MeteoTecnico Responsive

Francisco Aresta Branco de Matos Miranda
Instituto Superior Técnico
franciscoabmmiranda@tecnico.ulisboa.pt

Prof. João Manuel Brisson Lopes
Instituto Superior Técnico
brisson@tecnico.ulisboa.pt

ABSTRACT

With the constant evolution with technology, we observe that an increasing number of people are starting to have access to technological devices. More specifically, in regards to internet access, more people are accessing to the internet and they are using more and more different types of devices to do so. It becomes important to guarantee that websites are prepared to be used in such a different pool of devices. Responsive Web Design aims to tackle this problem by focusing some of the design process on creating methods of adapting an website layout according to the sizes and orientation of these devices. In this project, we focused on MeteoTecnico, which is a weather forecast website, and tried to find a solution to the the responsive problems we encountered when analysing this website and comparing it to other similar ones. We developed this solution with User Centered Design in order to have users validating our decisions throughout the development. We will present the different stages of the development and discuss the results of the tests of our solution.

Author Keywords

Weather, Meteorology, Web Design, Responsive Web Design, User-centered Design.

INTRODUCTION

Nowadays, the access to technology is gradually increasing. According to the Internet World Stats, by June of 2019, almost 59% of the population of the world, which corresponds to about 4,536 million people, have access to the internet [1]. This trend had been predicted by the Canadian philosopher Herbert Marshall McLuhan [1][2]. McLuhan stated that the world was changing towards what he called a "global village" which made reference to the integration of the planet in terms of connection between different places. He predicted that, with the evolution of technology, the world would become "connected" in such a way that we could think of it as if the whole planet was just a small village where events that happened in one part of the world could be felt in other parts of it. In fact, this became true with the emergence of the Internet. Through the Internet, people can communicate in real-time from opposite places in the planet and almost everyone, who has access to the internet, can obtain the same amount of information. This was, in part, made possible by the increasing number of available internet connected technological devices.

[1]Internet World Stats, Internet Growth Statistics, https://www.internetworldstats.com/ emarketing.htm, last access 2020/12/28

The share of households with a computer at home worldwide has been increasing since the computer market started growing [2]. Additionally, and more predominantly nowadays, the global smartphone penetration (percentage of global population with a smartphone) as increased significantly in the past years, increasing by 8% (from 33.5% to 41.5%) between 2016 and 2019 [3]. Theses changes regarding the access of people to technological devices affected not only the accessibility of internet to people, but also the way people interact with the internet, since it is now available in a wide range of devices.

Given the wider spread of internet access, the World Wide Web (WWW) becomes a more competitive environment where several sources of information and different products compete between them to gain notability. This encourages organizations to improve the overall quality of their websites in order to make them appealing to a broader range of people. With this urge, some new concepts emerged to help tackling the problems that arose. Most of these new concepts are related to the design of websites in terms of the interaction with users. Two of these concepts are Interaction Web Design and Human-Computer Interaction which focus on designing interactive products to support people in their daily and working lives. These new design concepts became one of the main concerns of software engineers [3] when developing websites and applications intended for public use. The concept of Interaction Web Design was an adaptation to the already existing concept of Interaction Design made to accommodate to the shift in technological advances. This new concept focus primarily in the principles of the design of websites. A good human-computer interaction design is necessary to make websites more effective, efficient and even enjoyable and rewarding. With better designs comes the realization and understanding of the potential of these tools when it comes to improving and facilitating the life of people.

Responsive Web Design (RWD) is also one of these new concepts. RWD refers to the need of having each web page resized and reshaped to adapt to different devices that may have different screen sizes, resolutions and orientations. As stated before, the global smartphone penetration is gradually increasing and with it comes the need of having websites adapt to all these devices. For that reason RWD is becoming one of

[2]Statista, Share of households with a computer at home worldwide from 2005 to 2019 https://www.statista.com/statistics/748551/worldwide-households-with-computer/, last access 2020/12/28

[3]Statista, Global smartphone penetration rate as share of population from 2016 to 2020, https://www.statista.com/statistics/203734/global-smartphone-penetration-per-capita-since-2005/, last access 2020/12/28

the most predominant web design concerns when developing new websites. More details on RWD will be discussed in the following sections.

Problem definition

MeteoTecnico [4] is a forecasting channel run by Instituto Superior Técnico. This website was firstly created to give information regarding the conditions of wind and luminosity to predict the production of renewable energies (wind and solar respectively). Therefore, this website was designed to be used by people with experience in the field of meteorology that used the available information for professional purposes.

However, recent study [5,6] showed that about 80% of people check some sort of weather forecast at least once a day. In this study it was also found that normally about 70% of the people check the weather online, either via phone apps or web searches. For this reason, MeteoTecnico became a website that reaches a wider set of users, from weather experts to common users that just want to check the forecast for the day. Thus, and given that a wider range of users translates into a wider range of devices being used, it becomes necessary to guarantee that the information in the website is displayed across different devices, independently of the actual device used.

Work Objectives

Taking into account the problem defined previously, the aim of this project was to create a new solution for MeteoTecnico that would incorporate new User Interface (UI) design approaches to the current solution. This UI design changes had a predominant focus in RWD in order to make MeteoTecnico more adaptable to different devices.

One of the requirements to achieve this solution was to use User Centered Design (UCD). This approach gives emphasis to the role of the user in creating a solution since it helps to accomplish a simple and intuitive solution that allows different users to have a fast way of checking the weather forecast. For that reason, we firstly analyzed different weather forecast websites to understand the most common practices in terms of design and even in regards to the most predominant meteorological information. We then used this knowledge to compare and make a critical analysis of the previous MeteoTecnico website. This way we could identify specific problems that needed to be solved or adapted in the new solution. Further ahead we will showcase the problems identified in the UI design with special emphasis on the responsive behaviour of the website.

After identifying the problems, we needed to the define the requirements for the solution, which are briefly described below:

- Maintain the main architectural structure of the website (website organization) untouched. This meant that the solution should maintain the same structure in terms of existing web pages.

- Properly implement RWD so that the information available in each web page was kept through different devices and different screen sizes. To achieve a proper RWD, we needed to ensure that any user was able to view the same information regardless of the device used.

- Guarantee that the website worked properly not only in different devices but also in different browsers. For this reason, we needed to ensure that the implemented solution would be available in the most used browsers available (Google Chrome, Safari, Firefox, Edge, Opera) [7].

To ensure the effectiveness of the solution and to develop the project according to the UCD user testing was conducted throughout some stages of the implementation. We firstly created a low-fidelity prototype that was tested by some users to assure that the main design features of the proposed solution were the most effective. Only then could we start implementing the high-fidelity prototype that would lead to the final solution. During the implementation and before submitting the solution to user testing, it was pivotal that we conducted functionality tests to ensure that the website worked properly and had the right RWD behaviour in different browsers and devices. A final evaluation with the users was needed to find design problems in the solution that would be fixed later.

Finally, we made a critical analysis of the final solution with regards to the previously defined requirements and also the results of the evaluation conducted with the users. We identified the key points of success of the solution and also the problems to be tackled in future iterations, which will be later discussed in this thesis.

Contributions of the project

As stated in the previous section, the aim of this thesis was to implement RWD to an already existing website. This meant that we firstly needed to do a research about UCD and RWD in order to understand the key concepts to implement in our solution. Knowing this, the goal of this project was to apply already existing knowledge and come up with an approach to solve a very specific problem. However, there was some areas of the project that needed to be done using slightly different approaches to some problems encountered during the development. This was specially true regarding the user testing phase and the period in which this testing needed to be done. This project was developed during the 2020 and 2021 covid-19 pandemic, which meant that physical encounters were limited and, for some time, prohibited. This lead to slightly different approaches when regarding the user testing. These different approaches will be discussed later on this paper.

RELATED WORK

[4] MeteoTecnico, https://meteo.tecnico.ulisboa.pt/, last access 2020/12/28

[5] https://fivethirtyeight.com/features/weather-forecast-news-app-habits/

[6] https://medium.com/information-expositions-f2019/who-checks-the-weather-dc7d37971e34

[7] Stat counter, Browser Market Share Worldwide https://gs.statcounter.com/browser-market-share, last access 2020/12/28

Key Concepts

User Interface

The concept User Interface (UI) design [3] consists in the development of web related technology (websites and applications for example), focusing on the user interaction. When using the UI design process we give special emphasis on the essential details of the program since that is also the main focus of the users. This simplifies the process of the development which increases the design speed. Furthermore, there is a direct relationship between relevancy of the items and their visibility on the web pages [4]. The information that is not that essential should be given a less important focus on the web pages by placing it away from the top center of the page or even by hiding that information. It is also important to ensure some consistency on the design of a solution[5].

User Centered Design

User centered design is a process for developing UI [6][7]. The major advantage of this design approach is the continuous involvement of users throughout all the stages of the design process of a web application. In fact, with this approach, the developers are able to continuously analyze the psychological and social nuances of the users that affect the interaction between them and the technology. Moreover, by having the participation of users, evaluating the solution during the different stages of the development becomes easier. This approach follows the development stages described below[8][9][10]:

1. Identification of the users. When developing any kind of technological solution it is pivotal to know your users. The users need to be defined in terms of their expectations for the solution and their level of expertise. This way, the developers can better define the needs of the users that will translate to software requirements.

2. Definition of the solution's requirements. The developers need to properly specify the requirements of the solution to be able to design the different prototypes prior to the final product. These requirements should be defined with regard to the solution they want to achieve and also the needs of the users.

3. Low-fidelity (lo-fi) prototype design. After defining the requirements, the developers are able to make a first iteration of the desired solution. In order to test some of the design ideas with the users, firstly the developers need to create a lo-fi prototype (normally designed on paper) and submit it to a user evaluation [4]. This way, users can evaluate the intended design approaches without a lot of development effort.

4. Continuous iteration. The developers should do several iterations of the design, with more detailed prototypes. This way, and through user testing, the main design issues can be found at the early stages of the development and can be improved for the following prototype iterations.

5. Implementation. Once most of the issues are found, the solution can be implemented at a higher level of fidelity. This will result in an high-fidelity (hi-fi) prototype that will later be tested by the users.

6. Maintenance. Finally, every product developed according to UCD needs to have continuous evaluation and improvements. In fact, with the continuous evolution of technology, it is crucial that the product of the development is submitted to constant analysis in order to prevent the technological stagnation of the solution.

Responsive Design

Responsive Web Design (RWD) refers to the design concerns needed when developing websites that are intended to be viewed in the whole range of web devices [11][12]. In fact, a good RWD ensures that a website works properly regardless of the type of device and the size of the screen. This was not a concern in the early web development period, since most users would access the web in similar desktop computers. For this reason, there were no major differences between the design of a website for different devices in that time. However, nowadays the developers need to be concerned with all the different available devices used since they can be different in screen resolutions, orientations and views [14][15].

One of the main concerns of RWD is the content consistency across different devices. In fact, it pivotal that websites maintain their features and displayed information regardless of the size of the device. Another concern is related to the adaptation of a web page to a smaller screen. When adapting a web page to be viewed in a smartphone, for instance, it is crucial that the components remain readable and usable. One of the problems that arises when reducing the screen size is called the *fat finger syndrome*[8]. This problem occurs when a component, such as a button, becomes so small during the size adaptation that that specific button becomes hard to use, not only because clicking on it is difficult, but also because it is placed next to other clickable components that are triggered when users try to click on the small button.

To fix some of the issues presented previously, there are three common techniques to ensure the design of a responsive website [13]:

- Fluid grid layouts (Relative-based grid). The layout of the website changes according to the screen size. The different components of the web page change their organization and create different layouts when changing the screen size.

- Flexible images and media. The components can also be resized according to the screen size. This way, the content of the page is not altered, only the design of some of the components.

- Media queries and screen resolution. With this, developers can define different behaviours for different components for specific screen resolutions.

[8] UX 24/7, Mobile Usability Testing: Fat Finger Syndrome, https://ux247.com/mobile-usability-testing-fat-finger-syndrome/

As discussed before, nowadays it is of extreme relevance to implement proper responsive behavior while developing websites. Some of the main advantages of a good Responsive Web Design are the following:

- Being able to maintain a single web page that adapts according to different devices. This way, developers only need to provide support to a single web page instead of having different web pages configurations for each device.

- User experience is simpler since there is no need to zoom in and out when searching for some specific information. The main information should be available immediately and the rest of the information should be either accessible by scrolling or by opening menus.

- Provide large links and buttons in smaller screens to avoid the *fat finger syndrome* described before.

- Maintenance simplicity, since their is only one implementation of each web page that performs adaptations to all the different screen sizes.

Meteorologic Websites

In order to be able to do a comparison with the current state of MeteoTecnico, we had to analyse other weather forecast websites to observe common practices used in them. Firstly, we did an analysis of the overall design of each of those websites, taking into account the definition of the key concepts mentioned in the previous section. Then, we assessed the techniques used to achieve a better responsive design and understand which information was given predominance over the other when we reduced the screen size. In order to test and analyze each of these websites we used two different devices to compare the different designs in different screen sizes. This approach allowed us to understand the practices used to achieve responsive design. This was done by comparing the content of the website in each different screen size, seeing which information would persist and be given a predominant role when going from a bigger screen to a smaller one. Furthermore, we analyzed the techniques used in practical design adaptations that allowed the websites to have the same behaviour regardless of the screen size.

To simulate a desktop environment (bigger screen) we used a MacBook Pro with a 13-inch monitor. This environment enabled the resize of the browser window which allowed us to observe the real-time adaptations made to the website when adapting to a smaller screen size. Additionally, we tested these websites in a smartphone environment (smaller screen) using an iPhone SE with a 4-inch screen. In each one of these screens the analysis was done using the default zoom level of each web page.

For a better understanding of the most common practices we conducted this research in six different websites since it allowed us to identify frequent design patterns used in the websites. We decided to choose the websites according to their ranking in google search results. This way we had a specific criteria that ensured that our analysis was based on some of the most visited websites. Additionally, we intended to review websites that represented both the common practices in Portugal as well as the common practices in international websites. For that reason, we chose three international websites and three Portuguese ones.

During the analysis of each website, we tried to review the two main web pages of each one. This separation was made because some of those websites had a main page where they displayed the weather forecast for Portugal as a whole and then different pages for every region of Portugal. This way we ensured that we analyzed different methods of data display.

This analysis allowed us to understand the different ways these websites display the same information.

Previous version of MeteoTecnico

After analysing other weather forecast websites, we did a review of each page of MeteoTecnico in order to find usability and RWD problems.

The main problems regarding this website were the following: navigation through the website; cluttered information; and omitted information when reducing the screen size. The navigation problem is related firstly to the fact that the header was not intuitive since it had a couple of dropdown menus which made it difficult to have an understanding of the overall structure of the website. Moreover, the website provided two very similar pages (forecast by regions and forecast by coordinates), where the only difference between the two are the maps. There was also a problem regarding cluttered information which happened especially on the tables of the forecast by region and forecast by coordinates pages. The amount of days and amount of information displayed by day made those tables hard to read and to find the information needed. Finally, there were several problems regarding missing information in smaller screen sizes. In fact, the RWD was not well implemented since some pages removed information previously displayed, as for example the forecast by regions page which lost the map and did not display information about the temperature in the table. Additionally, when information was not lost and the layout was not rearranged, some pages, such as the history of observations page, became cluttered and therefore hard to use.

Meteorologic Concepts

We used our previous review of weather forecast websites to gather data regarding the approaches used when displaying meteorological information. Firstly we identified the weather factors that were more predominant in those websites and compared the relevance given to each of these factors in terms of their displacement in the web page. Secondly, we analyzed the way in which these factors were being displayed, regarding the units and icons used to convey the information.

We found that the most relevant weather factors being displayed in these websites were temperature and cloud cover. The temperature would usually be divided in maximum and minimum temperatures for a specific day and was normally displayed in Celsius. As for the cloud cover, it was normally displayed in icon form, which usually displayed the precipitation forecast as well. The other weather factor that was commonly displayed was the wind information. It would also be displayed most of the times as an icon where it showcased the direction of the wind and the intensity according to the Beaufort scale [9] which is used to relate the wind speed to the observed conditions at sea.

CONTEXTUAL ANALYSIS AND REQUIREMENTS

In order to achieve and implement a suitable solution for the problems we identified with MeteoTecnico in the previous chapter, firstly we needed to understand the context in which the solution ought to be built. This means that we needed to analyze the overall scenario of the current and of the intended solution for MeteoTecnico, which led to an assessment of the different types of users for which the solution should be targeted to.

Furthermore, to achieve a solution that contemplated Responsive Web Design, we needed to define the type of devices and environments in which the solution would be deployed. This translated into the identification of all the different devices and all the essential requirements so that a solution that would incorporate the responsive behaviour intended for this application could be developed.

Finally, it was crucial to identify specific requirements for the overall solution and for each of the pages of the website. These requirements were set up by the comparative analysis made in the previous chapter between MeteoTecnico and the other similar websites.

Context

As referred before, weather forecast websites and apps are increasingly growing in popularity due to the widespread of internet access and technological devices. Therefore, it is crucial to properly identify and characterize the users that visit websites like this ones, and consequently MeteoTecnico. Additionally, we needed to understand the environments (the devices in this case) in which the users normally have access to the website.

Users

MeteoTecnico is a service that supplies really precise and high quality weather forecasts. For that reason, MeteoTecnico provides this service to other companies and organizations. Additionally, it is monitored by meteorologic experts that help in the development and improvement of the service provided. Given this, we can conclude that a lot of the users that visit the website are either weather experts or users that use the website on a professional level. However, and as mentioned in the work objectives, one of the objectives of the new solution was to have a more simplistic and intuitive version of MeteoTecnico in order to also be suitable for less experienced users. These users are the common users that check the weather forecast once a day, which only need a fast and simple way of checking the specific information they seek.

Given these two types of users (expert and common), we needed to come up with a solution that enable the common users to have a simple and quick interaction with the website but, at the same time, we needed to ensure that the more complex and detailed information would still be available for the experienced users who use the website for professional purposes.

[9] `https://en.wikipedia.org/wiki/Beaufort_scale`

Devices

In recent times, there has been considerable advancements regarding technology, in particular regarding technological devices. Concepts such as Internet of Things (IoT) have become more and more present in the of life of people. Not only that, but the most common internet driven devices, like cellphones and computers, have also evolved and are ever so more prominent nowadays.

Requirements

After making a critical analysis of the problems with the current state of MeteoTecnico regarding UI and RWD, and after analysing the context and target of the solution, we will now discuss the requirements that were defined for our solution. We will present the requirements regarding the design of the solution and also the technical requirements that will allow our solution to tackle the previously identified problems.

UI Requirements

The UI requirements are focused mainly on the design of the desired solution. We will discuss the intended structure for the whole website and also some design features that are pivotal to a functional solution.

Regarding the overall structure of the website, there are two main requirements that need to be taken into account: maintaining the website architectural layout and include all previous web pages. Since MeteoTecnico is an website with almost twenty years and with very frequent users, it is pivotal that the overall architecture of the website and of each web page is not changed massively.

There were four more requirements regarding the overall design of the website. maintain the coordinates map in the forecast and locations pages; maintain the satellite maps page; add a language changer; add a dark mode theme to the website.

Technical Requirements

The technical requirements are related to the functionality of the desired solution. There were four main technical requirements.

- Implement RWD behaviour.
- Ensure minimum behaviour even when failing the connection to the data API.
- Ensure safety of the data API and Database.
- Ensure that the solution worked properly in the main browsers: [10]: Google Chrome, Safari, Firefox, Edge, Opera.

[10] Stat counter, Browser Market Share Worldwide https://gs.statcounter.com/browser-market-share, last access 2020/12/28

ARCHITECTURE OF THE SOLUTION

In this section we will discuss the overall architecture and structure of our project. We will also showcase and analyse the available tools to develop our project.

Structure

We can see the overall structure of the proposed solution in figure 1. The solution consists in a website that will display data requested from a weather API. A server will support the website by requesting to the API the information the website requires. The requests made by the server to the API will be done by HTTPS requests from the website to the server, according to the needs of the user. This data will be sent by the API as a JSON [11] file which consist in a data-interchange format. The server will also serve as a security measure for malevolent requests that could be made to the API.

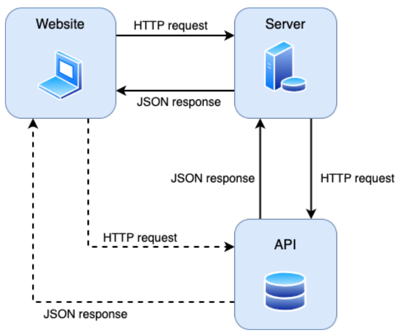

Figure 1: Project architecture

As said in the requirements section, we also wanted to have some persistent data in our server so that we could ensure the behavior of the main pages of the website even if the API was failing to respond. For that reason, we created a local DB in our server that stored the 5 day information for the 18 districts of Portugal. This DB would be updated every day with requests to the API. This way we could ensure that the forecast page and some location pages worked properly even if the API was off for a few hours of the day. Additionally, we would also reduce the calls to the API and subsequent overload of it since the forecast page could work properly without these requests since all the information was already stored in the hosting server.

Tools

In order to build our website we first needed to analyze some of the possible tools develop our solution. For that reason, we did a small review over some of the most used front end frameworks, some of the most used maps APIs and also regarding additional libraries necessary to develop the intended solution.

For the development of the website we used React JS [12]. We complemented the React chore with several other JS libraries such as Ant Design [13] as a UI toolkit and Font Awesome [14] for the icons. We also used Leaflet [15] to help display the maps. Finally, for the styling of the website, we used Sass [16]. For our server we used the Django framework [17]

[11]https://www.json.org/json-en.html
[12]https://reactjs.org/
[13]https://ant.design/
[14]https://fontawesome.com/

SOLUTION

In this chapter we will present the process of the development of the solution. We will discuss the several stages of the design process and also give a complete description of each step of the implementation of the solution

In order to come up with a solution to this very complex problem, we needed to follow the steps of the UCD discussed earlier. Following these steps was pivotal to allow us to develop an website in the most efficient and effective way. For that reason, after defining the context properly and identifying all the requirements of this project, we then proceeded to the development of the lo-fidelity prototype which is the first stage of the design process. Prototyping is not only necessary to define the overall design of a web application but it is also crucial to allow testing of the user experience (UX) of a proposed solution [17]. By designing a prototype at an early stage of the development we are able to identify immediately issues regarding the UI and UX of our solution. In the design process, it is really important to find the most relevant problems and issues of a solution as early as possible.

We firstly designed a low-fidelity (lo-fi) prototype which are used to showcase the design and layout of an application and are normally of rapid and cheap development. These usually consist of hand drawn papers that portray the screens and flow of the application.

Lo-fi prototype

In this section, we will discuss the methods used during the design, development and testing of the lo-fi prototypes.

In order to develop a lo-fi prototype, we needed to design not only all the screens of the desired web application, but also some of the interactions (buttons, graphs and maps) of the solution in order to test some possible UX issues. Additionally, and given the relevance of RWD in our work, we also needed to design each screen in a desktop environment and in a smartphone environment in order to see the differences in layout in different screen sizes.

To design each page, we firstly defined the the information to be displayed in each page and we analysed the importance we wanted to give to the different pieces of information. This way we were able to define how the layout was built since we wanted the most important information to be displayed as soon as the user entered each webpage.

The figure 2 showcases some of the prototypes made.

User Testing

After deciding on a design based on the requirements established, we needed to test our lo-fi prototype with users to see the major flaws in the final design.

[15]https://leafletjs.com/
[16]https://sass-lang.com/
[17]https://www.djangoproject.com/

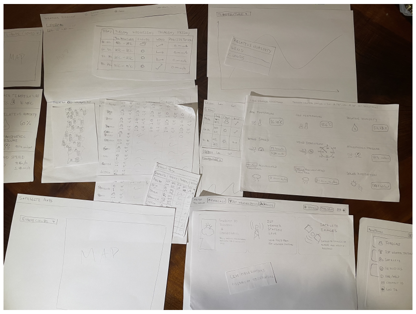

Figure 2: Lo-fi prototypes

Lo-fi prototype testing is usual done in a laboratory, or similar, with five to twelve people[4][15][6] where the users interact one by one with the lo-fi prototype. The testing is conducted by the designer who creates tasks that the users need to perform using the prototype. The designer simulates the behavior of the prototype by adding new screens or showing menus when the user interacts with different parts of the interface. The aim of these tests was to find design flaws that further ahead in the development could lead to a confusing and non intuitive interface for the users.

Due to the 2020 covid-19 pandemic, gathering with people was not recommended and even prohibited at some point. For that reason, we were not able to do these tests in person and needed to have a different approach to the making of theses tests. We wanted to make the testing experience as similar as possible to the known ones since we wanted our results to be as trustful as possible. Therefore, we decided to do the tests via skype call, where we would have a camera streaming a video of a table were the lo-fi prototype would be placed. We would then proceed to describe each task to the user, and the user would narrate each action he intended to do in order to complete the given task. As the users specified each intended action, we would simulate the behavior of the interface by changing the layout of the prototype, as we would do if we were next to the user that was testing our proposed solution. We will discuss further ahead the results and issues encountered both in our prototype as well as in our testing methods.

As said before, in order to test our lo-fi prototype we decided to create a few tasks that would allow each user to interact with all the parts of our website. The tasks were supposed to be simple and easily completed by the users. We took note of the tasks the the users had the most issues completing. We gave users around one minute to complete each task and we considered that the user failed the task if it took more than the established time. To test the RWD, we conducted the same testing both for the desktop and smartphone versions of the lo-fi prototype. To avoid biases in our testing, we divided the users in two groups: one that did the desktop version testing first and then the smartphone testing; and one group that made the smartphone testing first and then the desktop testing. This way, we were able to test both scenarios without causing possible inconsistencies in the results of the testing. Additionally, we conducted a small questionnaire, after the testing, to gather the feedback from the users regarding the layout and flow of the solution as well as the overall design.

Results

Most users found our proposed solution intuitive and easy to navigate. There were some issues pointed out by the users concerning the layout of some components. However, users were mostly satisfied with the design and organization of the webpages. The participants said that the simplicity of the solution lead to an easier interaction with this prototype.

These tests allowed to find issues at an early stage of the development. However, we believe that the fact that the test was done remotely lead to some issues that would not happen if they had been done *in loco*. The user interaction with the prototype was limited which lead to some errors that probably would not have happened otherwise. Nevertheless, and as said before, we were still able to find some issues with our design that could be resolved easily and early in the development.

Final prototype

In this section we will discuss all the stages of the implementation phase that lead us to create our final prototype. We will firstly talk about the weather API used to gather the data that we showcase in our website. We will then discuss the process and techniques used to create our website, both the server to collect the data and the front-end development needed to implement our intended solution.

Server

In order to develop the website we first needed set up all the required server side functionalities to guarantee that all the required data would be available. We used this server to connect the website to the API and for storage of same data. We made daily requests to the API to populate the local database in order to ensure the behavior of the forecast page and location pages for the 18 districts of Portugal, even if the API was down.

Frontend client

Having set up the server and the API connection we started developing our web application. As said before we used React to develop our solution. We use component base development [18] in order to organize our project and also to reuse code and thus reducing the complexity of the solution. We went through each page of our intended solution (our lo-fi prototypes) and identified the components needed to achieve the layout of each page.

We structured the website in six different webpages. The first page is the home screen which serves as a base for the whole website, where we can navigate to all the other pages. Then we have the main forecast page which displays a table with weather forecast information for the 18 districts of Portugal. We also have singular pages for every location of Portugal where we display detailed information about that specific region. We also have two pages connected to the IST Weather

[18] **https://www.tutorialspoint.com/software_architecture_design/component_based_architecture.htm**

Station which is a meteorological station that collects several types of data. We have a live observations page which showcases live data from the weather station and we also have an history of observations page where users can see the data for a specif time interval in the past. Finally, we have a satellite maps page where we showcase maps for the clouds and percipitation for the region of Europe.

EVALUATION

To evaluate our solution we made two testing phases: functional testing and user testing. In the functional testing we wanted to ensure our solution worked properly in different devices and in different browsers. For our user testing, we wanted to find deign and usability flaws in our solution.

Functional Testing

Firstly we tested different devices to ensure the RWD was well implemented. For that we used laptops, desktop screens, tablets and smartphones and tested every page to see issues with the responsiveness of the website. We did not found any major issues during this testing phase. We then tested our website in the mostly used browsers, defined in the requirements section, and we also did not find any major issues.

User Testing

In order to test our solution, we firstly needed to understand how we could test it and analyse the results. The main factors that define the efficiency and effectiveness of an UI are the following [3] [6]:

- Time to learn (average time for a user to learn to interact with the system).
- Rate of errors by users (average number of errors produced by a user or type of users)
- Retention over time (how much can users retain their knowledge about working with the system after a period).
- User satisfaction (usually measured with questionnaires or general feedback from the user after testing the solution).

Testing method

These tests are usually done in person, with the participants next to the developer which leads to immediate feedback and facilitated gathering of data. However, due to the covid19 pandemic, we once again had to do main part of the tests remotely. Nevertheless, in this phase we were able to have two different samples of tests: tests *in loco* and remote tests. We were able to gather two groups of participants and test our solution both in person and via computer. This way we could also do an analysis of the differences in results in the two types of tests.

In order to test our website we created a few tasks to test the interaction with the users, similarly to what we did in the lo-fi prototype. We wanted the tasks to cover the main functionalities of the website in order to collect feedback regarding the usability of each page. For that reason, we created tasks that simulated possible scenarios of when searching for weather forecast information.

These tasks covered every page of the website and all the features of each page. We wanted to understand how the users navigated through the website and if the gathering of information was done quickly. We measured the time required by each user to complete the task. Each task had a limit time for completion of ninety seconds. If the user was not able to complete within ninety seconds or if they gave up, we would consider the task as incomplete by that user. We also counted the errors, users made during each task. We considered errors when the participant navigated to a page or used a functionality not needed in order to complete a specific task.

In order to have comparable results, we also asked users to preform these tasks in the previous version of MeteoTecnico. This way we were able to compare our solution to the previous one and see if the changes made had a positive impact on the usability of the website, especially regarding the RWD.

For the testing phase, we were able to gather twenty four users. These users were divided in two main groups: *in loco* users and remote users. As said before, we did this differentiation so that we could compare results between this two methods. Additionally, in order to test the RWD, we also divided the users in testers of the desktop version and in testers of the smartphone version. Half the users tested the desktop version and the remaining half tested the smartphone version. Finally, to avoid biases in the results and since we were testing the previous version of MeteoTecnico as well, we needed to split the groups once again in two. Half of the people started by testing the previous version and then the new solution while the other half tested the new solution first and the old version after.

After the testing we did a small questionnaire regarding the satisfaction of the users. For that we used the System Usability Scale (SUS) [19] which is used to measure usability of websites.

Results

SUS results:

Our SUS questionnaires resulted in an average score of 86.04 amongst all the 24 participants. As stated before, a good UI usually obtains a score above 80, a result which our solution achieved. Therefore, we can conclude that most users where satisfied with our solution.

Overall results

After analysing each task individually, we can see that our solution improved the overall results both in the desktop version and especially in the smartphone version. We improved the times in an average of 4.43 seconds per task for the desktop version and 11.68 seconds per task in the smartphone version. We also reduced the amount of errors by 16 in total for larger screens and by 29 in smaller screens. We can see with these results that the users had a more efficient interaction with our solution then they had with the previous version of the website, especially when testing with smaller screen sizes. Besides the RWD, our tests showed that we improved massively

[19] `https://www.usability.gov/how-to-and-tools/methods/system-usability-scale.html`

the navigation through the website with our simplified header and site structure.

Regarding the remote and *in loco* testing comparison, we observed that testing in person achieved slightly better results in terms of average time spent per task. However, the margin is not enough to draw any major conclusions especially given the fact that our sample of users was not very big.

During the testing and with the feedback from the users, we were able to find a few usability issues in our solution. These problems were solved in order to achieve our final solution.

CONCLUSION

In this section we will do a broader analysis of the whole project. We will discuss the problems encountered during the development. We will also further analyse the results of our final testing phase and do a fair comparison to the previous version of MeteoTecnico. Finally we will discuss some topics that can be explored in future iterations of this work.

Project analysis

Following up the testing results, we consider that we achieved the proposed goals for this project. We created a UI solution for MeteoTecnico that made the website more user friendly and with a better RWD integration. We fulfilled all the requirements we proposed for our solution and we got positive results with the development results. We consider that part of the success of the project was due to the precise definition of key concepts and the extensive analysis of related work. By examining other weather forecast websites, we were able to see the best practices both in general UI development as well as in meteorological related issues, such as data to be displayed and ways to showcase information. By doing this analysis, we were able to identify the problems with the current solution of the website and, consequently, come up with solutions to those problems.

The development was the other key part of our project. We were able to properly organize the development in all the required phases of a UCD. Firstly, we did the lo-fi prototypes and tested them to find issues in our approach to some of the design decisions. Then, we created our solution using React, Sass and several other technologies which facilitated the development, with most of the focus on the RWD. With our solution done, we did functionality tests that ensured that the website worked properly in all main browsers and devices, and then we did our usability tests. The results of these tests showed that not only were participants satisfied with our solution, but the new version of the website was more intuitive which lead to better results on interactions with the users. However, we were able to find some issues with our project that we corrected for our final solution.

The major obstruction to the development of our solution was the covid19 pandemic. The testing phases were limited by the fact that we could not get together with people in order for them to test our solution. This lead to us trying a slightly different method, were most tests were done remotely. Especially regarding the lo-fi prototypes, where users are normally required to interact with the paper prototypes, we were able to preform the tests by filming the prototype as we asked the users to describe the actions they wanted to preform. This method caused some issues regarding the flow of the testing phase, but the participants were still able to give feedback about our proposed solution. In the final prototype testing phase, since we were testing an actual website, the remote tests did not have that much impact in our results. In fact, by the results gathered, we saw that the remote tests preformed slightly worse than the *in loco* tests, however, this difference was not significant, especially since we did not have a large sample size.

Future work

In this section we will discuss the possibilities for future improvements in our website. These are improvements that, either due to the time needed for implementation or due the lack of knowledge in certain areas of the development, were not integrated in our final solution.

One of the problems that we referred in our analysis of the previous version of MeteoTecnico was the lack of information given to the user about the specific location when the user was in a *Location forecast* page. We tried to find datasets that would allow us to match a coordinate to a specific region in Portugal. However, our efforts to find that dataset did not grant any results. With this datasets we would be able to display more information about a specific location to the user.

Customization of the website was one of the focus of our solution. For that reason, we created the ability for the users to change between light and dark modes and also to change the language of the website (Portuguese or English). However, in order for the website to be accessible to a larger audience, we could add different languages such as Spanish or French [20].

Another customization feature that could be added is the ability to change the unit of the displayed data, for example changing the temperature from Celsius to Fahrenheit or the wind speed from m/s to knots. This would aid users from different countries, since in, some places, people are used to utilize different measuring units.

One of the features that some of the analysed websites in the related work had was the possibility to view live data for specific regions. Users are able to have live updates of the weather forecast at each time of the day. This feature is dependant of the API where the data is gathered. In the API used in the development of this website, the only available live data was regarding the IST Weather Station, which meant that this information was only relevant to people leaving near Lisbon.

Important to note that a website like this, when in production (available on the internet for all the users), needs to have continuous user data analysis and bug reports in order to find usability issues that need to be corrected. There are also always new features, like the ones enumerated before, that could enhance the experience of the users when using this website.

ACKNOWLEDGMENTS

[20] `https://www.visualcapitalist.com/100-most-spoken-languages/`

I want to thank Professor João Brisson for the guidance and support given throughout all the development of this project. I also want to thank my family and my girlfriend for helping me and supporting me during this development.

REFERENCES FORMAT

References must be the same font size as other body text.

REFERENCES

1. McLuhan, M.: The Gutenberg Galaxy: The Making of Typographic Man. University of Toronto Press, Canada (1962)
2. McLuhan, M.: Understanding Media. McGraw-HillEducation, Canada (1964).
3. Sajedi, A., Mahdavi, M., Pourshirmohammadi, A., Monajjemi Nejad, M.: Fundamental Usability Guidelines for User Interface Design. In: IEEE Xplore, Computational Sciences and Its Applications ICCSA '08, Italy (2008) doi:10.1109/ICCSA.2008.45
4. Fonseca, M. J., Campos, P., Gonçalves, D.: Introdução ao Design de Interfaces. 2nd edn. FCA - Editora de Informática, Lisboa, Portugal (2016)
5. Friedman, M. G., Nelson Bryen, D.: Web accessibility design recommendations for people with cognitive disabilities. Technology and Disability **19**(4), 205–212 (2007) doi:10.3233/TAD-2007-19406
6. Abras C, Maloney-Krichmar D, Preece J.: User-Centered Design. In: Bainbridge W (ed). Encyclopedia of Human-Computer Interaction. Thousand Oaks: Sage Publications (2004)
7. Rossi, G., Schwabe, D., Guimarães, R.: Designing Personalized Web Applications. In: Proceedings of the 10th international conference on World Wide Web, 275–284 (2001) doi:10.1145/371920.372069
8. Dix, A., Finlay, J., Abowd, G. D., Beale, R.: Human-Computer Interaction. 3rd edn. Prentice-Hall, United States of America (2003)
9. Cato, J.: User-centered Web Design. 1st edn. Addison Wesley, United States of America (2001)
10. O.M.F. De Troyer, C.J. Leune: WSDM: A User Centered Design Method for Web Sites. In: Proceedings of the 7th International World Wide Web Conference, pp. 85–94. Elsevier Science Publishers B. V. , Brisbane, Australia (1998)
11. Almeida, F., Monteiro, J.: The Role of Responsive Design in Web Development. Webology **14**(2), 48–65 (2017)
12. Waseem, I., Hammouri, A. I.: Responsive Web Design Techniques. International Journal of Computer Applications **150**(2), 18–27 (2016) doi:10.5120/ijca2016911463
13. Giurgiu, L., Gligorea, I.: Responsive Web Design Techniques. International conference KNOWLEDGE-BASED ORGANIZATION, **23**(3) (2017) doi:https://doi.org/10.1515/kbo-2017-0153
14. Hardianto, D., Nizar Hidayanto, A.: analysis of User Experience Quality on Responsive Web Design from its Informative Perspective. International Journal of Software Engineering and its Applications **8**(5), 53–62 (2014) doi:10.14257/ijseia.2014.8.5.06
15. Marcotte, E.: Responsive Web Design. 2nd edn. A Book Apart, New York, United States of America (2014)
16. Steadman, R. G.: The Assessment of Sultriness. Part II: Effects of Wind, Extra Radiation and Barometric Pressure on Apparent Temperature. Journal of Applied Meteorology (1962-1982), **18**(7), 874–885 (1979) doi:https://doi.org/10.1175/1520-0450(1979)018<0874:TAOSPI>2.0.CO;2
17. Steadman, R. G.: A Universal Scale of Apparent Temperature. Journal of Climate and Applied Meteorology, **23**(12), 1674–1687 (1984). doi:https://doi.org/10.1175/1520-0450(1984)023<1674:AUSOAT>2.0.CO;2
18. Gigerenzer, G., Hertwig, R., Van Den Broek, E., Fasolo, B. and Katsikopoulos, K.V.: A 30% Chance of Rain Tomorrow: How Does the Public Understand Probabilistic Weather Forecasts?. Risk analysis **25**(3), 623–629 (2005) doi:10.1111/j.1539-6924.2005.00608.x
19. Eisfeld, H., Kristallovich, F.: The Rise of Dark Mode: A qualitative study of an emerging user interface design trend. (2020).
20. Nissinen, T.: User experience prototyping - a literature review. (2015).
21. Preece, J., Rogers, Y., Sharp, H.: Interaction Design: Beyond Human-Computer Interaction. 4th edn. John Wiley & Sons Inc, United States of America (2002).
22. Hoadley. E.: Investigating the effects of color. Commun. Association for Computing Machinery, **33**(2), 120–125 (1990). doi:https://doi.org/10.1145/75577.75578
23. Filzen, J.J., Simkin, M.: Using Color to Enhance Spreadsheet Accuracy and Usefulness. ISACA Journal (2016).
24. Still, B., Crane, K.: Fundamentals of User-Centered Design: A Practical Approach. 1st edn. CRC Press. United States of America (2017) doi:https://doi.org/10.4324/9781315200927