to handle it, they enjoy using it and find its contents helpful.

Regarding the survey's second section, which comprises more concrete questions of the different existing features, the results are shown in table 5.2. Starting with the first five questions related to the platform's main features, they all obtained similar results, except "Trending" that got a slightly higher rating than the other four. While we can consider the results slightly positive, since the scale used ranges from one to five, which means that the average for each answer is three, it is not by a significant margin, which indicates that all features have room for improvement. Finally, after analyzing the questionnaire's last four questions, that address some design decisions of the user interface, we can conclude that participants value the ability to select the time interval of the platform's several analysis, and not just the dates, and also find the filters used to control the displayed information on the charts included in "Query analysis" and "Page analysis" practical to use. However, participants did not consider the tables' redirects intuitive, which further validates the application's lack of intuitiveness. This problem could be addressed in future iterations of the application by adding more textual hints explaining what each feature offers and possibly update some sections' titles, mainly on "Query analysis" and "Page analysis", to more instinctive headlines. Finally, no one answered the last question, an open-end question that participants could use to add additional comments or suggestions about possible improvements to the platform.

Questions	**General**	**CM**	**Journalist**	**Designer**	**CS**
1. I think that I would like to use this system frequently.	3.11	3.25	3.00	2.00	4.00
2. I found the system unnecessarily complex.	3.33	3.00	3.64	4.00	3.00
3. I thought the system was easy to use.	3.00	3.00	3.33	2.00	3.00
4. I think that I would need the support of a technical person to be able to use this system.	2.55	2.50	2.33	4.00	2.00
5. I found the various functions in this system were well integrated.	3.33	3.25	3.33	3.00	4.00
6. I thought there was too much inconsistency in this system.	2.78	2.5	3.33	3.00	2.00
7. I would imagine that most people would learn to use this system very quickly.	3.44	3.25	3.33	3.00	5.00
8. I found the system very cumbersome to use.	2.67	3	2.33	4.00	1.00
9. I felt very confident using the system.	3	2.75	3.33	2.00	4.00
10. I needed to learn a lot of things before I could get going with this system.	2.78	2.5	2.67	4.00	3.00
Score (0-100)	45.56	45	45	67.5	27.5

Table 5.1: Questionnaire's first section results

Questions	**General**	**CM**	**Journalist**	**Designer**	**CS**
1. To what extent does the information on 'Trending' meet your needs?	3.44	3.75	3.33	2.00	4.00
2. To what extent does the information on 'Global analysis' meet your needs?	3.11	3.25	3.00	2.00	4.00
3. To what extent does the information on 'Query analysis' meet your needs?	3.11	3.25	3.00	2.00	4.00
4. To what extent does the information on 'Page analysis' meet your needs?	3.11	3.00	3.33	2.00	4.00
5. To what extent does the information on 'Full Results' meet your needs?	3.11	3.00	3.33	2.00	4.00
6. To what extent do the available filters meet your expectations?	3.33	3.50	3.33	3.00	3.00
7. How useful is it to have hours and minutes on the filters and not just date?	3.56	3.75	4.00	3.00	2.00
8. How intuitive are the redirects on each table?	2.89	2.75	3.00	3.00	3.00
9. How practical are the charts' filters on 'Query Analysis' and 'Page Analysis'?	3.67	3.75	3.67	3.00	4.00

Table 5.2: Questionnaire's second section results

Chapter 6

Conclusions

At the beginning of this project, we established two main goals: developing an analysis platform to display information related to search sessions on zerozero.pt's, and assessing the application's usability by conducting a survey directed to zerozero.pt's employees that should use it on a daily basis. Since the latter is not possible without the former, we started by identifying the main functionalities to include in the platform. Once we enter the application, it would be interesting to see the current search trends on zerozero.pt immediately, both in terms of executed queries and accessed pages through the results list displayed in response to queries. Also, we wanted that page to display various statistics of how those trends evolved in the last seven days. In addition to the trends, the application should have views to analyze in-depth specific queries and pages. Some of the data shown on a query's analysis being: the number of times zerozero.pt's visitors executed that query, what positions they clicked on the results list to access a result and how many of these sessions ended without any further click, what pages users are reaching when inserting that interrogation, and a summary containing statistics related to the query's popularity over the last seven days. Similar data should be displayed for the page's analysis, although with some differences: there is no need to display the number of sessions that result in no clicks since that will never happen when a page is accessed, and we are interested in learning what queries users are executing to reach the analyzed page. Finally, we want to see, for a selected time interval, what are the most executed queries, accessed pages, and the queries that result in no further click using zerozero.pt's search system. Regarding the second goal, we wanted to read users' opinions regarding the overall platform's usability and some more particular design decisions we made for the user interface. For that effect, we elaborated a survey comprised of two main sections, each one addressing one of the previously mentioned goals. The first section comprised the ten questions from System Usability Scale, a popular questionnaire that estimates an application's usability by giving it a score that ranges from zero to 100. Using an already established rating method enabled us to compare our system to other similar ones, which provided a better idea of the platform's quality. The second section consisted of five questions that assessed to what level each of the

application's main features met the participants' information needs and four questions related to some platform's components. Finally, participants can provide some additional comments or suggestions by answering an open-ended question placed at the end of the questionnaire.

We believe that we managed to achieve the project's primary goals. The constructed platform implements all the established requirements. First, on "Trending", users can view how visitors are currently using zerozero.pt's search system, namely the most executed queries and accessed pages, along with various related statistics and filters to customize the displayed information. Plus, the application allows users to conduct a detailed analysis for each query or page, including data on how the popularity of what is being analyzed evolved and complicated it was for visitors to find the information they were looking for. Additionally, for a selected time interval, users can view how zerozero.pt's visitors mainly use the website's search feature and the main difficulties found by them. While, at first, the platform only shows the top ten queries or pages that meet these criteria, users can access the complete list of results through a shortcut placed next to each section. Regarding the questionnaire results, these were not satisfactory. The platform obtained a 45.56 rating using SUS's scoring method, which is mediocre. However, after talking to some of the participants to try to understand the underwhelming results, they noted that their negative answers were due to difficulties understanding how the application worked at first. Once someone explained to them how to navigate through the application and how to interpret its information, their opinion towards the platform improved. These results point to a significant lack of intuitiveness. Moreover, the second section obtained positives results, with all questions related to the application's main features obtained identical scores, excepting "Trending" 's score, which was slightly higher. Still, worth mentioning that participants had struggled with understanding the tables' redirects, which further points to the system's lack of intuitiveness.

6.1 Future work

Regarding future improvements of the developed platform, it can be carried on in various directions. Even though the application is not using this information, it is possible to retrieve what domain a visitor was in when he performed a search. One can retrieve this information through each request's field "language" and include it as an additional filter for each analysis. This could be particularly interesting since if the same query is executed in different domains, it will return two different results lists. For example, when searching "Atletico" on the Portuguese domain, Atletico CP is returned on the second position. However, executing the same query on the Spanish domain does not return that same team on the first ten results. Another factor of the search sessions that would be interesting to explore would be zerozero.pt's version the visitor was using during the search session. The Portuguese domain provides users with different specialized website versions dedicated to a specific sport, including soccer, volleyball, or handball. Similar to the visitor's domain, including a filter that allowed platform users to analyze searches executed on a selected website version could also be very attractive. However, unlike the previously mentioned suggestion, it is impossible to retrieve this knowledge from the current zerozero.pt messages' structure,

so that would require an additional attribute containing that information. Finally, we already mentioned the need to address the platform's lack of intuitiveness. We can address this problem in future iterations by adding more textual hints explaining what each feature offers and possibly update some sections' titles, mainly on "Query analysis" and "Page analysis", to more instinctive headlines.

Appendix A

Zerozero.pt's API request

```
{
    "id": "10",
    "time": "2021-05-15 18:32:03",
    "date": "2021-05-13",
    "method": "",
    "page": "player.pp?id=155284&search=1",
    "referer": "https://www.zerozero.pt/",
    "IP": "194.65.93.237",
    "sess_id": "0uu63nh7veidmqnvrli04rc4q5",
    "username": "",
    "agent": "Mozilla/5.0 (Linux; Android 10; SM-G960F Build/QP1A.19
       0711.020; wv) AppleWebKit/537.36 (KHTML, like Gecko) Version/
       4.0 Chrome/88.0.4324.93 Mobile Safari/537.36 [FB_IAB/FB4A;
       FBAV/302.0.0.45.119;]",
    "language": "pt",
    "bot": "0",
    "page_number": 1,
    "mysql_id": "5",
    "cache": "0",
    "server_ip": "fontaine",
    "geo_location": "pt",
    "search_string": "porto",
    "tp_item": "3",
    "fk_item": "9"
}
```

Listing A.1: Example of a zerozero.pt's request

Appendix B

User stories

B.1 Trending

- **US01** - As a user, I want to have access to queries' statistics from the last seven days, so that I can easily see how each query's popularity evolved during this period.
- **US02** - As a user, I want to have access to pages' statistics from the last seven days, so that I can easily see how page's popularity evolved during this period.
- **US03** - As a user, I want to be able to sort by any attribute, so that I can better analyze what are the best and worst entries for each attribute.
- **US04** - As a user, I want to filter by a minimum number of searches, so that I can control what information is displayed.
- **US05** - As a user, I want to filter the pages by entity type, so that I can only view records from a specific entity.
- **US06** - As a user, I want the information to be updated periodically, so that I have access to refreshed statistics without interacting with the platform.
- **US07** - As a user, I want to be able to disable the periodically updates, so that I can analyze the data without worrying about them refreshing.

B.2 Global analysis

- **US11** - As a user, I want to view the ten most searched queries during a determined time interval, so that I can learn what users have been searching the most.
- **US12** - As a user, I want to view the ten queries that result in unsuccessful sessions more often during a determined time interval, so that I can help users find what they desire.

- **US13** - As a user, I want to view the ten most accessed pages during a determined time interval, so that I can understand what type of content users have been accessing the most through the search system.
- **US14** - As a user, I want to be able to toggle visibility on and off, so that I can decide how the information is being displayed.

B.3 Query analysis

- **US21** - As a user, I want to view a chart containing how many times a query was executed per day, so that I can learn how its popularity evolve through time.
- **US22** - As a user, I want to have access to a summary, both from a determined time interval and the last 24 hours, so that I can quickly visualize several statistics from the analyzed query.
- **US23** - As a user, I want to know the number of times each position of the results list of a query was clicked, so that I can analyze if the most relevant results are being shown on the first ranks.
- **US24** - As a user, I want learn how many sessions resulted in users not clicking a result, so that I understand if users are having difficulties when executing the analyzed query.
- **US25** - As a user, I want to see the pages reached by clicking on a selected position, so that I can visualize what results are being placed on that rank.

B.4 Page analysis

- **US31** - As a user, I want to view a chart containing how many times a page was reached by using the search system, so that I can learn how its popularity evolve through time.
- **US32** - As a user, I want to have access to a summary, both from a determined time interval and the last 24 hours, so that I can quickly visualize several statistics from the analyzed page.
- **US33** - As a user, I want to have access to know the number of times the analyzed page was accessed by clicking on each position of the results list, so that I can analyze if the page is being placed on the top ranks.
- **US34** - As a user, I want to see the executed queries that lead to the analyzed page to be reached by clicking on a selected position, so that I can visualize what queries are leading users to reach this page.

B.5 Full results

- **US41** - As a user, I want to view the complete list of results from a previously performed analyses, so that I have access to more data than just the top ten.
- **US42** - As a user, I want to be able to change the number of results being shown, so that I can control the what is being displayed.

B.6 General

- **US51** - As a user, I want to be redirected to "Query analysis" from any row of a table containing queries, so that I can automatically view a query's statistics during a time interval.
- **US52** - As a user, I want to be redirected to "Page analysis" from any row of a table containing pages, so that I can automatically view a page's statistics during a time interval.
- **US53** - As a user, I want to access a page's zerozero.pt's link from any table containing pages, so that I can easily be redirect to the page being analyzed.
- **US54** - As a user, I want to have access to buttons that automatically run an analysis for the last 30 minutes, 60 minutes and 24 hours, so that I can quickly view its results.
- **US55** - As a user, I want the application to function with dynamic links, so that I am able to directly run an analysis by adding parameters to an URL.

Appendix C

Survey

1. Which is your role at zerozero.pt? (Journalist, content manager, computer scientist ...)
Open-end answer

C.1 First section

2. I think that I would like to use this system frequently.
Strongly disagree 1 ○ ○ ○ ○ ○ 5 Strongly agree

3. I found the system unnecessarily complex.
Strongly disagree 1 ○ ○ ○ ○ ○ 5 Strongly agree

4. I thought the system was easy to use.
Strongly disagree 1 ○ ○ ○ ○ ○ 5 Strongly agree

5. I think that I would need the support of a technical person to be able to use this system.
Strongly disagree 1 ○ ○ ○ ○ ○ 5 Strongly agree

6. I found the various functions in this system were well integrated.
Strongly disagree 1 ○ ○ ○ ○ ○ 5 Strongly agree

7. I thought there was too much inconsistency in this system.
Strongly disagree 1 ○ ○ ○ ○ ○ 5 Strongly agree

8. I would imagine that most people would learn to use this system very quickly.
Strongly disagree 1 ○ ○ ○ ○ ○ 5 Strongly agree

9. I found the system very cumbersome to use.
Strongly disagree 1 ○ ○ ○ ○ ○ 5 Strongly agree

10. I felt very confident using the system.
Strongly disagree 1 ○ ○ ○ ○ ○ 5 Strongly agree

11. I needed to learn a lot of things before I could get going with this system.
Strongly disagree 1 ○ ○ ○ ○ ○ 5 Strongly agree

C.2 Second section

12. To what extent does the information on 'Trending' meet your needs?
Not at all 1 ○ ○ ○ ○ ○ 5 Extremely well

13. To what extent does the information on 'Global analysis' meet your needs?
Not at all 1 ○ ○ ○ ○ ○ 5 Extremely well

14. To what extent does the information on 'Query analysis' meet your needs?
Not at all 1 ○ ○ ○ ○ ○ 5 Extremely well

15. To what extent does the information on 'Page analysis' meet your needs?
Not at all 1 ○ ○ ○ ○ ○ 5 Extremely well

16. To what extent does the information on 'Full Results' meet your needs?
Not at all 1 ○ ○ ○ ○ ○ 5 Extremely well

17. To what extent do the available filters meet your expectations?
Not at all 1 ○ ○ ○ ○ ○ 5 Extremely well

18. How useful is it to have hours and minutes on the filters and not just date?
Not at all 1 ○ ○ ○ ○ ○ 5 Extremely

19. How intuitive are the redirects on each table?
Not at all 1 ○ ○ ○ ○ ○ 5 Extremely

20. How practical are the charts' filters on 'Query Analysis' and 'Page Analysis'?
Not at all 1 ○ ○ ○ ○ ○ 5 Extremely

21. Do you have any additional comments or suggestions about possible improvements to the platform?
Open-end answer

References

[1] In *Trustworthy Online Controlled Experiments: A Practical Guide to A/B Testing, publisher=Cambridge University Press, pages=166–170, title=Choice of the Randomization Unit in Online Controlled Experiment, author=Shaojie Deng, Roger Longbotham, T. Walker and Y. Xu, year=2011*.

[2] Jason Burby, Angie Brown and WAA Standards Committee. *Web Analytics Definitions*. Web Analytics Association, 2007.

[3] Ahmed Hassan and Yang Song and Li-wei He. A Task Level Metric for Measuring Web Search Satisfaction and Its Application on Improving Relevance Estimation. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, CIKM '11, page 125–134, New York, NY, USA, 2011. Association for Computing Machinery.

[4] Ahmed Hassan, Xiaolin Shi, Nick Craswell and Bill Ramsey. Beyond clicks: Query reformulation as a predictor of search satisfaction. In *Proceedings of the 22nd ACM International Conference on Information Knowledge Management*, pages 2019–2028, 10 2013.

[5] Ahrefs. Ahrefs - SEO Tools & Resources To Grow Your Search Traffic. Available at https://ahrefs.com/, 2021. [Online; accessed May 2021].

[6] Alex Deng, Ron Kohavi, Ya Xu and Toby Walker. Improving the Sensitivity of Online Controlled Experiments by Utilizing Pre-Experiment Data. In *Proceedings of the Sixth ACM International Conference on Web Search and Data Mining*, Feb. 2013.

[7] Amit Singhal. Modern Information Retrieval: A Brief Overview. *IEEE Data Eng. Bull.*, 24 (4):35–43, 2001.

[8] Andrea Gazzarini, Alessandro Benedetti and Eric Pugh. Rated Ranking Evaluator. Available at https://github.com/SeaseLtd/rated-ranking-evaluator/wiki, 2018. [Online; accessed January 2021].

[9] Andrea Gazzarini, Alessandro Benedetti and Eric Pugh. Domain Model. Available at https://github.com/SeaseLtd/rated-ranking-evaluator/wiki/Domain%20Model, 2018. [Online; accessed February 2021].

[10] Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, USA, 2008.

[11] Corey Frankosky. Google Search Console: Complete Guide For 2020. Available at https://surfsideppc.com/google-search-console/, Aug 2020. [Online; accessed January 2021].

[12] Crazy Egg. Crazy Egg Website Optimization | Heatmaps & A/B Testing. Available at https://www.crazyegg.com/, 2021. [Online; accessed May 2021].

[13] Cyril Cleverdon. *The Cranfield Tests on Index Language Devices*, page 47–59. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1997. ISBN 1558604545.

[14] Dan Sommerfield, Ron Kohavi, Roger Longbotham and Randal M. Henne. Controlled Experiments on the Web: Survey and Practical Guide. *Data Mining and Knowledge Discovery*, 18(1):140–181, Feb. 2009.

[15] Danielle L. Booth and B. Jansen. A Review of Methodologies for Analyzing Websites. 2010.

[16] Diane Kelly. *Methods for Evaluating Interactive Information Retrieval Systems with Users*, volume 3. Now Publishers Inc., 2009. doi: 10.1561/1500000012.

[17] Elasticsearch. Elasticsearch reference. Available at https://www.elastic.co/guide/en/elasticsearch/reference/current/index.html, 2020. [Online; accessed February 2021].

[18] Eric T. Peterson. *Web Analytics Demystified*. Celio Group Media and CafePress, 2004.

[19] Express.js. Express - Node.js web application framework. Available at https://expressjs.com/, 2017. [Online; accessed May 2021].

[20] Filip Radlinski and Nick Craswell. Comparing the sensitivity of information retrieval metrics. In *Proceedings of the 33rd international ACM SIGIR conference on Research and development in information retrieval*, pages 667–674, jan 2010.

[21] Filip Radlinski and Nick Craswell. Optimized Interleaving for Online Retrieval Evaluation. WSDM '13, page 245–254, New York, NY, USA, 2013. Association for Computing Machinery. ISBN 9781450318693.

[22] Google. Google Search Console. Available at https://search.google.com/search-console/welcome, 2015. [Online; accessed January 2021].

[23] Google. Analytics. Available at https://analytics.google.com/, 2021. [Online; accessed May 2021].

[24] Google. Cohort analysis. Available at https://support.google.com/analytics/answer/9670133?hl=en#zippy=%2Cin-this-article, 2021. [Online; accessed May 2021].

[25] Google. Lifetime Value. Available at https://support.google.com/analytics/answer/6182550?hl=en#zippy=%2Cin-this-article, 2021. [Online; accessed May 2021].

[26] Google. About Demographics and Interests. Available at https://support.google.com/analytics/answer/2799357?hl=en#zippy=%2Cin-this-article, 2021. [Online; accessed May 2021].

[27] Google. Google Forms: Free Online Surveys for Personal Use. Available at https://www.google.com/forms/about/, 2021. [Online; accessed May 2021].

[28] Google. How Site Search metrics are calculated - Analytics Help. Available at https://support.google.com/analytics/answer/1032321?hl=en&ref_topic=1031951#zippy=%2Cin-this-article, 2021. [Online; accessed May 2021].

[29] Google Blog. What's happening on your site right now? Available at https://analytics.googleblog.com/2011/09/whats-happening-on-your-site-right-now.html, 2011. [Online; accessed May 2021].

[30] Hotjar. Hotjar: Website Heatmaps & Behavior Analytics Tools. Available at https://www.hotjar.com/, 2021. [Online; accessed May 2021].

[31] James R. Lewis. IBM Computer Usability Satisfaction Questionnaires: Psychometric Evaluation and Instructions for Use. *Int. J. Hum.-Comput. Interact.*, 7(1):57–78, Jan. 1995.

[32] Jane Li, Huffman, Scott and Akihito Tokuda. Good Abandonment in Mobile and PC Internet Search. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, page 43–50, New York, NY, USA, 2009.

[33] Jeff Sauro. Measuring Usability With The System Usability Scale (SUS). Available at https://www.userfocus.co.uk/articles/measuring-usability-with-the-SUS.html, May 2016. [Online; accessed June 2021].

[34] Jeff Sauro and James R. Lewis. Chapter 8 - Standardized usability questionnaires. In J. Sauro and J. R. Lewis, editors, *Quantifying the User Experience (Second Edition)*, pages 185–248. Morgan Kaufmann, Boston, second edition, 2016.

[35] Jim Ross. The Role of Observation in User Research. Available at https://www.uxmatters.com/mt/archives/2018/09/the-role-of-observation-in-user-research.php, Sep 2018. [Online; accessed February 2021].

[36] John Brooke. *SUS – a quick and dirty usability scale*, pages 189–194. 01 1996.

[37] John P. Chin, Virginia A. Diehl and Kent L. Norman. Development of an Instrument Measuring User Satisfaction of the Human-Computer Interface. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '88, page 213–218, New York, NY, USA, 1988. Association for Computing Machinery.

[38] Joseph S. Dumas and Janice C. Redish. *A Practical Guide to Usability Testing*. Intellect Books, GBR, 1st edition, 1999.

[39] Jurek Kirakowski and Mary Corbett. SUMI: the Software Usability Measurement Inventory. *British Journal of Educational Technology*, 24(3):210–212, 1993.

[40] Justin Boyan, Dayne Freitag and Thorsten Joachims. A Machine Learning Architecture for Optimizing Web Search Engines. *AAAI Workshop on Internet Based Information Systems*, 02 1970.

[41] Katja Hofmann and Lihong Li and Filip Radlinski. *Online Evaluation for Information Retrieval*, volume 10. Now Publishers Inc., 2016.

[42] Layla Hasan. The usefulness of user testing methods in identifying problems on university websites. *JISTEM - Journal of Information Systems and Technology Management*, 11:229 – 256, 08 2014.

[43] Matomo. Matomo Analytics - The Google Analytics alternative that protects your data. Available at https://matomo.org/, 2021. [Online; accessed May 2021].

[44] MySQL. MySQL Documentation. Available at https://dev.mysql.com/doc/, May 2021. [Online; accessed May 2021].

[45] Olivier Chapelle, Thorsten Joachims, Filip Radlinski and Yisong Yue. Large-Scale Validation and Analysis of Interleaved Search Evaluation. *ACM Transactions on Information Systems - TOIS*, 30:1–41, Feb. 2012.

[46] React.js. React – A JavaScript library for building user interfaces. Available at https://reactjs.org/, 2021. [Online; accessed May 2021].

[47] Ron Kohavi and Roger Longbotham. *Online Controlled Experiments and A/B Testing*, pages 922–929. 01 2017.

[48] Ron Kohavi and Stefan Thomke. A/B Testing: How to Get It Right. Available at https://hbr.org/2017/09/the-surprising-power-of-online-experiments, Sep 2020. [Online; accessed January 2021].

[49] Semrush. Features | SEMrush. Available at https://www.semrush.com/features/, 2021. [Online; accessed May 2021].

[50] Solr. Apache Solr Features. Available at https://lucene.apache.org/solr/features.html, 2019. [Online; accessed January 2021].

[51] Thorsten Joachims. Optimizing Search Engines using Clickthrough Data. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 08 2002.

[52] Trang Nguyen and Phuong-Nam Cao. Web analytics tools and benefits for entrepreneurs. 2017.

[53] Vennevar Bush. As We May Think. *Atl. Mon.*, 176(1):101–108, 1945.

[54] Web Technology Surveys. Usage statistics of traffic analysis tools for websites. Available at https://w3techs.com/technologies/overview/traffic_analysis, 2021. [Online; accessed May 2021].

[55] William R. Shadish, Thomas D. Cook and Donald T. Campbell. *Experimental and quasi-experimental designs for generalized causal inference*. Houghton Mifflin Company, 2002.