



# UNIVERSITAT DE BARCELONA

Final Degree Project  
**Biomedical Engineering Degree**

**“Development of an app to study the attention  
through the migraine cycle of a patient”**

Barcelona, 15th January 2021

Author: Lizaveta Shchasnovich

Director/s: Xim Cerdà Company

Tutor: José M<sup>a</sup> Gómez Cama

## **Abstract**

Migraine is a brain disorder that causes episodes of pain with high intensity. This pathology has a high disability rate, and it affects people worldwide in their everyday basis. Therefore, this article looks forward to developing an application to test the attentional response of patients that have migraine, to be able to establish in the future the variability in attention as a predictor of migraine attacks. Through the development of different tests such as a reaction time test, or an object orientation test, the application will show the difference in the attentional response of the users.

This project has been done with the collaboration of the Research Laboratory in Headache and Neurological Pain of the VHIR (Vall d'Hebron Research Institute).

## **Key Words**

Medical Applications, Migraine, Reaction Time, Object Orientation, Attentional Response

## INDEX

1. Introduction.....	4
2. Background.....	5
3. Market analysis.....	8
4. Conception engineering.....	8
5. Detail engineering.....	12
6. Execution chronogram.....	18
6.1 Work Breakdown Structure and dictionary of tasks.....	18
6.2 Dependency Matrix.....	21
6.3 PERT/CPM diagram and critical path.....	22
6.4 Gantt Chart.....	23
7. Technical feasibility.....	24
8. Economic feasibility.....	25
9. Regulations and legal aspects.....	26
10. Conclusions and future overview.....	27
11. References.....	29
12. Annexes.....	30

## 1. INTRODUCTION

Migraine is a brain disorder which causes moderate or severe chronic headache. It is characterized by episodes of pain with high intensity, and it has a high disability rate. According to the WHO, its prevalence is more than 10% worldwide, and it is the sixth highest cause worldwide of disability with respect to lost years, suffering and lack of productivity. [1] The cause of this pathology is related to the trigeminal nerve and the trigeminovascular system, which is one of the key factors in the expression of the headache as it transmits the sensitivity of the head through the trigeminal ganglion's peripheral axons that reach the meninges. When the fibers of this nerve are activated or irrigated, they release inflammatory substances which act on the surrounding blood vessels located in the meninges. Therefore, during a migraine attack, inflammation of the meninges occurs as a result of the release of these substances, causing a pain that gets worse with movement. For this reason, being at rest is a relief for the patient. The duration of a migraine attack can last from four to seventy-two hours, and its periodicity goes from once a year to several times in a row, which causes the high disability rate. [2] Migraine has four very definite phases: the prodrome (before the attack), the attack, the postdrome (after the attack) and the interictal (the phase between the prodrome and the postdrome), which have each different characteristics and indicators.

Only a better understanding of the mechanisms of this pathology will allow a greater awareness of the complexity of this brain disease, pushing new treatments available for patients that will help them to recover in the future.

In the Hospital of Vall d'Hebrón, they have a specialized Headache Unit, which collaborates with the Research Laboratory in Headache and Neurological Pain of the VHIR (Vall d'Hebron Research Institute). With professionals such as Dra. Patricia Pozo Rosich, they have been studying this disorder from a multidisciplinary approach using genetics, neuroimaging, and electrophysiology. She has also developed the Migraine Adaptive Brain research program; she has written many articles and she has created a webpage to help patients understand the disease better and improve their lifestyle. [3]

Therefore, the goal of this project is to develop an Android and iOS application for mobile phones to study the attention of the patients through the cycle of the migraine, which will be used by the VHIR to investigate further the illness and treat in the best way possible the patients, correlating the variability in attention as a predictor of migraine attacks. The name of the application was chosen to be "Mibrain(e)", making a play on the words migraine and brain.

The long-term goal of this project is that this application becomes a helpful tool within this field, and that others can develop more functionalities to the app that would sum up to the ones that will already be there, improving the study of migraine overall.

The main limitations of this process are the time available to develop the app, which will be limited to September, and the health situation in which we are, which restricts the time available to test and improve the application with patients from September until January, if it is possible to access the hospital at that time due to the epidemic of Covid19. Other limitations include the programming language constraints, considering that the goal is to program the app for both mobile operating systems.

Specifically, the app will have two tools to test the attention of the patients:

1- A simple tool to measure the reaction time, with a screen/buttons that changes colour and with the patient that must click on it as fast as possible.

2- Visual Search: define a target object which the patient will look for and show many objects. He/she should click on the screen in case this object is present. For example, having a screen full of straight lines and one of them with a different orientation. Apart from measuring the reaction time, we could also measure the degrees of difference in the orientation of the line to be detected.

For future applications and developments of this app, changes in size, colour, contrast, shape and so on could also be considered, but due to the limitations present in this project, only these two tools will be programmed.

The app is intended to be easy to use and visually pleasing, as it is targeted for all patients of all ages suffering from migraine. Once it has been programmed (maximum until September), the idea is to test its effectiveness with trials that we will do at the Hospital with few patients, after which I will correct and launch the final version of it and present it on January.

## **2. BACKGROUND**

Reaction time is a measure of how quickly an organism can respond to a particular stimulus. Nowadays, there are many different tests available to measure our reaction time in different platforms. Some examples of this are the webpage “human benchmark”, which does a test of the brain, measuring different abilities which include number, visual and verbal memory, hearing and typing capacities; and tracking your stats. [4] Also, there are some other webpages like the one made by Jim Allen in 2002 for

the University of Washington [5], which consists in a traffic light that changes colours, or the one developed by Justpark, a parking application that also has this test as a driving simulation [6].



Figure 1. Justpark.com website to test the reaction time.

In the field of mobile applications, approximately 90 reaction time test apps were identified in the Google Play platform for Android, which had different designs and uses. From simple games to competitive and brain stimulating variants, there were also some applications focused on brain exercising, but none of those apps was specifically designed to measure the reaction time of patients with migraine.

Therefore, measuring our reaction time is something which is useful in many different areas of life (from driving, to playing videogames, and so on), which explains the great number of webpages and applications that offer simple tests to measure it.

Looking at the works published in the database NCBI (National Center for Biotechnology Information), in the last ten years there have been 46 studies published that include reaction time and migraine. Also, there have been 8 studies about mobile applications and migraine, which include studies about the privacy issues surrounding apps or progressive reminder system (so patients with migraine take their medication); and there have been 22 studies which included reaction time and mobile applications. No studies have been found with the three parameters (mobile phone apps, migraine and reaction time).

Some of the most related studies on the attention of patients with migraine that have been found are the following:

- “Sumatriptan nasal spray and cognitive function during migraine: results of an open-label study (2001)”, which examined measures of cognitive function during acute migraine, before and after treatment with sumatriptan nasal spray [7]. To do so, they used the Headache Care Center-Automated Neuropsychological Assessment Metrics, a computerized neuropsychological assessment battery, testing the cognitive function of the subjects (which include the simple reaction time, sustained attention/ concentration, working memory, visual-spatial processing),

and also their alertness/ fatigue. As results, they found that the patients with migraine had a worse performance compared with migraine-free people, and this was rapidly restored following the sumatriptan nasal spray, but it also depended on the pain severity and clinical disability of each subject.

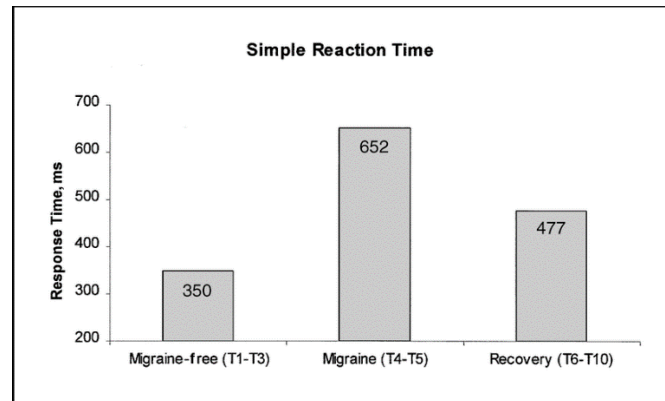


Figure 2. Mean simple reaction time at three subject conditions.

- “Duration and Frequency of Migraines Affect Cognitive Function: Evidence From Neuropsychological Tests and Event-Related Potentials (2017)”, which evaluated the changes in the cognitive performance of patients with migraine with different tests in language, memory, executive functions, calculation and orientation. For the attention measurement, they did a vigilance test (reaction time) and digit span test (visual memory). The authors also concluded that cognitive performance decreases during migraine, depending on the frequency and duration of the attack.
- “Migraine understood as a sensory threshold disease (2019)”, which studies the altered sensitivity to sensory inputs in patients with migraine, quantifying sensory threshold changes over time. [15] The article concluded that people with migraine in general have lower sensory thresholds compared to healthy controls, which changes in the different phases during a migraine cycle.
- “A Review of Motion and Orientation Processing in Migraine (2019)”, an article that reviews two visual tasks, global motion discrimination and orientation, to track changes in performance during the migraine cycle, suggesting that visual test can have practical uses to treat and manage migraine. [16]

Following the discoveries in recent studies on the subject, this application can be useful in correlating migraine with attention problems and, therefore, investigating further the ways to improve people’s lives.

### **3. MARKET ANALYSIS**

This project is aimed at the professionals from the VHIR who are studying patients with migraine, and it will help them to understand better how the attention is altered during the different cycles of this illness. Through the measure of the reaction times to stimuli, they will evaluate the attentional function of the subjects, how these responses vary throughout the migraine cycle, and how to study the predictive character of these reaction times.

As we saw in the background section, there are already many apps that test the reaction time focused on several markets (health, gaming, driving, among others). I am focused on developing an app to specifically cover a research need of the VHIR migraine research group, so for this reason no commercial action is expected from it, and I will not be competing in the market with these other apps.

Gradually, with the advances in technology more apps of this kind are being developed. Gaming, for example, is becoming more popular than ever, and it requires high attention and reaction levels from the users. In 2016, 60.5% of people in North America as well as in Western Europe owned a smart device, a figure that is expected to increase significantly by 2020 and each year. [9] Consequently, the apps will increasingly have more weight in the medical as well as in other fields, considering that phones are becoming a must have in our everyday life, and apps they are visual, easy to use, fast and comfortable.

### **4. CONCEPTION ENGINEERING**

To create the application, many software options are available. The usual languages for programming are Java, C/C++, HTML5 + CSS, Javascript, among others. Depending on the difficulty level and programming skills in each case, some options are better than others. Apart from the programming languages, there are also IDEs (Integrated Development Environments), software suites that consolidate the basic tools required to compile and run the code directly, making it easier to create the app. IDEs include text editors, code libraries, compilers and test platforms, an integrated toolset that simplifies software development acting as a single framework, application or service, and minimizing coding mistakes and typos. Not using an IDE would mean that the engineering would have to select, deploy, integrate and manage separately everything, which would highly increase the amount of work required. IDEs can be open source (free for users) or commercial offerings. Some examples of these are Android Studio, Eclipse, IntelliJ IDEA, Visual Studio or Flutter. [10]



In my case, to program the application I will be coding in C# and .NET Framework, which are integrated in the free IDE Visual Studio, specifically in Xamarin, a cross-platform mobile IDE.

Considering that I want to develop a compatible APP for Android and iOS, Xamarin is a good option because it can create code for multiple mobile platform types, delivering native apps for iOS, Android, and Windows with a single .NET shared codebase.

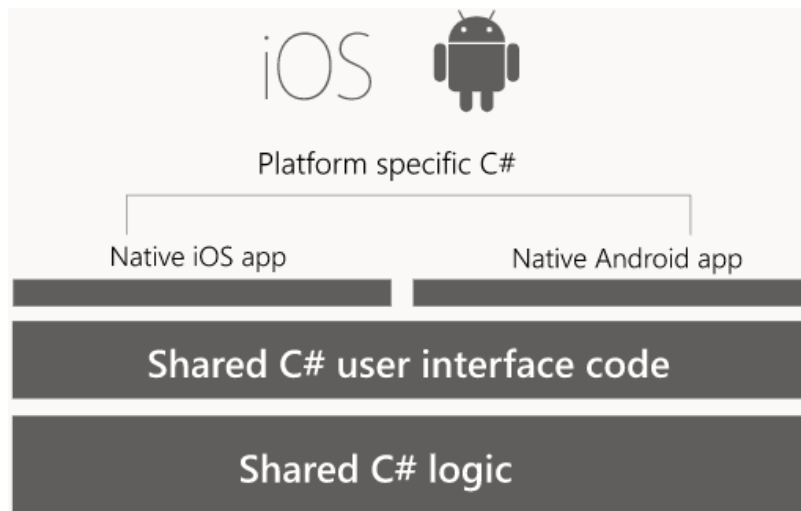


Figure 3. Scheme of C# as a shared base unique code for iOS and Android.

More in detail, C# is a programming language designed to create applications that run on the .NET Framework. It is effective and object-oriented, and it is based on the .NET Compiler Platform "Roslyn", which provides sophisticated code analysis APIs (application program interfaces) and has an open code source on GitHub. APIs are a set of routines, protocols, and tools for building software applications, specifying how its components should interact. [11]

As it was specified in the introduction, the application will have two tools to test the attention of the patients: a screen that changes colour and the patient that must click on it as fast as possible (measure the reaction time), and a visual search screen, with a target object which the patient will have to identify in between many objects.

Shown below is the first draft and flow diagram of how the application will function in terms of interface and screen displacement:

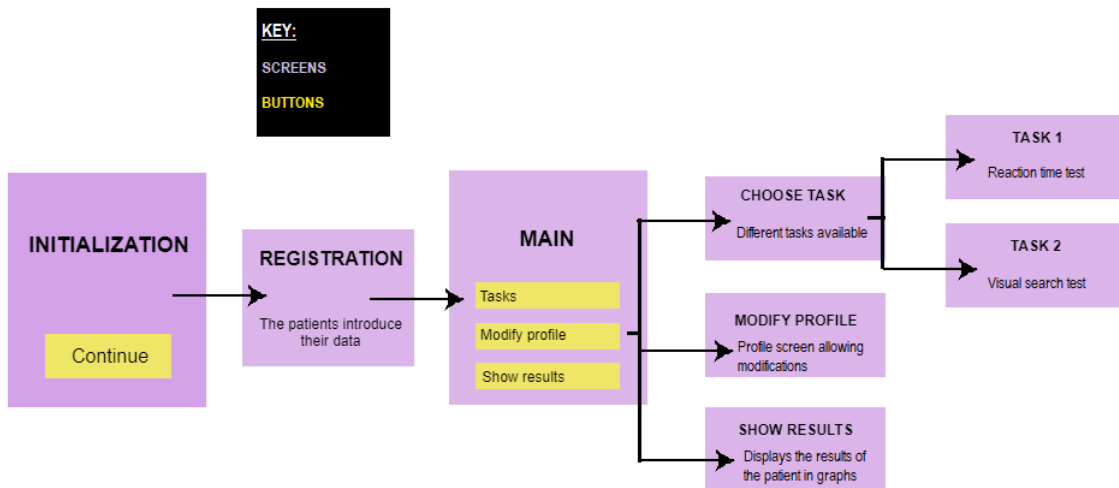


Figure 4. Design of the screens of the app.

The application will have different screens with different functions, always having in mind that it should be easy to use and visual, as it is intended for people of all ages. The first screen will welcome the patient to the application. Once he presses the “Continue” button, it will show the registration screen, which will only pop up the first time we install the application, as it is intended for one user only. Here, it will ask for the registration of the patient, which will include the following data: name, email, birthday, gender, and some questions about his experiences with migraine. Afterwards, the main screen will appear, which will have several functions: tasks; an overview of the results of the tasks so far; and a button to view and edit the profile (to be able to change or update the data that was introduced in the first time of registration).

On the tasks screen, there will be two options depending on the type of test that is going to be performed: the reaction test or the visual search test, which will have each a button. Each test will have a screen where the subject will have to press on a button/object, and after the click the result will automatically appear on the screen, allowing the user to save the result in and to go back to repeat the test. The idea was that from the database, the user would be able to export the information obtained into different formats (pdf, excel, and so on).

The first idea that I had to design the application looked like the following:

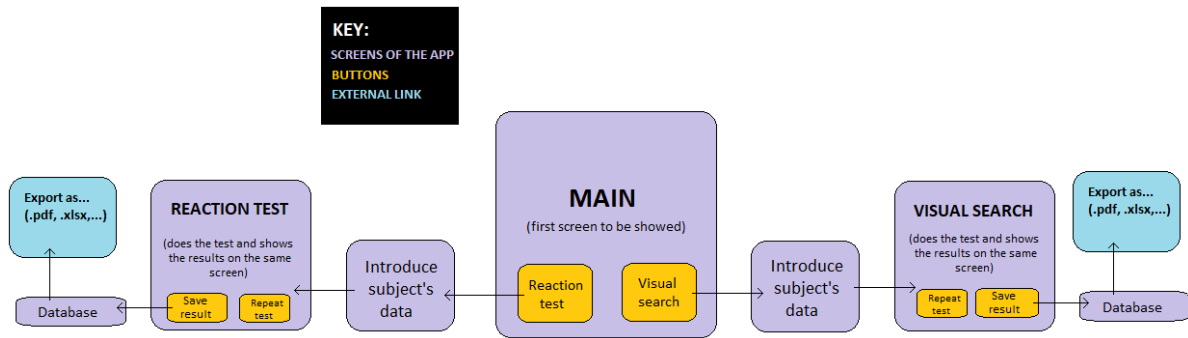


Figure 5. First interface design.

This was rejected because we realized it would be more comfortable to have the application installed and used directly by the patients, so they could perform the test at home or at any moment as they cannot control when and where the migraine attacks start.

The next model that I came up with was this one:

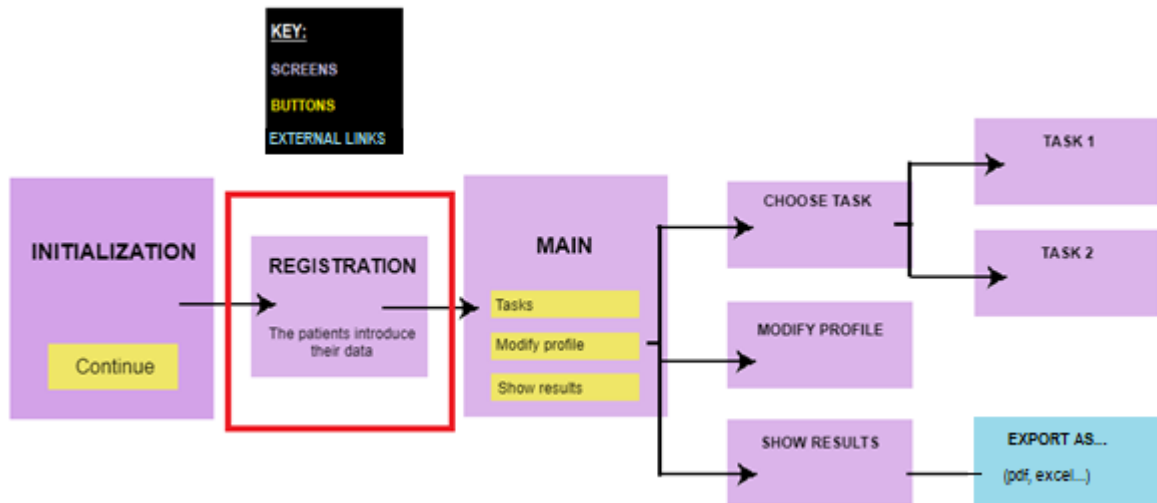


Figure 6. Second interface design.

To display the results in a more appealing way, we thought that showing them in a graph on the screen would be more visual for the patients, it would encourage them to improve their results, and it would show their progress in a more intuitive way. Also, we realized it wasn't useful for them to be able to export those results directly from the application, as it is something that was needed for the researchers, so we removed the screen with the external exportation, creating the final design of the application:

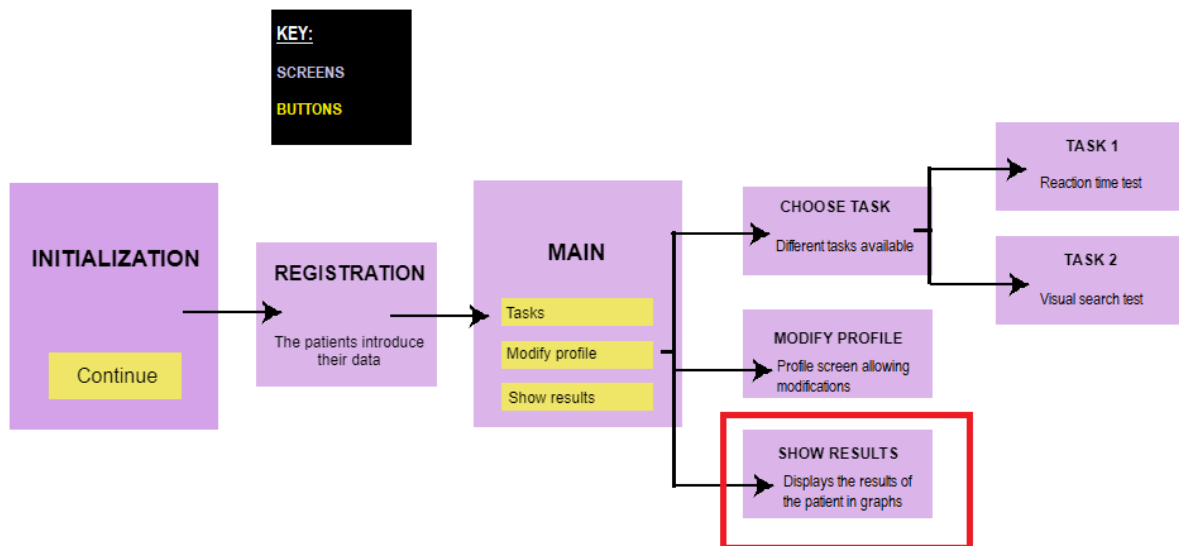


Figure 7. Final take on the interface design.

## 5. DETAIL ENGINEERING

For the detail engineering, I will design each screen accurately so it can be programmed afterwards through Visual Studio, which allows different interfaces, colours, alignments, names and many other options. This superficial part of the app will be done programming through the XAML Designer (.xaml files), which provides a visual user interface, and will be integrated with all the coding part of the project in C# (.xaml.cs files), which will make its functionalities possible.

```

3
4 using Xamarin.Forms;
5 using Xamarin.Forms.Xaml;
6 using System.Runtime.Serialization.Formatters.Binary;
7 using SkiaSharp;
8 using Microcharts;
9 using System.Collections.Generic;
10
11 namespace Mibraine
12 {
13     [XamlCompilation(XamlCompilationOptions.Compile)]
14     public partial class DatosGuardados : ContentPage
15     {
16         private Data data;
17
18         public DatosGuardados()
19         {
20             InitializeComponent();
21
22             try
23 
```

Figure 8. Screenshot of the Visual Studio project.

### 5.1. Initialization, registration, and main screen

When first accessing the app, the initialization screen will pop up, which will have the registration button. Before accessing the registration, the Privacy Policy page will pop up, which will have to be accepted if the user wants to continue and register in the app.

On the registration page it will ask for some personal information of the user (name, email, gender, birth date). After this information is introduced, the user will press on the registration button, and he will be redirected to the main screen of the app.

Once the user is identified for the first time on the phone, these screens will not appear anymore, showing to the patient directly the main screen. This will be thanks to saving the information of the user in the internal storage of the cell phone, so it can be checked every time that he/she wants to use the app.

The main screen is the screen that presents to the user the function of the desired application. This screen consists of 3 buttons and the application logo, which will be in the central part of it.

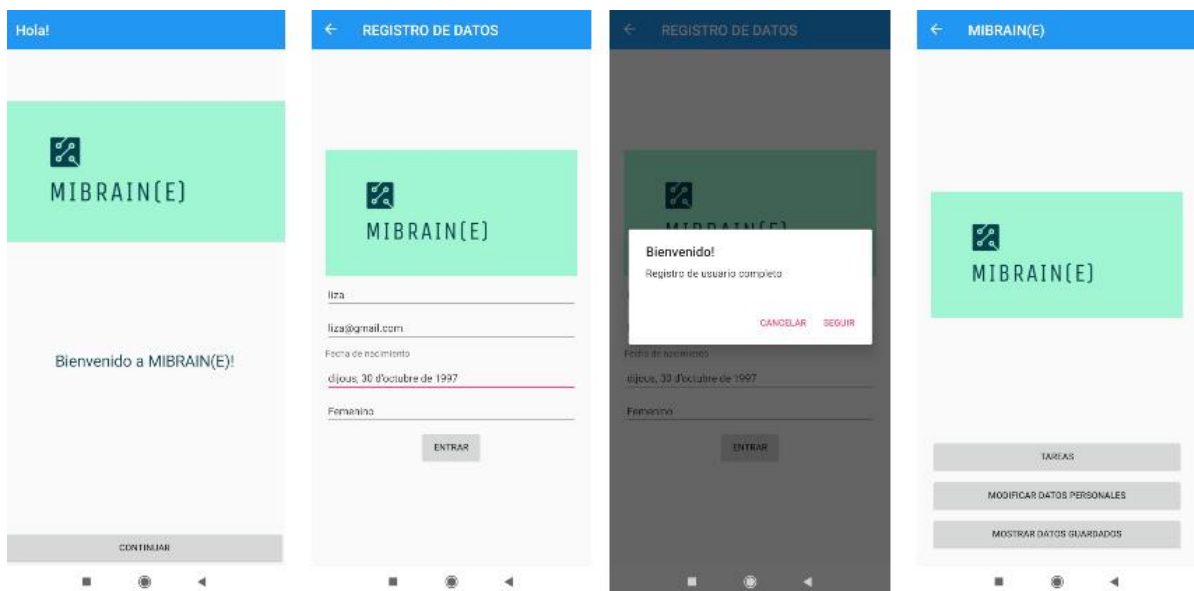


Figure 9. From left to right, overview of the initialization screen, registration screen, and main screen of the application.

To illustrate on how this first process works, a diagram flow has been created. It is very important to save the user's personal information properly, and to be able to check it afterwards every time they access the app to avoid rewriting and duplicating data.

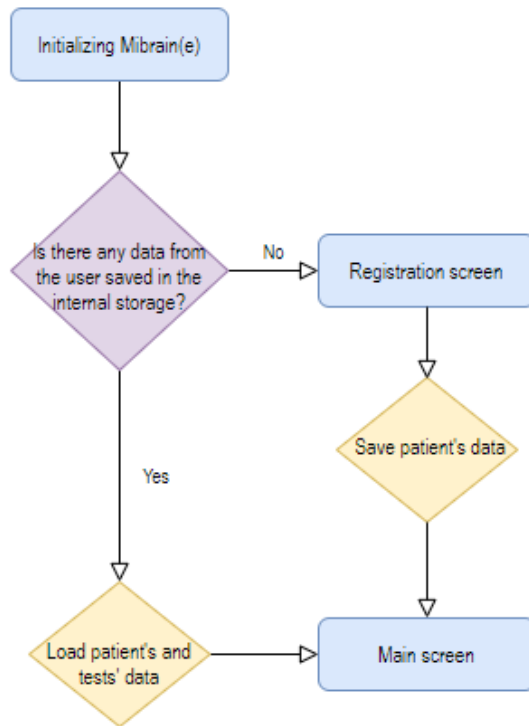


Figure 10. Flow diagram of the first screen of the app.

For further information about the workflow of the screen, go to the Annex in order to visualize the programming code and its detailed explanation.

## 5.2 Tasks screens

From the 'Main' menu, the user will be able to move to the 'Tasks' menu. In this part of the app, the 'Tasks' screen will be used as a pivot, which means that to change from one task to another the user will not have to get back to the 'Main' screen and, thus, avoiding having too many options displayed. This part of the app will be focused on storing the important information about the current state of the user as well as displaying this information in a visually friendly way, so he will be able to take a look at his current results and try to make better landmarks.

For now, this screen will have two buttons displayed which will let the user swap screens between tasks, one being 'Reaction Time', and the other being 'Object Orientation'. The long-term goal of this application is that more tests are developed to help the researchers to investigate the attention cycle of patients with migraine.

The screen of each task will include its title, a brief description of the procedure they will come across, explaining how the test works, and a button to start it. This button is located on the bottom part of the screen, so it is more comfortable for the user to press it with the thumb, allowing a faster response to it.

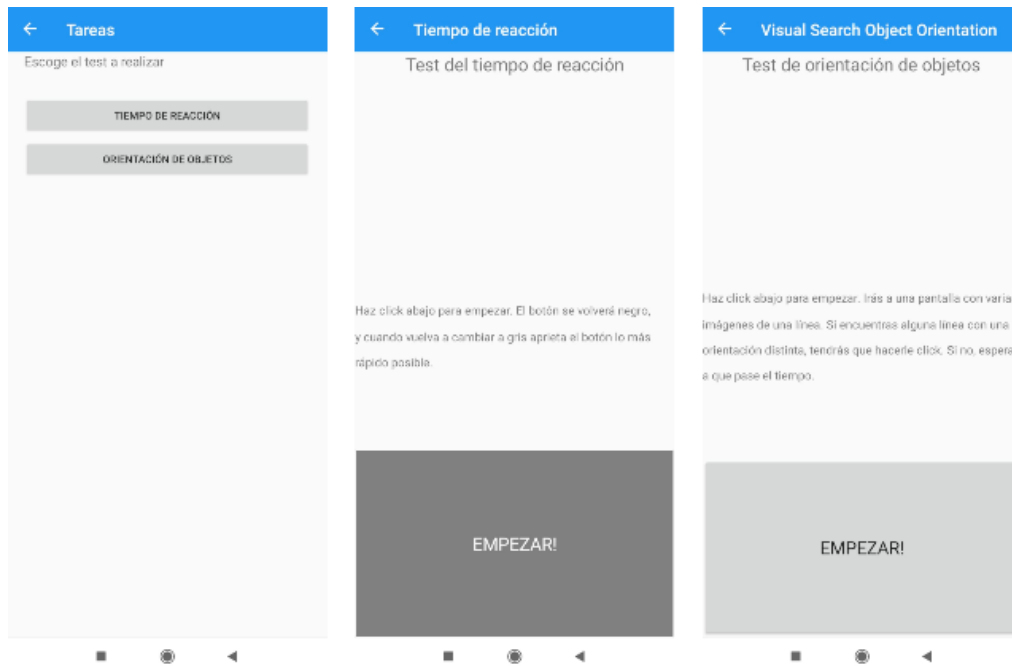


Figure 9. From left to right, overview of the tasks, reaction time and object orientation screens.

In the reaction test, once the user clicks on the button, it will directly change its colour to black, and when it goes to gray again it will detect how long does it take for the user to press it. Then, the result will be directly displayed on the screen, allowing the user to save it, and returning to the task screen so the patient can try to do the test again.

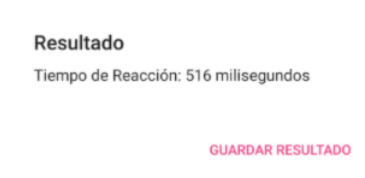


Figure 10. Pop-up screen that will show the user his result after the reaction time test.

With these results, researchers will be able to understand better the neurophysiological reactivity before a migraine attack and how it changes through the different migraine cycles. Sensory neurons are the ones responsible of converting the stimulus into an electro-chemical signal, which flows through the length of the sensory neuron.

The average visual reaction time of people in general is between 200ms to 300ms, even though this depends on many different factors such as gender, physical condition, fatigue, intelligence, and others. Another interesting element to be considered is that there are significant differences in the reaction time of the visual system for different coloured light stimuli, which is due to the systems involved in the visual processing, colour perception and activation of cone cells. [17]

The control group results will be contrasted with the pathological results to see significant changes in the simple reaction times.

In the visual search test, the user will be redirected to another screen with lines, and if he finds one that has a different orientation than the other, he will have to click on it. The program will save if it was a successful result or if the user was mistaken, and it will also save the time it took for him to press the button. If all the lines have the same orientation, after 3 seconds of awaiting the screen will show a message informing the user that he has got the test result right.

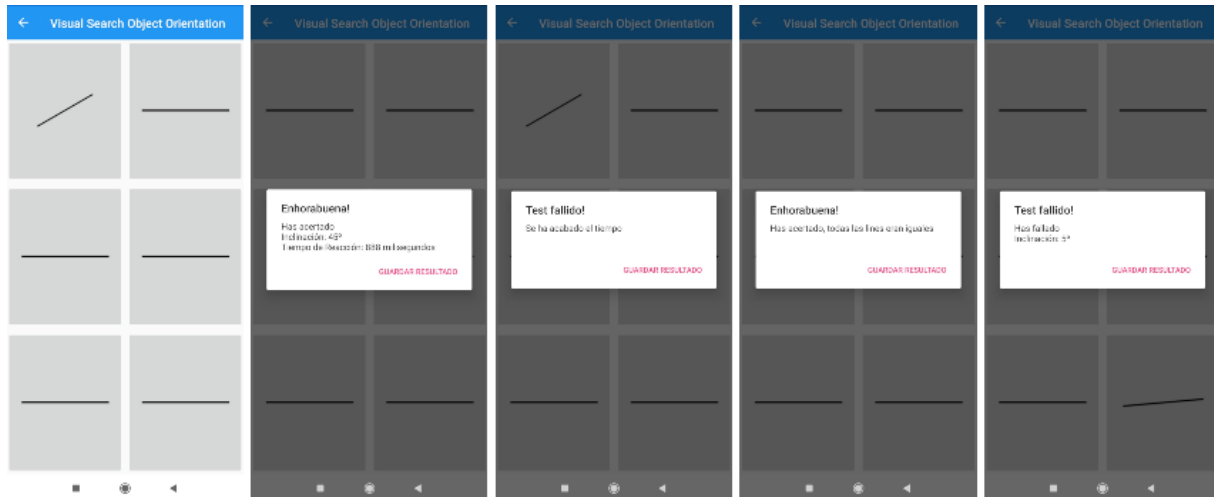


Figure 11. Different screens for the results of the object orientation test.

To get a clearer picture of how the data is obtained from this test, the flow diagram from below can be consulted. By saving the right and wrong answers, the results can be displayed with different colours afterwards in the graphs.

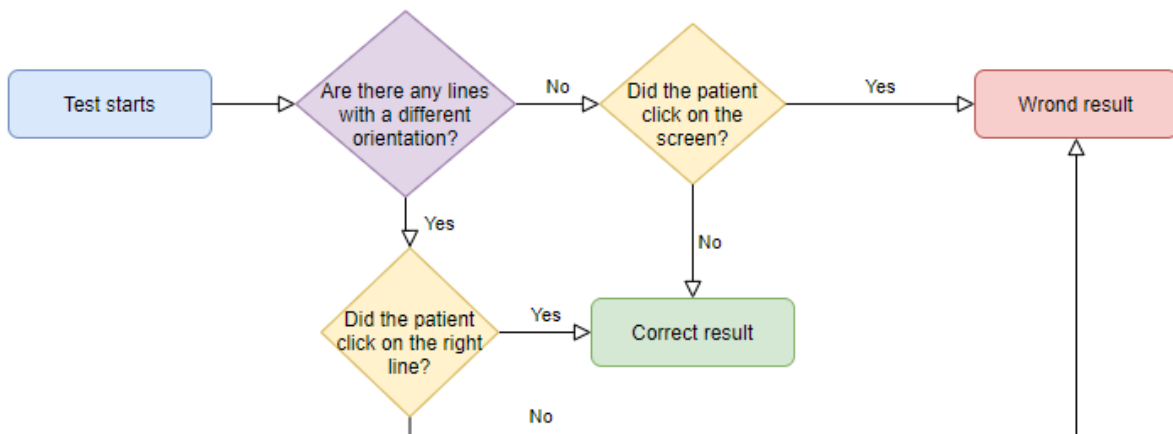


Figure 12. Flow diagram corresponding to the object orientation test.

On this second task, these kinds of orientation stimuli are processed by neurons in the first cortical regions. It has been observed that there are systematic differences in the processing of the stimuli between patients and control people, and differences in this type of task have also been observed, showing that the migraineur's apparent response time advantage in the low-level tasks provides a reason to think of an oversensitivity to visual stimuli. [18]



### 5.3 Modify profile and show results screens

Finally, the app will also provide the function of reviewing all the data saved from the user and his tests. In the 'Modify profile' screen, accessible from the main menu, the patient will be redirected to a screen where he will be able to modify his personal data that he introduced the first time he logged in. From the main menu, there will also be the option to see the tests' results. To be able to graphically display all these data, the application will show the results with the Microcharts.Forms NuGet package (from Xamarin.Forms). In this way, users will be able to visually appreciate the distribution of their attention test result values over time.

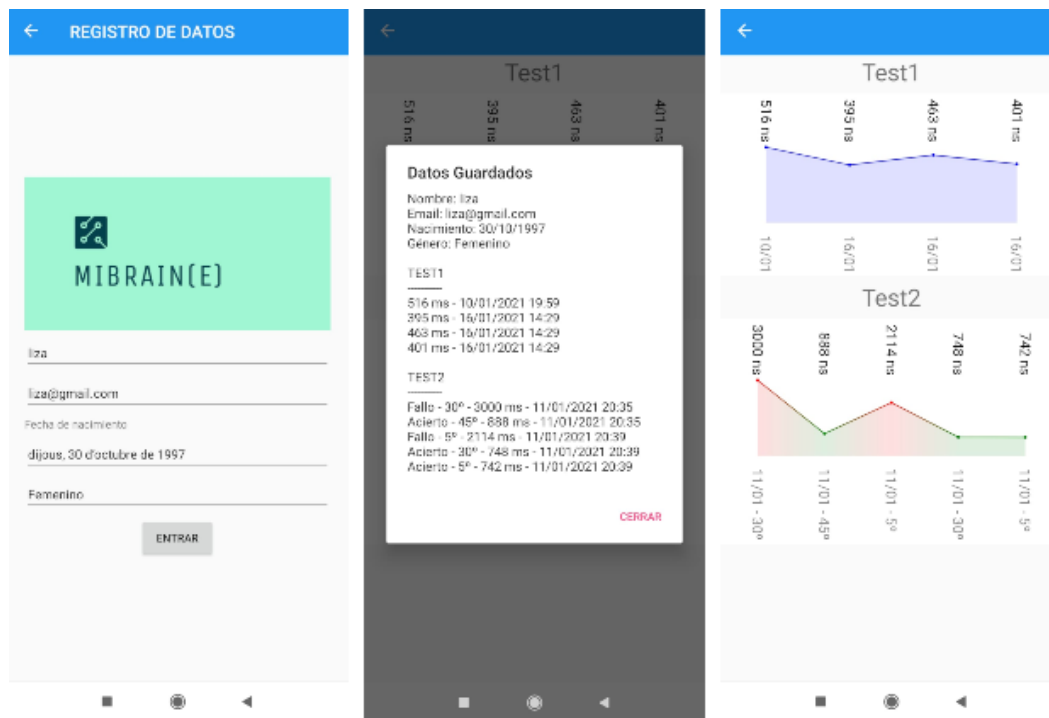


Figure 13. From left to right, screen to modify the profile of the user, results pop-up screen, and graphs.

By comparing the data from patients with migraine to the control group's data, the goal of the researchers from Vall d'Hebron is to confirm that people with migraines have a longer reaction time (they respond more slowly) than control people.

Also, through the development of the app, patients will be asked to perform the tasks every day without having to come to the Unit, having a larger number of subjects as well, rather than if the experiments were organized there.

## 6. EXECUTION CHRONOGRAM

### 6.1 Work Breakdown Structure and dictionary of tasks

The proper management of the project is an extremely important part of it. A Work Breakdown Structure is a hierarchical decomposition of the work into smaller, easier to manage work packages that will be scheduled, costed, and controlled to achieve the project objectives and create the required deliverable products. Therefore, the WBS is essential to organize and distribute the tasks to complete the project in a successful way.

Down below, there is the schematic of the WBS, as well as the definition and description of each task in detail, which conforms the WBS's dictionary.

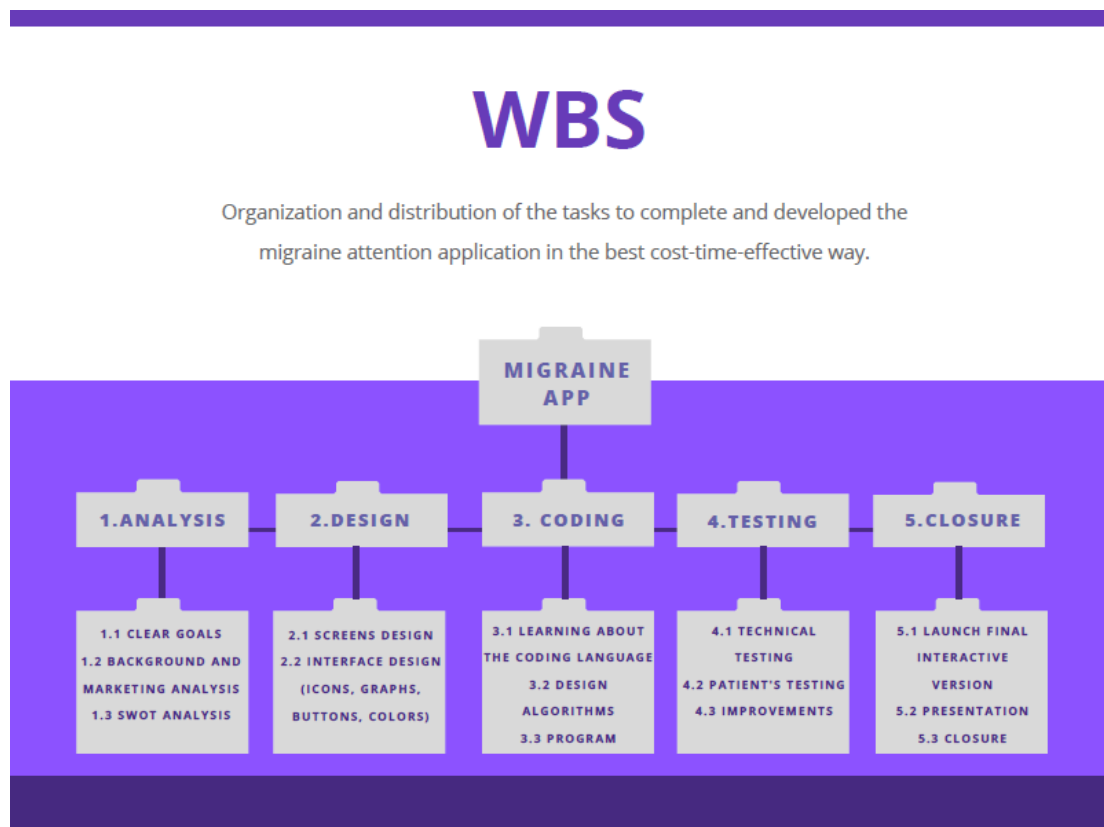


Figure 14. WBS diagram of the project.

The following table shows the coding for the tasks of the project, which relates each activity with a letter and its identification number.

I will use this codification in the following task descriptions and graphs.

<b>Reference in WBS</b>	<b>Code</b>	<b>Activity</b>
1.1	A	Clear goals
1.2	B	Background and marketing analysis
1.3	C	SWOT analysis
2.1	D	Screens design
2.2	E	Interface design (icons, graphs, buttons, colours, and so on)
3.1	F	Learning about the coding language
3.2	G	Design algorithms
3.3	H	Program
4.1	I	Technical testing
4.2	J	Patient's testing
4.3	K	Improvements
5.1	L	Launch final interactive version
5.2	M	Presentation
5.3	N	Closure

Down below, there is the description for each task, from the general ones to the more specific ones, labelled with their corresponding code.

1. Analysis: This is the first step of the project, in which I will the gather information and define the application. Also, it includes an analysis of the things to be considered when designing it.
2. Design: During this section I will do the design of how the application will look like.
3. Coding: In this step, all the programming and coding for the app will be done.
4. Testing: I will check the application's performance on both technical and practical aspects.
5. Closure: Final section of the project where it will be finally launched, closed and presented.

<b>Code</b>	<b>Description of the task</b>
A	Define specific and clear goals of the project.
B	Research the state of the current situation.
C	Find the pros and cons that will influence the creation of the app.
D	Design the number of screens and their order.
E	Design how the screens will look (icons, graphs, buttons, colours, and so on).

<i>F</i>	Intensively learn about how to program with Visual Studio.
<i>G</i>	Design the general algorithms to follow.
<i>H</i>	Code and program the app.
<i>I</i>	Run the application once it has been programmed to test it for errors.
<i>J</i>	Try the application with people to see if it's comfortable and easy to use.
<i>K</i>	Fix and improve the application with the people's feedback and errors.
<i>L</i>	Launch final interactive version of the app so the VHIR can use it.
<i>M</i>	Presentation of this application.
<i>N</i>	End of the project.

Apart from the description of the tasks, it is very important to plan how long will each one take, which can be seen in the following table. The time is in time units, which are days. OP= Optimistic time, PbT = Probable time, PsT= Pessimistic time, Et = Expected time.

<b>Activity Code</b>	<b>OP (a)</b>	<b>PbT (m)</b>	<b>PsT(b)</b>	<b>Et</b>
<i>A</i>	1	1	2	1.2
<i>B</i>	1	2	3	2
<i>C</i>	1	1	2	1.2
<i>D</i>	2	2	3	2.2
<i>E</i>	2	3	4	3
<i>F</i>	3	5	7	5
<i>G</i>	1	2	3	2
<i>H</i>	6	8	10	8
<i>I</i>	7	8	9	8
<i>J</i>	8	9	10	9
<i>K</i>	2	2	3	2.2
<i>L</i>	3	5	7	5
<i>M</i>	3	4	5	4
<i>N</i>	-	-	-	-
	<b>39</b>	<b>51</b>	<b>67</b>	<b>59</b>

## 6.2 DEPENDENCY MATRIX

In this section I have related the different tasks depending on their temporal precedence. This and the previous table of times are fundamental to be able to do the PERT and the GANTT Chart diagrams.

<b>Code</b>	<b>Activity</b>	<b>Precedence</b>
<i>A</i>	Clear goals	-
<i>B</i>	Background and marketing analysis	A
<i>C</i>	SWOT analysis	B
<i>D</i>	Screens design	A
<i>E</i>	Interface design (icons, graphs, buttons, colours, and so on)	D
<i>F</i>	Learning about the coding language	C, E
<i>G</i>	Design algorithms	F
<i>H</i>	Program	G
<i>I</i>	Technical testing	H
<i>J</i>	Patient's testing	H
<i>K</i>	Improvements	I, J
<i>L</i>	Launch final interactive version	K
<i>M</i>	Presentation	L
<i>N</i>	Closure	M

### 6.3 PERT/CPM DIAGRAM AND CRITICAL PATH

The PERT diagram (Project Evaluation and Review Techniques) creates a dependency map between the different tasks of the project. These dependencies can be used to identify optimized workflows. Therefore, I have identified in purple the critical path, which is the optimum one that shows the minimum time for the project to be completed.

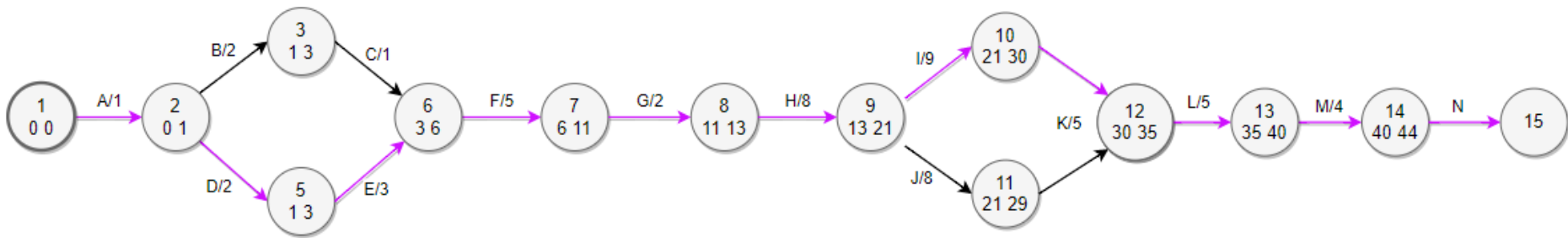


Figure 15. PERT diagram with the critical path of the project.

## 6.4 GANTT CHART

The GANTT chart is a bar plot that represents the activities, tasks or events displayed against time. The duration of each task has been expressed from start to finish, and we can also see which of them can be done in parallel.

# GANTT Chart

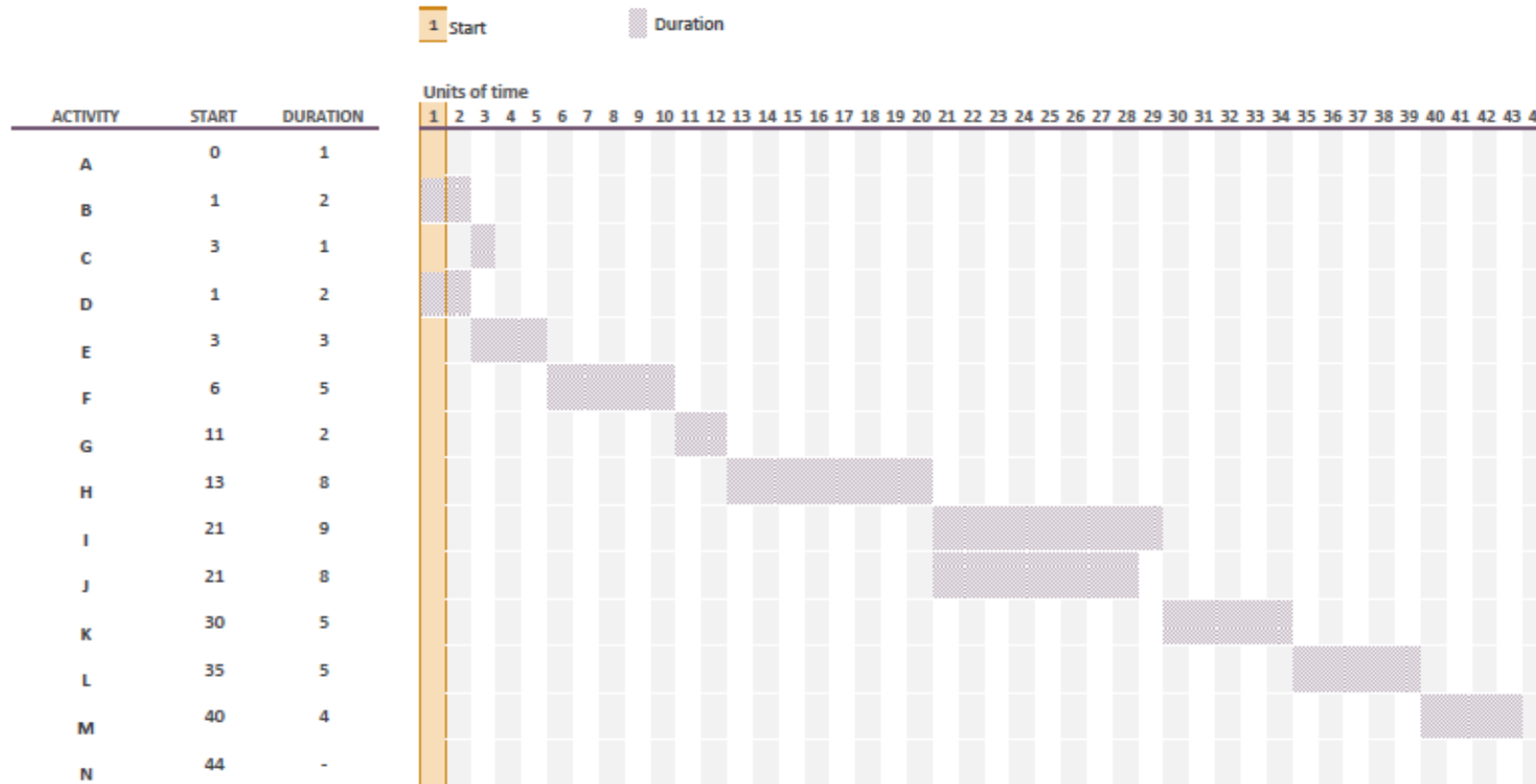


Figure 16. GANTT diagram of the project.

## 7. TECHNICAL FEASIBILITY

I did the SWOT diagram of the app to build it on our strengths, minimize risks and take the best advantage of chances for success, also facing the weaknesses I have.



Figure 17. SWOT analysis of the application.

The SWOT analysis is a very useful tool to properly develop the business strategy.

On the internal factors of the environment, the main strengths are that I will develop a simple app to which more features can be added easily afterwards, and it will be done with a free platform. It will be done for a specific research group from the VHIR following their necessities, and the app will work for both Android and iOS. Nevertheless, I still need training time in app development and the time is limited, also obtaining no profit from the app because it won't have a commercial use.

On the external factors that affect my project, the opportunities are that there is a huge community of app developers and there is a lot of open source codes, which makes it easier when learning to program. Also, I don't have competition because it's a very specific app, and it can help to the VHIR to further investigate migraine. On the other hand, the threats are that there might be cybersecurity issues (with



all the data protection), the coronavirus situation doesn't allow me to visit the hospital until it is more controlled, and also there might be new developments in the are of app programming that might leave behind mine.

## 8. ECONOMIC FEASIBILITY

To develop the application, the estimated time is of 44 days. Considering that a working shift consists of 5 hours per day, this would mean having 220 hours of work at the end. With an average salary of 7€/h for all the tasks (taking into account that I am only one person, so the easy and hard tasks get compensated and it is more logical to have an average salary for all the hours no matter which tasks they are at), that would mean a budget of 1540€ for the worked time.

In this graph the budget can be seen broken down for each task, considering that some tasks are done at the same time, so the budget is shared between them.

<b>Code</b>	<b>Duration (days)</b>	<b>Duration (hours)</b>	<b>Budget (7€/hour)</b>
A	1	5	35
B	2	10	70
C	1	5	35
D	2	10	70
E	3	15	104
F	5	25	175
G	2	10	70
H	8	40	280
I	8	40	280
J	9	45	315
K	2	10	70
L	5	25	175
M	4	20	140
N	-	-	-

Apart from the price per worked hour to develop the project, the programming of the app will be done with a free platform, so no more costs will be added.

Applications might take many hours and money to make, depending on their different features, API integration, back-end server, multi-language, design, and so on. The differences are huge, from 200 to 1500 hours to program, which translates in between 5000 and 50000€ depending on the hourly rate of the professional. Below there's a graph that shows the hourly rate of freelancers and companies depending on the chosen technology and operating system. [12]



Figure 18. Hourly rate of freelancers and companies.

## 9. REGULATIONS AND LEGAL ASPECTS

Usually when creating a commercial application, the owner has to be registered as a freelancer, he has to protect the intellectual and industrial property of the app through copyright and registration of the brand, also he has to take into account the image rights and the data protection (users' personal information), and many other points. [13]

However, in this case the application will be free to use, it will not generate any benefits and it will be limited to a group of users from the VHIR Unit, so it will not be displayed for all users. As a result, most of legal aspects and regulations that usually must be followed are much simpler.

Regarding the regulations in Spain, there is the "Real Decreto 1112/2018, de 7 de septiembre, sobre accesibilidad de los sitios web y aplicaciones para dispositivos móviles del sector público"[14] in the BOE ("Boletín Oficial del Estado"), which regulates some mandatory minimum accessibility requirements and establishes standards applicable to the design, construction, maintenance and updating of applications for mobile devices. In the case of this project, some of this regulations will not affect it as it is an internal application for the VHIR researchers, and it will not have any global accessibility problems with them.

When collecting users' personal information, the duty is to include a privacy policy. It is very important to be transparent with users about why the data is needed, so their personal information is kept secure.

Therefore, Mibrain(e) must follow up the law of GDPR (General Data Protection Regulation) (EU) 2016/67934 on data protection and privacy in the European Union (EU) and the European Economic Area (EEA). So, it must offer the customer protection of the privacy of his personal information, and on the registration page the user will have to accept the apps privacy policy to use it, that he can directly consult from clicking on the link on the registration page when first accessing the app. This privacy policy goes over the following points:

- Purpose of the collection of personal data.
- Name and contact information of the person responsible for the treatment.
- Legitimation for the treatment of this data.
- Recipients of the data.
- Period of conservation of said data.
- Information to the interested parties about the exercise of their rights and means through which they can exercise them.
- Request specific consent for each of the personal data that the app will access.
- Proper regulation of the data of minors.

A copy of the privacy policy that the users accept can be seen in the Annexes. The objective of this legal document is to offer legal guarantee and reliability for the users of Mibrain(e), regarding the processing of their personal data.

## **10. CONCLUSIONS AND FUTURE OVERVIEW**

This final chapter is dedicated to my personal conclusions of the project, drawn from the development of the entire process.

The objectives mentioned in the introductory chapters have been met, although I wished the epidemiological situation was better, so I could have tested the application with patients at the VHIR. The main one of them was to develop an Android and iOS application for mobile phones to study the attention of the patients through the cycle of the migraine. The developed application offers the option to perform two tasks (reaction time and object orientation tests), that can be used to correlate the attentional response of patients with their phase and state of the migraine attack. Although the results of the tasks are not uploaded to a server due to difficulties in synchronizing the project with the internal server from the VHIR, the data from the patients and the tests is saved internally on the phone, and the researchers can access it anytime they see the patient, or they can ask them to send it back.

Considering the technical development of the project, it was difficult to synthesize all the ideas and to fulfil the requirements once the direction was clear. Visual Studio has many advantages in the field of app development, but it was also very difficult to create some of the processes as there was not enough literature about it, so they had to be created from zero.

In addition to having learned to develop a project from start to finish, going through all its phases, I have acquired new experience on C#, Android, SQL and other necessary knowledge that has been required during all the different phases of the app development.

In conclusion, therefore, from this project I have seen how the engineering (a mobile application) and the biomedical fields correlate, and the great potential of technologies to help researchers, doctors, and many others from the medical discipline to improve people's everyday lives.

I am very grateful to the Research Laboratory in Headache and Neurological Pain of the VHIR (Vall d'Hebron Research Institute) for giving me this project, and especially to Xim Cerdà Company, the director of this work, who helped me with the programming of the application; to José M<sup>a</sup> Gómez Cama, my tutor, who reviewed the project and improved its quality; to Patricia Pozo Rosich, Head of the Headache Unit; and to my younger brother, who is studying IT Engineering and who solved many of my doubts.

## 11. REFERENCES

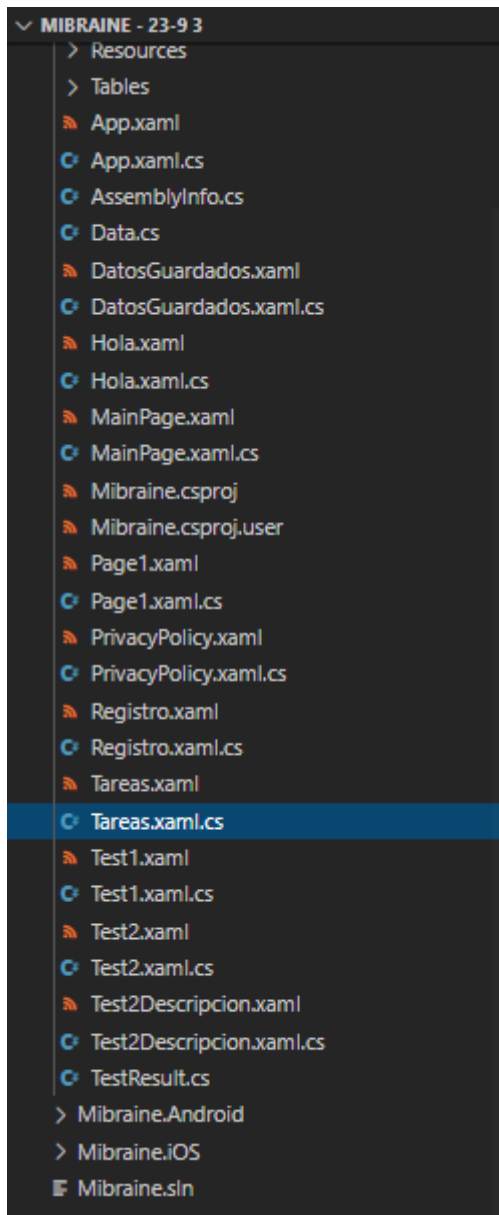
- [1] Headache disorders [Internet]. Who.int. 2020 [cited 5 May 2020]. Available from: <https://www.who.int/news-room/fact-sheets/detail/headache-disorders>
- [2] Puledda F, Messina R, Goadsby PJ. An update on migraine: current understanding and future directions. J Neurol. 2017;264(9):2031-2039. doi:10.1007/s00415-017-8434-y
- [3] Migraña [Internet]. Vall d'Hebron Barcelona Hospital Campus. 2020 [cited 5 May 2020]. Available from: <https://hospital.vallhebron.com/es/enfermedades/migrana>
- [4] Human Benchmark [Internet]. Humanbenchmark.com. 2020 [cited 6 May 2020]. Available from: <https://www.humanbenchmark.com/>
- [5] Allen J. Online Reaction Time Test [Internet]. Faculty.washington.edu. 2002 [cited 6 May 2020]. Available from: <https://faculty.washington.edu/chudler/java/redgreen.html>
- [6] Do you have the reflexes of an 18 year old? [Internet]. Justpark.com. 2020 [cited 6 May 2020]. Available from: <https://www.justpark.com/creative/reaction-time-test/>
- [7] Farmer, K & Cady, Roger & Bleiberg, Joseph & Reeves, D & Putnam, G & O'Quinn, S & Batenhorst, A. (2001). Sumatriptan nasal spray and cognitive function during migraine: results of an open-label study. Headache. 41. 377-84.
- [8] Huang L, Juan Dong H, Wang X, Wang Y, Xiao Z. Duration and frequency of migraines affect cognitive function: evidence from neuropsychological tests and event-related potentials. J Headache Pain. 2017;18(1):54. doi:10.1186/s10194-017-0758-6
- [9] S. O'Dea. Cell phone sales worldwide 2007-2020 | Statista [Internet]. 28 Feb 2020 [cited 7 May 2020]. Available from: <https://www.statista.com/statistics/263437/global-smartphone-sales-to-end-users-since-2007/>
- [10] Rouse M., Gillis A. and Silverthorne V. What is integrated development environment (IDE)? - Definition from WhatIs.com [Internet]. SearchSoftwareQuality. 2020 [cited 15 May 2020]. Available from: <https://searchsoftwarequality.techtarget.com/definition/integrated-development-environment>
- [11] .NET Development | Visual Studio [Internet]. Visual Studio. 2020 [cited 15 May 2020]. Available from: <https://visualstudio.microsoft.com/es/vs/features/net-development/>
- [12] ¿Cuánto cuesta crear una app? Precios por tipo, ciudad, tecnología... [Internet]. YeePLY. 2020 [cited 1 June 2020]. Available from: <https://www.yeePLY.com/blog/cuanto-cuesta-crear-una-app/>

- [13] Barberán Molina P. Aspectos jurídicos de las aplicaciones móviles (APPS). [Internet]. 2016 [cited 1 June 2020];. Available from: <https://www.acta.es/medios/informes/2016002.pdf>
- [14] «BOE» núm. 227, de 19 de septiembre de 2018, páginas 90533 a 90549 (17 págs.). [cited 1 June 2020]. Available from: [https://www.boe.es/diario\\_boe/txt.php?id=BOE-A-2018-12699](https://www.boe.es/diario_boe/txt.php?id=BOE-A-2018-12699)
- [15] Peng KP, May A. Migraine understood as a sensory threshold disease. *Pain*. 2019 Jul;160(7):1494-1501. doi: 10.1097/j.pain.0000000000001531. PMID: 31219950.
- [16] Shepherd AJ. A Review of Motion and Orientation Processing in Migraine. *Vision*. 2019; 3(2):12.
- [17] Amini Vishteh R, Mirzajani A, Jafarzadehpour E, Darvishpour S. Evaluation of Simple Visual Reaction Time of Different Colored Light Stimuli in Visually Normal Students. *Clin Optom (Auckl)*. 2019;11:167-171.
- [18] Wray SH, Mijović-Prelec D, Kosslyn SM. Visual processing in migraineurs. *Brain*. 1995 Feb;118 (Pt 1):25-35. doi: 10.1093/brain/118.1.25. PMID: 7895008.

## 12. ANNEXES

### 12.1 CODING

The overall scheme of Visual Studio with all the screens is the following, where we can see the overall programming of the app, and if there is anything in particular for Android or iOS (if not, the code is automatically adapted for both platforms from the main folder):



When initializing the application, the program has to check if there is any data saved in the internal memory of the app in the phone, so we know if it's a new user (and it has to go through the registration process), or if it's a user that already exists. When uninstalling the app from the patient's phone, this data is erased.

```

private async void Continuar_clicked(object sender, EventArgs e)
{
    if(File.Exists(Path.Combine(Environment.GetFolderPath(Environment.SpecialFolder.ApplicationData), "dataMibraine.dat")))
    {
        // Si en la memoria existen datos de registro, va directamente a la main page. Si no, va al registro (initialization)
        {
            await Navigation.PushAsync(new MainPage());
        }
    }
    else
    {
        await Navigation.PushAsync(new PrivacyPolicy());
    }
}
}

```

After the initialization, the privacy policy is shown:

```

Title="Política de Privacidad">
<ContentPage.Content>
<StackLayout>
<ScrollView VerticalScrollBarVisibility="Always">
<StackLayout>
<Label Text="POLÍTICA DE PRIVACIDAD APP MIBRAIN(E)"
VerticalOptions="Center"
HorizontalOptions="CenterAndExpand"
FontSize="Subtitle"
TextDecorations="Underline"
FontAttributes="Bold" />
<Label Text="Recogida y tratamiento de datos de carácter personal"
VerticalOptions="Center"
HorizontalOptions="StartAndExpand"
FontSize="Subtitle"
FontAttributes="Bold" />
<Label Text="Los datos de carácter personal son los que pueden ser utilizados para identificar a una persona o ponerse en contacto con"
VerticalOptions="Center"
HorizontalOptions="StartAndExpand" />
<Label Text="MIBRAIN(E) puede solicitar datos personales de usuarios al acceder a la aplicación para mejorar los productos y servicios"
VerticalOptions="Center"
HorizontalOptions="StartAndExpand" />
<Label Text="Estos son algunos ejemplos de las categorías de datos de carácter personal que MIBRAIN(E) puede recoger y la finalidad par"
VerticalOptions="Center"
HorizontalOptions="StartAndExpand" />
<Label Text="- Nombre, apellido, correo electrónico, sexo, y fecha de nacimiento."
VerticalOptions="Center"
HorizontalOptions="StartAndExpand" />
<Label Text="- Información relacionada con padecer migraña y factores de riesgo asociados."
VerticalOptions="Center"
HorizontalOptions="StartAndExpand" />
<Label Text="Propósito del tratamiento de datos de carácter personal"
VerticalOptions="Center"
HorizontalOptions="StartAndExpand" />

```

Only when this privacy policy is accepted, the registration page will pop up:



```

[XamlCompilation(XamlCompilationOptions.Compile)]
public partial class PrivacyPolicy : ContentPage
{
    public PrivacyPolicy()
    {
        InitializeComponent();
    }

    private async void Rechazo_Clicked(object sender, System.EventArgs e)
    {
        await Navigation.PopAsync();
    }

    private async void Acepto_Clicked(object sender, System.EventArgs e)
    {
        await Navigation.PushAsync(new Registro());
        Navigation.RemovePage(this);
    }
}

```

The registration will ask for some personal data of the patient, and it will save it in the internal folder of the app in the internal storage of the phone:

```

public Registro()
{
    InitializeComponent();
    // primero miraremos si hay datos de registro ya guardados en la app
    try {
        Stream stream = File.Open(Path.Combine(Environment.GetFolderPath(Environment.SpecialFolder.ApplicationData), "dataMibraine.dat"), FileMode.Open);
        BinaryFormatter formatter = new BinaryFormatter();
        data = (Data)formatter.Deserialize(stream);
        stream.Close();

        // si hay datos guardados, los mostraremos para que el usuario los pueda ver y modificar en caso necesario

        EntryName.Text = data.GetNombre();
        EntryEmail.Text = data.GetEmail();
        EntryBirth.Date = data.GetFecha();
        EntrySexo.SelectedIndex = data.GetGenero();
    }
    catch (FileNotFoundException){ // si no hay datos guardados es que el usuario no se ha registrado aún, no pasa nada
    }
    catch (Exception) // para cualquier otro error, ponemos un mensaje de error para que no pete todo
    {
        this.DisplayAlert("ERROR", "Se ha producido un error al cargar los datos", "Cerrar");
    }
}
}

```

We will check if the user has put all the necessary fields before confirming a successful registration:

```

void Entrar_Clicked(object sender, System.EventArgs e)
{
    if (EntryName.Text == null || EntryName.Text == "")
    {
        Show_Dialog("Nombre inválido");
    }

    else if (EntryEmail.Text == null || EntryEmail.Text == "")
    {
        Show_Dialog("Email inválido");
    }

    else if (EntrySexo.SelectedIndex == -1)
    {
        Show_Dialog("Seleccione un género");
    }

    else
    {
        try
        {
            Device.BeginInvokeOnMainThread(async () =>
            {
                var result = await this.DisplayAlert("Bienvenido!", "Registro de usuario completo", "Seguir", "Cancelar");
            }
        }
    }
}

```

On the main page, which has three buttons, it redirects the user to the different options: tests, modify profile, and view results and data. The first image corresponds to the layout of the page, coded through .XAML language, and the second image corresponds to the functionality of the page, which is coded in C#.

```

<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             xmlns:d="http://xamarin.com/schemas/2014/forms/design"
             xmlns:mc="http://schemas.openxmlformats.org/markup-compatibility/2006"
             mc:Ignorable="d"
             x:Class="Mibraine.MainPage"
             Title="MIBRAIN(E)">
    <StackLayout Margin="20,35,20,20">
        <Image Source="logo.jpg"
              HorizontalOptions="FillAndExpand"
              VerticalOptions="FillAndExpand" />
        <Button Text="Tareas" HorizontalOptions="Fill" Clicked="Tareas_clicked"/>
        <Button Text="Modificar datos personales" HorizontalOptions="Fill" Clicked="Datos_clicked"/>
        <Button Text="Mostrar datos guardados" HorizontalOptions="Fill" Clicked="Mostrar_clicked"/>
    </StackLayout>
</ContentPage>

```

```

namespace Mibraine
{
    [DesignTimeVisible(false)]
    public partial class MainPage : ContentPage
    {
        public MainPage()
        {
            InitializeComponent();
        }
        private async void Datos_clicked(object sender, EventArgs e)
        {
            await Navigation.PushAsync(new Registro());
        }

        private async void Tareas_clicked(object sender, EventArgs e)
        {
            await Navigation.PushAsync(new Tareas());
        }

        private async void Mostrar_clicked(object sender, EventArgs e)
        {
            await Navigation.PushAsync(new DatosGuardados());
        }
    }
}

```

Task 1: reaction time. When initializing the screen, we will check the user's data to save the results of the test with their personal information.

```

public Test1()
{
    InitializeComponent();

    clicked = false;

    try
    {
        Stream stream = File.Open(Path.Combine(Environment.GetFolderPath(Environment.SpecialFolder.ApplicationData),
        BinaryFormatter formatter = new BinaryFormatter();
        data = (Data)formatter.Deserialize(stream);
        stream.Close();
    }
    catch (Exception)
    {
        this.DisplayAlert("ERROR", "Se ha producido un error al cargar los datos", "Cerrar");
    }
}

```

Once we have checked the data, we will do the following process: when the user clicks on the 'Start' button, its background colour will change from gray to black. Then, in a random time unit from 2 to 6 seconds, it will change again, and once the user presses the button, the time will stop and the result will be displayed on the screen.

```
private void Empezar1Clicked(object sender, EventArgs e)
{
    Button btn = (Button)sender;
    if (btn.BackgroundColor.Equals(Color.Gray))
    {
        if (!clicked)
        {
            clicked = true;
            btn.Text = "";
            ChangeColor();
        }
        else
        {
            tiempo.Stop();
            clicked = false;
            btn.Text = "EMPEZAR!";

            this.DisplayAlert("Resultado", "Tiempo de Reacción: " + tiempo.ElapsedMilliseconds + " milisegundos", "Guardar resultado");

            data.SaveTest1(tiempo.ElapsedMilliseconds);
        }
    }
}
```

This shows in detail the 'ChangeColor' function that is used to change back the colour of the button from black to gray again in a random time.

```
async Task ChangeColor()
{
    Random num = new Random();
    btn.BackgroundColor = Color.Black;

    await Task.Delay(TimeSpan.FromSeconds(num.Next(2, 6)));

    btn.BackgroundColor = Color.Gray;
    tiempo = System.Diagnostics.Stopwatch.StartNew();
}
```

Task 2: visual search. After checking the user's data, the program will create a random number that will correspond to each of the 6 images of a line.

```
Random num = new Random();
int opt = num.Next(6);
btn = num.Next(1, 7);
string image = "";
clicked = false;
```

We will establish 5 cases with different orientations (from 5 to 45 degrees of difference with respect to the straight line), and one or none of them will be randomly displayed in one of the buttons of the screen.

```
switch (opt)
{
  case 1:
    image = "grados5.png";
    deg = 5;
    break;

  case 2:
    image = "grados10.png";
    deg = 10;
    break;

  case 3:
    image = "grados15.png";
    deg = 15;
    break;

  case 4:
    image = "grados30.png";
    deg = 30;
    break;

  case 5:
    image = "grados45.png";
    deg = 45;
    break;

  default:
    deg = 0;
    break;
}
```

The source of each of the buttons will be the image, and one of them will be randomly turned.

```

if (opt != 0)
{
    switch (btn)
    {
        case 1:
            btn1.Source = image;
            break;

        case 2:
            btn2.Source = image;
            break;

        case 3:
            btn3.Source = image;
            break;

        case 4:
            btn4.Source = image;
            break;

        case 5:
            btn5.Source = image;
            break;

        case 6:
            btn6.Source = image;
            break;

        default:
            this.DisplayAlert("ERROR!", "Error inesperado", "Cerrar");
            break;
    }
}

```

As said before, the test will wait for 3 seconds before it is considered finished. If the user has not pressed any buttons or if he has made a mistake, and he did not get the image with some certain degrees of difference, it will be considered as a failed test. On the other hand, if all the images had the same orientation or the user pressed on the right one, it will be considered a successful test. In both cases, the results will be saved in the internal storage of the phone.

```

tiempo = System.Diagnostics.Stopwatch.StartNew();
TimeEnded();

async Task TimeEnded()
{
    await Task.Delay(TimeSpan.FromSeconds(3));

    if (!clicked)
    {
        tiempo.Stop();

        if (deg != 0)
        {
            await this.DisplayAlert("Test fallido!", "Se ha acabado el tiempo", "Guardar resultado");
            result = "Fallo";
        }
        else
        {
            await this.DisplayAlert("Enhorabuena!", "Has acertado, todas las líneas eran iguales", "Guardar resultado");
            result = "Acierto";
        }

        data.SaveTest2(result, deg, 3000);
    }
}

```

```

private void BtnClicked(int numBtn)
{
    tiempo.Stop();
    clicked = true;

    Device.BeginInvokeOnMainThread(async () =>
    {
        if (deg == 0 || btn != numBtn)
        {
            await this.DisplayAlert("Test fallido!", "Has fallado\nInclinación: " + deg + "º", "Guardar resultado");
            result = "Fallo";
        }
        else
        {
            await this.DisplayAlert("Enhorabuena!", "Has acertado\nInclinación: " + deg + "º\n" +
                "Tiempo de Reacción: " + tiempo.ElapsedMilliseconds + " milisegundos", "Guardar resultado");
            result = "Acierto";
        }

        data.SaveTest2(result, deg, tiempo.ElapsedMilliseconds);
    });
}

```

Displaying the results in a graph: to be able to display the test answers in a graphical way, I installed different packages from Xamarin.Forms.

```

using System;
using System.IO;

using Xamarin.Forms;
using Xamarin.Forms.Xaml;
using System.Runtime.Serialization.Formatters.Binary;
using SkiaSharp;
using Microcharts;
using System.Collections.Generic;

```

From those packages, the proper functions were used, adding the corresponding entries for each of the results saved from the tests.

```

List<ChartEntry> entries1 = new List<ChartEntry>();
foreach (TestResult result in data.GetTest1())
{
    entries1.Add(new ChartEntry(result.GetTime()) { Label = result.GetDate().Substring(0,5), ValueLabel = result.GetTime() + " ns", Color = SKColors.Blue });
}
GraficoTest1.Chart = new LineChart()
{
    LineMode = LineMode.Straight,
    Label textSize = 45f,
    Entries = entries1
};

```

In the second graph, the right and wrong answers were also classified with different colours to be able to differentiate them. The maximum time is 300 ms (after which the test stops if you have not given a response to it by clicking one of the buttons), so it is the maximum value displayed. In the correct results, having a time of 300 ms will mean having the result right, as it means that no lines had a different orientation and the time just ran by.

```

List<ChartEntry> entries2 = new List<ChartEntry>();
foreach (TestResult result in data.GetTest2())
{
    if (result.GetResult().Equals("Acierto"))
        entries2.Add(new ChartEntry(result.GetTime()) { Label = result.GetDate().Substring(0, 5) + " - " + result.GetDegree() + "°", ValueLabel = result.GetTime() + " ns", Color = SKColors.Green });
    else
        entries2.Add(new ChartEntry(result.GetTime()) { Label = result.GetDate().Substring(0, 5) + " - " + result.GetDegree() + "°", ValueLabel = result.GetTime() + " ns", Color = SKColors.Red });
}
GraficoTest2.Chart = new LineChart()
{
    LineMode = LineMode.Straight,
    Label textSize = 45f,
    Entries = entries2
};

```

## 12.2 PRIVACY POLICY

### POLÍTICA DE PRIVACIDAD APP MIBRAIN(E)

#### Recogida y tratamiento de datos de carácter personal

Los datos de carácter personal son los que pueden ser utilizados para identificar a una persona o ponerse en contacto con ella.

MIBRAIN(E) puede solicitar datos personales de usuarios al acceder a la aplicación para mejorar los productos y servicios ofrecidos. Si no se facilitan esos datos personales, en muchos casos no podremos ofrecer los productos o servicios solicitados.

Estos son algunos ejemplos de las categorías de datos de carácter personal que MIBRAIN(E) puede recoger y la finalidad para los que puede llevar a cabo el tratamiento de estos datos.

- Nombre, apellido, correo electrónico, sexo, y fecha de nacimiento.
- Información relacionada con padecer migraña y factores de riesgo asociados.



## **Propósito del tratamiento de datos de carácter personal**

MIBRAIN(E) podrá utilizar los datos personales recabados para:

- Los datos de carácter personal recopilados permiten mantenerle informado acerca de las actualizaciones de software disponibles y los próximos eventos.
- También se utilizan los datos de carácter personal como ayuda para elaborar, perfeccionar, gestionar, proporcionar y mejorar los productos, servicios y contenidos de la aplicación, y con el propósito de evitar pérdidas y fraudes.
- Pueden utilizarse los datos de carácter personal para comprobar la identidad, colaborar en la identificación de usuarios y decidir los servicios apropiados.
- También se utilizan esos datos de carácter personal con propósitos internos, incluyendo el análisis de datos, con fines de investigación para el Grupo de Investigación en Cefalea y Dolor Neurológico del VHIR (Instituto de Recerca del hospital de Vall d'Hebron).

## **Recopilación y tratamiento de datos de carácter no personal**

MIBRAIN(E) también recopilará datos de un modo que, por sí mismos, no pueden ser asociados directamente a una persona determinada. Estos datos de carácter no personal se pueden recopilar, tratar, transferir y publicar con cualquier intención. Estos son algunos ejemplos de las clases de datos de carácter no personal que MIBRAIN(E) puede recopilar y los fines para los que se realiza su tratamiento:

- Datos tales como resultados de tests de diversos tipos (tiempo de reacción, orientación de objetos, etc.) para comprender mejor la conducta de los usuarios y mejorar la aplicación, a la vez que se progresa en la investigación para comprender la migraña en sus distintas fases.

Si juntamos datos de carácter no personal con datos personales, los datos mezclados serán tratados como datos personales mientras sigan estando combinados.

## **Divulgación a terceros**

Ocasionalmente MIBRAIN(E) puede facilitar determinados datos de carácter personal a investigadores asociados del VHIR que trabajen con nosotros para ayudarnos en la investigación. No se compartirán los datos con ningún tercero para sus propios fines.

### **Protección de datos de carácter personal**

MIBRAIN(E) garantizará la protección de los datos personales mediante cifrado durante el tránsito y, los alojados en instalaciones, con medidas de seguridad físicas.

MIBRAIN(E) garantizará la exactitud y la calidad de los datos personales, se conservarán durante el tiempo necesario para cumplir los fines para los que fueron recabados, salvo que la ley exija conservarlos durante más tiempo.

MIBRAIN(E) es consciente de la necesidad de establecer precauciones adicionales para preservar la privacidad y la seguridad de los menores que utilizan las aplicaciones y exigir consentimiento de sus progenitores en caso de que no tengan la edad mínima exigida por la legislación (en España, 14 años).

Si se han recopilado datos personales de un menor de 14 años, sin el consentimiento necesario, se debe eliminar esa información lo antes posible.