



INSTITUTO
UNIVERSITÁRIO
DE LISBOA

User Emotional Interaction Processor: A tool to support the development of GUIs through physiological user monitoring

Duarte Maria Stock da Cunha Santiago Pinto

Master's in Computer Science and Engineering

Supervisor:

Doctor José Luís Cardoso da Silva, Assistant Professor,
Iscte

Co-advisor:

Doctor Armando César Ferreira Lima, Assistant Professor, Iscte

September, 2021

User Emotional Interaction Processor: A tool to support the development of GUIs through physiological user monitoring

Duarte Maria Stock da Cunha Santiago Pinto

Master's in Computer Science and Engineering

Supervisor:

Doctor José Luís Cardoso da Silva, Assistant Professor,
Iscte

Co-advisor:

Doctor Armando César Ferreira Lima, Assistant Professor, Iscte

September, 2021

Acknowledgment

First of all, I would like to thank my family for having supported and motivated me throughout this long course of the thesis.

Secondly, I would like to thank my supervisors, Doctor José Luís da Silva and Doctor César Lima, for all the help and guidance they gave me for the completion of this dissertation and for all the patience they had to have for me.

And last but not least, I want to thank Miguel Rangel, Luís Matos Silva, and Pedro Freitas Rodrigues, for all the advice, teaching and help they gave me.

Resumo

Desde que os computadores entraram na vida diária dos humanos, a atividade entre o ecossistema humano e o digital tem aumentado. Este aumento estimula o desenvolvimento de interfaces humano-computador mais inteligentes e apelativas ao utilizador. No entanto, para testar estas interfaces, os meios de interação têm sido limitados, em grande parte restritos à interface convencional, a interface “manual”, onde é preciso *input* físico, onde os participantes que testam estas interface, usam um teclado, um rato ou um *touch screen*, e onde a comunicação dos participantes com os designers é necessária. Existe outro método, que será aplicado nesta dissertação, que não necessita de input físico dos participantes, que se denomina de *Affective Computing*.

Esta dissertação apresenta o desenvolvimento de uma ferramenta de suporte ao desenvolvimento de interfaces gráficas, baseada na monitorização de aspetos psicológicos e fisiológicos do utilizador (emoções e atenção), visando melhorar a experiência do utilizador final, com o objetivo último de melhorar o *design* da interface.

O desenvolvimento desta ferramenta será descrito. Os resultados, dados por *designers* de uma empresa de IT, sugerem que esta é útil, mas que a interface otimizada gerada pela mesma tem ainda algumas falhas. Estas falhas estão, principalmente, relacionadas com a ausência de consideração de um contexto geral no processo de geração da interface.

Palavras-chave: Interface de utilizador atenta; Interface de utilizador afetiva; Resposta fisiológica.

Abstract

Ever since computers have entered humans' daily lives, the activity between the human and the digital ecosystems has increased. This increase encourages the development of smarter and more user-friendly human-computer interfaces. However, to test these interfaces, the means of interaction have been limited, for the most part restricted to the conventional interface, the "manual" interface, where physical input is required, where participants who test these interfaces use a keyboard, mouse, or a touch screen, and where communication between participants and designers is required. There is another method, which will be applied in this dissertation, which does not require physical input from the participants, which is called Affective Computing.

This dissertation presents the development of a tool to support the development of graphical interfaces, based on the monitoring of psychological and physiological aspects of the user (emotions and attention), aiming to improve the experience of the end user, with the ultimate goal of improving the interface design.

The development of this tool will be described. The results, provided by designers from an IT company, suggest that the tool is useful but that the optimized interface generated by it still has some flaws. These flaws are mainly related to the lack of consideration of a general context in the interface generation process.

Keywords: Attentive User Interface; Affective User Interface; Physiological Response.

Index

Acknowledgment	i
Resumo.....	iii
Abstract	v
Index.....	vii
Table Index.....	ix
Figure Index	xi
List of Abbreviations and Acronyms	xiii
Chapter 1 – Introduction	14
1.1 Theme Framework	14
1.2 Motivation	15
1.3 Research question and objectives.....	15
1.4 Investigation method.....	16
1.5 Structure and organization of the dissertation.....	17
Chapter 2 – Related Work.....	18
2.1 Attentive User Interface (AUI)	18
2.2 Affective Computing.....	21
2.3 Techniques	23
2.3.1 Electroencephalography (EEG).....	23
2.3.2 Eye Tracking	26
2.3.3 Heart-rate Variability (HRV)	28
2.3.4 Electrodermal Activity (EDA)	29
2.3.5 Facial Expressions.....	30
2.4. Conclusions.....	33
Chapter 3 – Approach	36
3.1. Equipment.....	36
3.2. Architecture.....	37
3.3. Emotional State and Pixel Data Collection.....	39
3.4. Cognitive State and Pixel Data Collection.....	40
3.5. Database - PhpMyAdmin.....	41
3.6. Merging GUIs	45
Chapter 4 - Case Study and Evaluation.....	51
4.1. Design.....	51
4.1.1 Neuronal network reliability	51
4.1.2 Overlays interpretation.....	52
4.1.3 Tool usefulness.....	52
4.2. Experimental apparatus.....	53

4.3. Procedure.....	54
4.3.1 Pilot Tests.....	54
4.3.2 Collecting Data and Training the Neuronal Network	56
4.3.3 Testing the Neuronal Network and Generate the Optimal Interface.....	57
4.3.4 Designers Manually Construct the Optimal Interface.....	58
Chapter 5 - Results and discussions	63
5.1 Users.....	63
5.2 Overlays	63
5.3 Optimal Interface.....	66
5.4 Designers Results and Discussion	67
Chapter 6 – Conclusions	71
6.1. Main conclusions	71
6.2. Limitations	72
6.3. Future Work.....	73
References	74
Appendix	77
Appendix A – User’s questionnaire	77
Appendix B – Script to users with tasks for them to do.....	79
Appendix C – Clickmap overlay example	80
Appendix D – Emotionmap overlay example	81
Appendix E – Final result example – Optimal interface.....	82
Appendix F – Designer’s questionnaire	83
Appendix G – Designer’s script	90
Appendix H – Designers manual results	91
Appendix I – EEG Devices [21]	92
Appendix J – Designer’s Results	93
Appendix L - Non-inserted sections interface.....	101

Table Index

Table 2.1 – Classifying activities according to attentional state	26
Table 2.2 – Data collected from common webcam and AffectivaSDK	32
Table 2.3 – Literature Review works features	33
Table 3.1 – EEG devices features	37

Figure Index

Fig 2.1 eyeWINDOWS with zooming focus window [26]	19
Fig 2.2 Attentive Headphones [26]	20
Fig 2.3 PAUI heart rate monitor (left) and EEG (right) [26]	21
Fig 2.4 Two radio buttons presenting "their" emotion.	23
Fig 2.5 Various EEG devices.	24
Fig 2.6 Russell’s circumplex model of affect.	25
Fig 2.7 – User interruption preference	26
Fig 2.8 Tobii 4c	27
Fig 2.9 SEMI RED 250	27
Fig 2.10 Bitalino	29
Fig 2.11 Intel RealSense Camera	29
Fig 2.12 Electrodermal Activity Measurements	30
Fig 2.13 - Cognitive Heatmap	34
Fig 3.1 – General Diagram	38
Fig 3-2 – Database Model	41
Fig 3.3 - "Emotion" table	42
Fig 3.4 - "Clickmap" table	42
Fig 3.5 - Emotional States Process	42
Fig 3.6 - Cognitive States Process	43
Fig 3.7 - "Bitalinotobiimap" table	43
Fig 3.8 - Timeline	45
Fig 3.9 - Query Example	46
Fig 4.1 - ECG electrodes position [35]	52
Fig 4.2 - EDA electrodes position [36]	53
Fig 4.3 - Designer process part 1	58
Fig 4.4 - Designer process part 2	58
Fig 4.5 - Designer process part 3	58
Fig 5.1 - User 5 neuronal network results	63
Fig 5.2 - User 20 neuronal network results	63
Fig 5.3 - User 18 Emotion Map, Code Zone	64
Fig 5.4 - String present in the interface	65
Fig 5.5 - String present in the interface	65

List of Abbreviations and Acronyms

ACC - Accelerometer

AUI – Attentive User Interface

ECG - Electrocardiogram

ECoG - Electrocorticography

EDA – Electrodermal Activity

EEG - Electroencephalography

FACS - Facial Action Coding System

GSR – Galvanic Skin Response

GUI – Graphical User Interface

HCI – Human Computer Interaction

HF – High Frequency

HR – Heart Rate

HRV – Heart Rate Variability

LF – Low Frequency

MEG - Magnetoencephalography

PAUI – Physiological Attentive User Interface

PNS - Parasympathetic Nervous System

RMSSD - Root Mean Square of the Successive Differences

SCR - Skin Conductance Responses

SDNN - Standard Deviation of Normal to Normal

SNS - Sympathetic Nervous System

UI – User Interface

VLF - Very Low Frequency

Chapter 1 – Introduction

1.1 Theme Framework

Since computers have been created, the human and the digital ecosystems relationship has been increasing. The relationship between humans and computers, called Human-Computer Interaction (HCI), was from many-to-one, then one-to-one, and nowadays is from one-to-many. This increase encourages the development of smarter and more appealing Human-Computer Interfaces, so that this relationship is maintained over time. If an interface is not appealing or user friendly, the user will stop using it, breaking up the relationship. However, to test these interfaces, the means of interaction have been limited, for the most part restricted to the conventional interface, the “manual” interface, where people use a keyboard, a mouse, or the touch screen, and the conventional testing method, that requires communication between the testers and the designers. Science is studying methods to help understand human intentions without much need of physical input from humans [3].

To improve HCI, Vertegaal et al. [4] wrote about a framework for Attentive User Interface (AUI). It is through sensing, communication, augmentation, control and availability of the user that it is possible to optimize communication between Humans and Machines without a keyboard, a mouse or a touch screen. AUIs try to understand where the user is looking at, measuring the user's attention, and this measure may tell if the user is performing a given task or not.

To better understand human intentions without much need for physical input from humans, it is also needed to talk about Affective Computing. It uses human emotions to adapt an interface or an application to the users' mood, affective states and behavior. User emotion is estimated via a range of signals, including brain signals, heart signals, and eye-gaze. For example, Freitas et al. [5] developed a system that recognizes human emotions, named *EmoCam*, that has the potential to improve HCI via affective information.

Physiologically Attentive User Interface (PAUI) uses these emotional predictions in real time to change dynamically as a function of the emotional and attentional state of the user. It is a combination of Attentive User Interface and Affective Computing. These are interfaces that adjust depending on the user's monitoring, to optimize their performance. The user state (e.g., emotional, physical, mental), their characteristics and context are important aspects to be able to adapt the interface effectively. The goal is a better Human-Computer interaction, leading to potential benefits in terms of usability and user experience. [4]

1.2 Motivation

People can profit from a PAUI because it can help the user to accomplish a certain task through his emotions and attention, displaying information in a way that increases the effectiveness of the interaction [2]. Because of this, in this work, a tool to support the development of graphical interfaces will be developed, based on user monitoring (emotions and attention), aiming to improve the experience and usability of the end user, thereby improving the interface design.

With this tool, a designer can monitor a user, knowing where he looked and with what cognitive state, knowing where he clicked and with what facial expression, in different design versions of the same interface. Through this monitoring, the tool, in the end, merges the in different design versions of the same interface, cutting the parts where the user reacted positively, giving in the end an optimal interface. This tool is similar to a PAUI, since it aims to increase the effectiveness of the interaction. The difference between them is that a PAUI adapts in real time to the user's emotions, while the tool developed here needs the user data (e.g., clicks, emotional states, eye gaze, cognitive states) stored in a database, to then use these data to cut the parts where the user reacted more positively and paste them in a generated optimized interface, consisting only of the parts where the user reacted more positively.

1.3 Research question and objectives

This section presents the main research questions and objectives that were raised for the elaboration of this work.

After finishing the literature review, very few papers were found that had developed a tool to support the development of graphical interfaces, based on user monitoring (emotions and attention).

The research question that will be answered is:

- Is a tool semi-automatically suggesting, based on the physiological monitoring of users, an optimized Graphical User Interface useful for GUI designers?

In addition to the research question, it is also important to define the objectives for the elaboration of this project. That said, the objectives are:

- To develop a tool based on user monitoring (emotions and attention), aiming to improve the experience and usability of the end user, improving the interface design;
- To perform a user study to verify whether the solution developed brings improvements compared to the conventional method used (in particular by the IT company from where the designers are).

1.4 Investigation method

Three research methods were adopted for the development of this work. One is called Exploratory Research, the second Qualitative Research and the third Quantitative Research, the last two being part of the Methodology and research paradigms [7].

With the Exploratory Research, the theme and objectives were properly identified, and a literature review was conducted. This type of research is used when seeking an understanding about the general nature of the problem, alternative hypotheses and the relevant variables that need to be considered. It increases the knowledge of the researched problem, defines the focus and priorities of the study, and aims to understand what other researchers have done about the researched problem. It serves to hypothesize and uncover unknown characteristics on topics about which there is no knowledge [6]. In this initial research phase, scientific documents were analyzed to increase the knowledge on the subject under study.

Qualitative Research focuses on obtaining data through open ended communication or conversational communication. It is about “why” people think in a certain way and not only “what” people think. This method allows for in-depth and further probing and questioning of respondents based on their responses, where the interviewer/researcher also tries to understand their motivation and feelings. Understanding how the audience makes decisions can help derive conclusions in the research. It collects data at the sight of the experience when the participants are experiencing issues or problems. It is real-time data. Researchers usually collect many types of data rather than just relying on a data source. It helps solving complex problems by breaking it into small inferences that are easily read and understood. Being communicative, people can gain confidence in the researcher and the information they give the researcher is raw and unadulterated [8].

Quantitative Research is the systematic process of collecting observable and quantifiable data based on the observation of objective facts, phenomena and events that exist independently of the researcher. This method assumes that all data are quantifiable and can be translated into numbers, opinions and information to be classified and analyzed, usually using statistical methods. The objective of quantitative research is to quantify data and generalize sample results to the target population. These types of data are used to enrich and highlight the growth of this technology [9]. They will also be used when performing usability questionnaires, divided into five steps: (1) construction of the questionnaire; (2) pre-test of the questionnaire; (3) procedure and follow-up of the questionnaires; (4) data collection and processing; (5) analysis of results.

1.5 Structure and organization of the dissertation

This dissertation is organized in five chapters.

The first one introduced the investigation theme, the motivation and the objectives.

The second chapter refers to work related to the theme of this dissertation and its theoretical framework.

The third chapter presents the architecture of the tool and the technical approach used in its implementation, illustrating the various steps, explaining each tool/file developed for the major tool to work and explaining the reasons for the decisions that were made.

The fourth chapter explains the tests that were made, to answer the research questions.

The fifth chapter contains the evaluations that were made, the corresponding results and discussions from the tests made in chapter 4.

The sixth and final chapter presents the conclusions of this work as well as the recommendations, limitations, and future work.

Chapter 2 – Related Work

This section will present work that is related to the theme of this dissertation.

To find relevant work, a combination of keywords was searched, like Attentive User Interface (AUI), Affective Computing, Affective User Interface, Emotion, Attention, Brain-Computer Interface (BCI), Human-Computer Interaction (HCI), Adaptive User Interface, Emotion Recognition.

To deepen the various concepts presented below, some papers found will be from the 90s and early 2000s. However, for the application of the concepts, papers from 2010 upwards were searched and found. As there are papers that present an idea but then do not test it with a case study to draw conclusions, of these papers, only those with case studies were considered, as it was then known if the developed tools were tested and if they reached their purpose.

2.1 Attentive User Interface (AUI)

As Vertegaal et al. [26] state, AUI are interfaces that aim to support the user so that he/she is more efficient, knowing when he/she should or should not be interrupted. AUI optimizes their communication with the users, that is, the information that the interface shows to the user depends on their tasks. By estimating the user's attention for objects and people in their everyday environment, these interfaces can create patterns of interruption. They allow the users to move smoothly between the foreground and periphery of user activity, filling the gap between them. There are some characteristics and behaviors that define an attentive user interface. For example, by observing eye contact and presence, devices know if they are being focused on by the user; windows and icons can increase or decrease their size knowing if they are being looked at or not, as illustrated in Fig 2.1; menus and alerts are replaced by a negotiated turn taking process between users and devices.

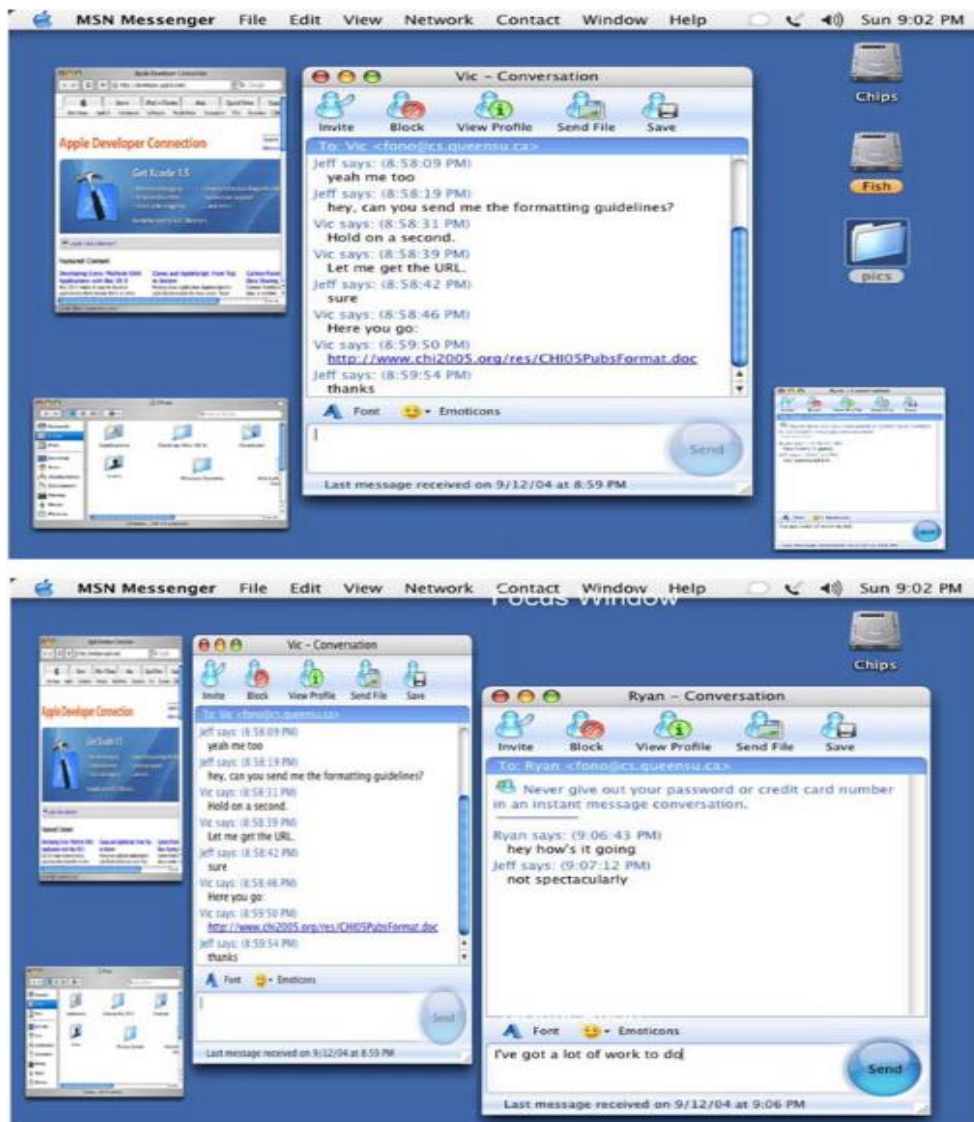


Fig 2.1 eyeWINDOWS with zooming focus window [26].

Attentive User Interface can also be used to enhance the user's cognitive processes, for example, by filtering out irrelevant information. This allows the user to focus on the most important things and to potentially be more efficient. For example, there are some noise cancelling headphones, like in Fig 2.2. They usually serve to block out any noise around the user, however, these headphones isolate the user from their co-workers. The eye contact sensor on the headphones lets you know when or not to turn on noise cancellation [26].



Fig 2.2 Attentive Headphones [26].

An attentive user interface has five key properties [26]:

1. *Sensing attention*: By tracking a user's physical proximity, body orientation, and eye fixes, interfaces can determine which device, person, or task the user is likely to participate in.
2. *Reasoning about attention*: Statistically modelling simple interactive user behavior, interfaces can estimate a prioritization of user tasks.
3. *Communication of attention*: Interfaces should provide information about users' attention to other people and devices. Communication systems should convey to whom or to what users are paying attention and if the user is available for communication.
4. *Gradual negotiation of turns*: Like turn taking, interfaces should determine the readiness of the user to interruption: (a) verify the priority of his request, (b) progressively signal that request through a peripheral channel, and (c) detect or acknowledge the request user before executing foreground.
5. *Augmentation of focus*: The ultimate goal of all AUIs is to increase the attention of their users. Similarly to the cocktail phenomenon, AUIs may, for example, broaden information focused upon by the user, and attenuate peripheral detail.

There is a type of attentive user interface, the PAUI. According to Vertegaal et al. [26], this type of AUI lets you know the user's attentional state. In addition to eye contact, electrical signals such as those on the electrocardiogram (ECG), from the heart, and the

electroencephalogram (EEG), from the brain, are examined, as illustrated in Fig 2.3. With those devices, specific attentional states can be determined that would be difficult to obtain from external data.



Fig 2.3 PAUI heart rate monitor (left) and EEG (right) [26].

The example that Vertegaal gives [26] is between a user and a cell phone. Their PAUI uses the ECG and the EEG to distinguish between four attentional states of the user, at rest, moving, thinking and busy. If the user is in a busy state, then the cell phone should not ring but only vibrate, in order not to disturb the user.

2.2 Affective Computing

As the name implies, affective computing is related with feelings, attitudes, and moods. It appeared in the 90s as a research area that aims at designing systems that can synthesize, recognize, and interpret emotions [27]. It has been growing in the last decade, with numerous developments in how people model, detect and understand a user's affective state [10] and detect human emotions to adjust applications to the user's behavior, moods, and affective state. Typically, measuring these emotions is difficult because of the lack of existing technology needed for emotion sensing [5].

In the work of K. Kusher et al. [19], the authors propose a visualization tool that represents excitement, both individually and in groups, for emotional self-awareness and group-level

awareness. The data visualized is obtained from smart wristbands worn by the users. The visualization uses animated glyphs to generate a real-time representation for each individual's excitement levels. The smart wristbands are a set of Microsoft Band 2 that measure the galvanic skin response (GSR) or electrodermal activity (EDA). Through the GSR they obtain the excitement levels that are being experienced. This data was streamed in real-time. If the wristbands were fit too tight, the GSR values could fluctuate, because this would lead to excessive sweating, so they measured accelerometer (ACC) values that corresponded to manifested excitement levels, and they would compare these values with the ones obtained from the Microsoft Band 2. If the user was exercising physical activity detected by the ACC and the wristband, an increase in the level of arousal would not be reported. They propose two different visualizations, one focusing on capturing real-time and previous excitement levels, and one more suited for capturing individual excitement peaks and the distribution of excitement in larger groups.

In the work of T. Baltrusaitis et al. [24], the authors present a toolbox that analyzes facial behavior. It detects facial landmarks, estimates the head pose, estimates the eye-gaze, and recognizes facial action units. It is capable of real-time performance and it only uses a webcam. Through the action units they can recognize facial expressions.

In the work of D. Cernea et al. [15], the authors introduce a solution to visualize user emotions while they are interacting with a specific UI component, like a button or a check box. Their goal was to help the user, providing him/her with information about the emotions felt by previous users. For example, there is a button that, if the user clicks, it would reproduce an unpleasant sound. This emotion would be recorded and then displayed below this button with a red color, warning a subsequent user that such button is associated with an unpleasant emotion. They adapt the original graphical user interface (GUI) in order to show which type of emotions the UI components are associated with. Positive emotions, highlighted with a green color, and negative emotions, highlighted with a red color, can be seen in Fig 2.4.



Fig 2.4 Two radio buttons presenting "their" emotion. The top bar includes over 50 EEG readings, from different users [15]

2.3 Techniques

Electroencephalography (EEG), Eye tracking, Heart-rate variability (HRV), Electrodermal Activity (EDA) and Facial Expressions have been used to reveal user's emotional and attentional states. In this section, we discuss the importance of each one of these techniques, how they are implemented, for what they are used and some experiments using them.

2.3.1 Electroencephalography (EEG)

According to Kerous et al. [13], people can use EEG to link brain activity with computers. Brain activity is detected in the form of weak electrical potentials on the surface of the human scalp and EEG devices (see Fig 2.5) are easier to use when compared to other alternatives, such as electrocorticography (ECoG) and magnetoencephalography (MEG). To obtain EEG signals, people place electrodes on the scalp and the signal has excellent temporal resolution, measurable in the range of milliseconds. To be recorded, the signal must cross the skull and scalp. Moreover, additional noise is introduced by the subject (eye movement, muscle activity) as well as by environmental disturbances (electronic devices, power line noise). This causes a lot of information to be diluted and lost in the background. To improve signal quality, conductive gel or saline is usually applied to the electrode tips, although specialized dry electrodes may also be used. This brain activity signal is constantly detectable and appears in different wave bands: alpha (7.5 to 12 Hz), beta (12 to 30 Hz), gamma (31 Hz and above), delta (0.5 to 3.5 Hz) and theta (3.5–7.5 Hz). EEG monitoring can be performed in the capacity of physiological recording, where it becomes an indicator of the user's emotional and attentional

state. According to Vertegaal et al. [26], with EEG people can get user states and motor activity. With this, we can separate four different user states of the person: at rest, moving, thinking and busy.



Fig 2.5 Various EEG devices.

In the work of D. Cernea et al. [15], the authors use this technique to acquire emotion data. Whenever the user interacts with an input GUI widget, the system records EEG brain signals through an EPOC headset, an EEG device. Signals are then interpreted as emotional states and stored with a reference to the GUI widget. Emotions captured this way can be viewed in the corresponding widget through emotion scents, real-time emotion perception or post-task analysis and decision support. The emotional states are based on Russell’s circumplex model of affect [28], represented in Fig 2.6, a two-dimensional model: valence (positive or negative) and arousal (excited or calm). Valence corresponds to a pleasure/displeasure scale, whereas arousal corresponds to a sleepiness/excitation scale. These scales define a circumplex space where it is possible to locate any ‘folk-psychology’ emotional term.

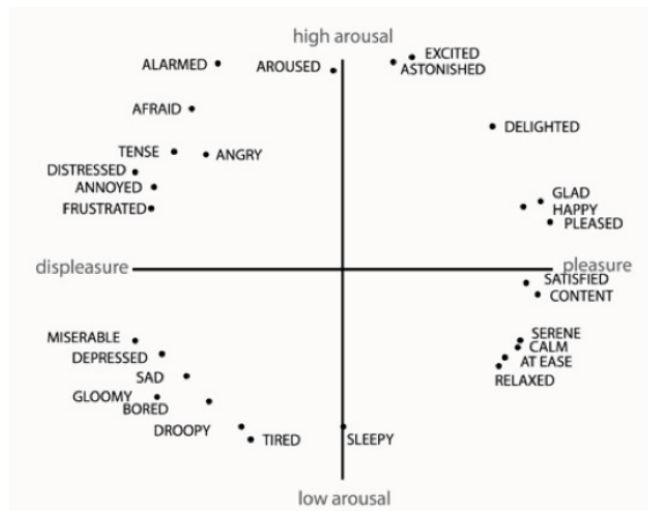


Fig 2.6 Russell's circumplex model of affect.

They use this model because the emotions captured by the EPOC headset are the same as those that are present on the model.

In the work of D. Chen and R. Vertegaal [16], the authors analyze the motor activity through an EEG to inspect four distinct states of the user and use them to predict the user availability for interruptions. These four states have interruption costs for both speech and motor activities, like a meeting or running, respectively. By using the user's physiological state and cost for interruptions (indicated by the user how he/she wants to be interrupted in each of the states, for each type of notification, whether it's an email, IM or calls, as seen in Fig 2.7), the system decides whether the user should be interrupted or not. As seen in Table 2.1, the first state of the system exhibits a very low degree of attention, where the user is not actively engaged with any task and can be interrupted without any interruption cost. The second state has a low interruption cost for audio-related interrupts but has high interruption costs for motor interrupts, such as messaging. For example, the phone is ringing but the emails and messages are set to vibrate. This state is associated with transit activities, such as walking or running. The third state is mental engagement while at rest, which results in high interruption costs for auditory

interruptions. The fourth and last state is of higher activity engagement, where any cost of interruption is high, and the user should not be disturbed.

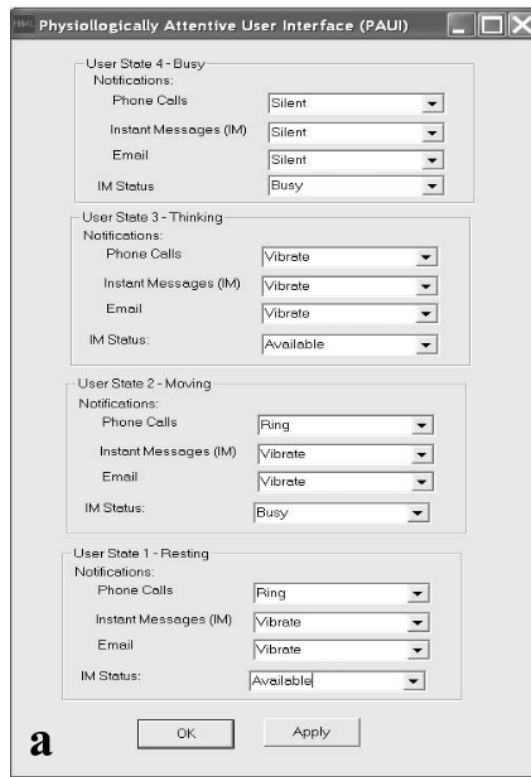


Fig 2.7 – User interruption preference

Table 2.1 - Classifying activities according to attentional state [16].

	Low Motor Activity (EEG)	High Motor Activity (EEG)
Low LF Power (ECG)	<p><i>User State 1</i></p> <p>-Low mental activity -At rest</p> <p><i>Candidate Activities</i> Pausing, Relaxation.</p>	<p><i>User State 2</i></p> <p>-Low mental activity -Sustained movement</p> <p><i>Candidate Activities</i> Moving.</p>
High LF Power (ECG)	<p><i>User State 3</i></p> <p>-High mental load -At rest</p> <p><i>Candidate Activities</i> Driving, Reading, Thinking</p>	<p><i>User State 4</i></p> <p>- High mental load - Sustained movement</p> <p><i>Candidate Activities</i> Meeting, Lecturing, Writing</p>

2.3.2 Eye Tracking

This technique is used to estimate where the user is looking at, to which devices the user is looking at, such as a mobile phone or a computer screen. This is called the eye gaze estimation.

It gathers and delivers information about the person's attention, but it cannot distinguish if the user is paying attention to something or is simply looking at something [4]. That is why people use other techniques like ECG, EDA, and EEG to know if the user is really engaged or not. This technique can also be used to study human emotions, as seen in [5], where pupil dilation is used as a measure of physiological arousal.

In the work of G. Singh et al. [3], the authors developed an emotional state estimator that provides input for runtime GUI redefinition in the context of life critical robot teleoperation. They use a device named Tobii 4c (see Fig 2.7) to estimate the user's emotional state and attentional state. They extract general fixation information, such as number of fixations, total time, total fixation duration, average fixation duration, repeated fixations, biggest fixation at, and extract the fixation map (containing information for each fixation), like the number of visits, start time, fixation duration, total fixation duration.



Fig 2.8 Tobii 4c

In the works of K. Yadati et al. [17], the authors use an eye-tracker named SMI RED 250 (see Fig 2.8), that records eye-movement data at a frequency of 250Hz. They propose a way of delivering advertisements considering the emotion and attention of the user. To do that they use an eye-tracker and a camera/webcam (facial expressions). If the user is seeing a video about cars and he/she is in a high arousal state, a car ad must be shown to the user. This way, the probability of the user remembering the ad is higher and his arousal state keeps up. They conclude that their framework had a good performance, meaning the users remembered the ads because it was related to the theme of the video they were seeing, and their arousal state kept up.



Fig 2.9 SEMI RED 250

In the work of V. Georges et al. [18], the authors show where the users were looking when they experienced specific cognitive and emotional states through gaze heatmaps, physiological heatmaps and facial expressions heatmaps. Beyond eye-tracking, they use electrodermal

activity, heart rate, vocal and visual cues, body gestures or facial expressions to infer different physiological and behavioral signals. They used a Tobii X-60 for gaze estimation. In their experiment, the participants were supposed to look at complex websites and perform some tasks to trigger different levels of cognitive load. They concluded that it is “possible to visually analyze users' various emotional and cognitive states for specific areas of a given interface”.

2.3.3 Heart-rate Variability (HRV)

With an electrocardiogram (ECG), people can get people's heart-rate variability (HRV) and heart rate (HR) that can be used to estimate a user's emotional state. According to Georges et al. [18], a user's heart rate signal can be associated with a change in cognitive load or emotional load. With an ECG, people can also estimate the user's attentional state, by examining the electrical signals from the ECG [27].

According to Chen and Vertegaal [16], the heart emits the highest electrical activity of all organs of the body, providing robust physiological data of the user's load levels. The sympathetic nervous system (SNS) and the parasympathetic nervous system (PNS) influence the electrocardiogram signal (ECG) because both components of the nervous system regulate the heart. When a person is in a high stress level, the SNS acts to increase the heart rate. The PNS counters the effect of the SNS by decreasing the heart rate. It is the PNS that regulates our heart in normal situations. The heart-rate variability (HRV) is calculated using the standard deviation of the heart rate data. The authors objective was to estimate the user's attentional state using the heart-rate variability signal and brain activity (EEG), to see if the user could be interrupted or not, depending on the user state. There were four distinct user states. They measured the Low Frequency (LF) power and if the Low Frequency (LF) was low, then the user was considered to be in state 1 (low mental activity, at rest) or state 2 (low mental activity, sustained movement), depending on the EEG signal. If the Low Frequency (LF) was high, then the user was considered to be in state 3 (high mental load, at rest) or state 4 (high mental load sustained movement), depending on the EEG signal. They used the EEG signals for motor activity. They applied this automated regulation of notifications in a mobile phone.

In the work of G. Singh et al. [3], the authors use a device named Bitalino (see Fig 2.9) to assess the heart rate (HR), heart-rate variability (HRV), the Standard Deviation of Normal to Normal (SDNN), the Root Mean Square of the Successive Differences (RMSSD), the Very Low Frequency (VLF), the Low Frequency (LF) and the High Frequency (HF). All these measures are related to the user's heart. With these features the authors were able to estimate the user's emotion and attention.

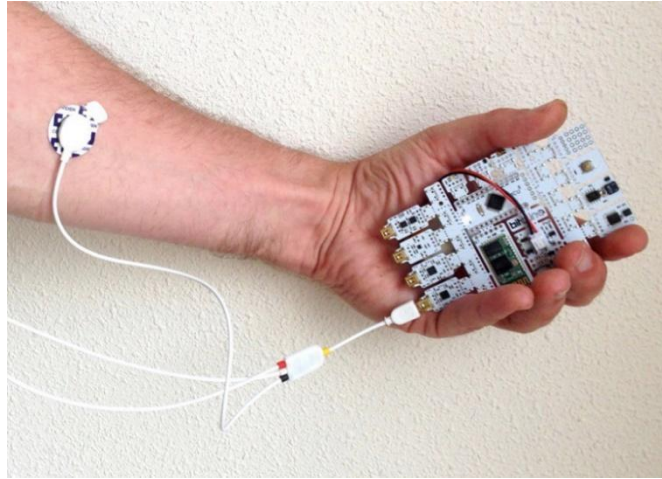


Fig 2.10 Bitalino

In the work of D. Freitas et al. [5], the authors only use a camera, named Intel RealSense (see Fig 2.10) to detect the heart rate (HR). To detect it, they use the RealSense SDK that comes with the camera Intel RealSense and it detects color changes and uses them to see if a pulse happened. This method is not the best because, as they observed, “The heart rate value is very unstable and sensitive to light changes;” and “(...) heart rate estimations can be done when the head- to-camera distance is approximately 30 cm to 60 cm”. Their objective was to create a panel that estimated the user's emotions and attention and use it in a game, to change its difficulty or its events, for example.



Fig 2.11 Intel RealSense Camera

2.3.4 Electrodermal Activity (EDA)

According to the statements of Kucher et al. [19] based on [20]– [22], over the decades, electrodermal activity (EDA) and galvanic skin response (GSR) have shown promising results

in terms of correlation with the user's arousal levels. In this technique, two electrodes measure the conductivity of a user's skin. Conductivity is affected by the moisture that is generated by the eccrine sweat glands, which can be found on our hands and feet, and are more sensitive to alter as a result of changes in emotional arousal. Activation of these sweat glands is not under a person's conscious control and can be linked especially to a high level of arousal and cognitive workload. To get the EDA signals, we can use various devices.

In V. Georges et al. [18], the authors used a device named Biopac MP150 (see Fig 2.11) to get electrocardiogram (ECG) signals and electrodermal activity (EDA) signals. They state "EDA measures the activity of the eccrine sweat glands and has been shown to be correlated to arousal. It can be used to measure emotions and cognitive load." The objective of this paper was to present a tool that describes users' physiological and behavior signals while interacting with a system. The tool shows where the person was looking when experiencing a certain emotion and cognition. They concluded that for a specific area in an interface it is possible to analyze the user's emotional and cognitive states.



Fig 2.12 Electrodermal Activity Measurements

2.3.5 Facial Expressions

According to McDaniel and Si [10], facial expressions are important indices of people's emotions, attitudes, and intentions. When a person is welcoming another person, to show a positive attitude, normally the person smiles. Arif and Grap [27] state something similar. They say that facial expressions are a valuable behavioral measure for the study of emotion, cognitive

processes, and social interactions. Lim et al. [10] gives an example. When a student answers a question incorrectly, a sad expression will be displayed.

Facial expressions are measured according to the Facial Action Coding System (FACS), invented by Ekman and Friesen [3]. There are basic elements of FACS named action units (AUs). One single AU represents movement of a single muscle or group of muscles [2]. According to Baltrusaitis et al. [24], there are not many tools available that do AU recognition, but there are available commercial systems that perform AU recognition, amongst other functionalities, like Affdex (from Affectiva). The drawback of these commercial systems is unknown algorithms, unknown training data and the product may be discontinued, like happened with FACET.

But there are some problems in collecting facial expressions to estimate people's emotions. Ward and Marsden [29] wrote that some facial expressions show reactions not apparent in the psychophysiology data and vice versa. They stated that some people show higher facial expressivity and others little, but that does not mean that the person that is showing higher facial expressivity is experiencing stronger emotions, or the person with little facial expressivity is experiencing weaker emotions. That is why people use other techniques like ECG, EEG, EDA, Eye trackers, to be more precise about what emotion the person is experiencing.

Freitas et al. [5] states that the latest approaches to collecting facial expressions to estimate a person's emotion just use cameras, computer vision and machine learning. In this project, they use a camera and an SDK, Intel RealSense Camera and AffectivaSDK, respectively. These tools can capture eight different emotions (anger, contempt, disgust, engagement, fear, joy, sadness, and surprise), five facial expressions and fourteen hand gestures. The AffectivaSDK can detect twenty facial expressions and detect the same emotions as the Intel RealSense Camera and AffectivaSDK combined. Their goal was to develop a software toolkit, named EmoCam Panel, that streamed the affective data to video games developed in Unity3D, allowing game developers to include emotional intelligence in their applications.

In the work of G. Singh et al. [3], they also use the AffectivaSDK and a common camera to detect human emotions. The SDK processes images frames and detects emotions from faces present in those frames. With this, they could extract 9 facial emotions (joy, fear, disgust, sadness, anger, surprise, contempt, valence, and engagement), 21 facial expressions and facial orientation information, as can be seen in Table 2.2.

Table 2.2 - Data collected from common webcam and AffectivaSDK [3]

Camera	Emotions	Joy
		Fear
		Disgust
		Dadness
		Anger
		Suprise
		Contempt
		Valence
		Engagement
	Expressions	Smile
		Inner Brow Rise
		Brow Rise
		Brow Furrow
		Nose Wrinkle
		Upper Lip Raise
		Lip Corner Depressor
		Chin Raise
		Lip Pucker
		Lip Press
		Lip Suck
		Mouth Open
		Cheek Raise
		Dimplier
		Eye Widen
		Jaw Drop
		Lip Tighten
		Lip Stretch
		Smirk
		Eye Closure
Attention		
	Face Orientation	Pan
		Tilt
		Yaw

In the work of V. George et al. [18], to detect emotions through facial expressions, they use a software named FaceReader 6, that detects seven discrete emotions (happy, sad, angry, surprised, scared, disgusted, and neutral), through probability. The objective of this paper was to present a tool that describes users' physiological and behavior signals while interacting with a system. The tool shows where the person was looking when experiencing a certain emotion and cognition.

In the work of K. Yadati et al. [17], to detect emotions through facial expressions, they use an “out of the box” emotion analyzer, named eMotion. It gives continuous probability scores on the chosen canonical emotions. It detects six canonical emotions (neutral, happy, surprise, anger, sad, fear and disgust). They propose a way of delivering advertisements considering the emotion and attention of the user. To do that they use an eye-tracker and a camera/webcam (facial expressions). The authors see what kind of video the user is watching and, if it is in a high arousal state, an ad related to the theme of the video interrupts the main video. They concluded that with this technique, the probability of the user remembering the ad is greater and that, after the ad, the high arousal status kept up.

In the work of D. Cernea et al. [15], to detect facial expressions, they do not use a camera. They use the Emotiv EPOC EEG headset that has an EPOC framework. The device has 14 electrodes to capture the electrical signals sent by the brain. This EPOC framework can interpret these signals as facial expressions, cognitive commands, and emotional states. However, the focus of their researcher does not care about detection techniques, but they state, “While the framework itself is proprietary, previous research suggests that the detection of facial expressions and emotional states through the device framework is reasonably accurate”.

2.4. Conclusions

After finishing the literature review, very few papers were found that have done a tool to support the development of graphical interfaces, based on user monitoring (emotions and attention). A table was made to synthesize all the literature reviews that have case studies, Table 2.3.

Table 2.3 - Literature review works features

Reference	EEG	Eye Tracking	ECG	EDA	Facial Expressions	Considers		Optimization			Task Support	GUI DEV
						Emotions	Attention	Interruptions	Visualization	Interaction		
[5]	NO	NO	NO	NO	YES	YES	NO	NO	NO	YES	NO	NO
[19]	NO	NO	NO	YES	NO	YES	NO	NO	NO	NO	NO	NO
[16]	YES	NO	YES	NO	NO	YES	NO	YES	NO	NO	YES	NO
[3]	YES	YES	YES	YES	YES	YES	YES	NO	YES	YES	YES	NO
[17]	NO	YES	NO	NO	YES	YES	YES	YES	NO	YES	NO	NO
[18]	NO	YES	YES	YES	YES	YES	YES	NO	YES	NO	NO	NO
[31]	NO	YES	NO	YES	YES	YES	YES	NO	YES	YES	YES	NO
[32]	NO	YES	NO	NO	YES	YES	YES	NO	YES	YES	YES	NO
[30]	YES	YES	NO	YES	YES	YES	YES	NO	YES	YES	NO	NO
[37]	YES	YES	NO	YES	YES	YES	YES	NO	YES	YES	NO	NO

Many papers described above are related only to task support (interruptions, helping users do some task) or to visualize where the user looked and with what cognitive state, but no paper

was found that takes this data and tries to improve the interface by building a new one, developing a new GUI.

However, three papers similar to the work presented in this thesis were found. In the works of V. Georges et al. [18][30], F. Courtemanche et al. [37], the authors developed a UX evaluation tool that measures users' physiological and behavioral signals. These measures are then shown on a heatmap, which is composed of clouds that represents the number of times the user has been looking at a certain area, with a more loaded cloud being one of the places where the user looked more. Each one of the clouds has a color: green, yellow or red. Red represents negative valence, yellow represents positive valence and green represents cognitive load (seen in Fig 2.12).

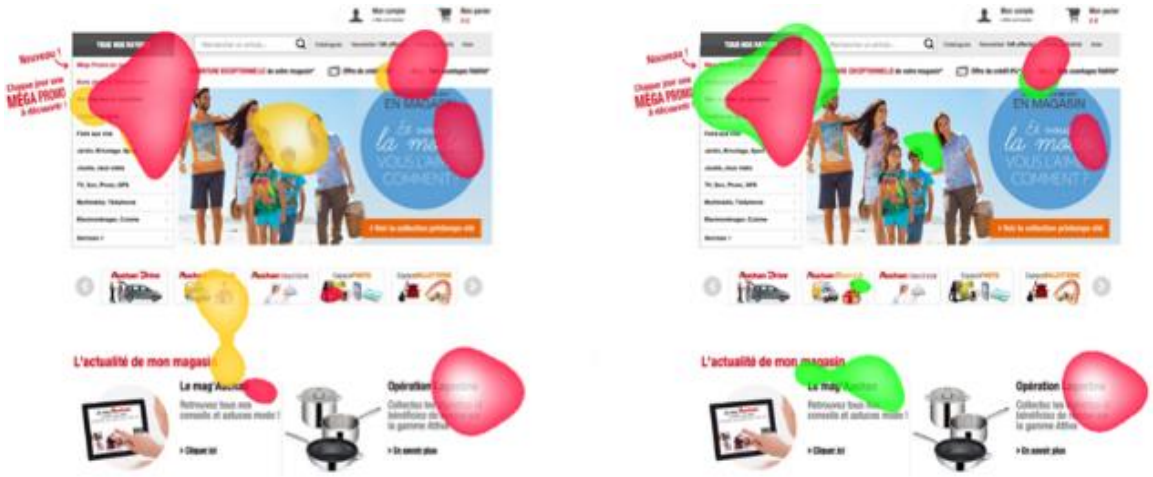


Fig 2.13 - Cognitive Heatmap

In this dissertation, the user's clicks, his/her facial expressions, eye gaze and cognitive states will all be monitored and stored in a database. Similar to the work done by V. Georges et al. [18][30], F. Courtemanche et al. [37], two overlays are generated, one with the clicks associated with the facial expressions of the user and the other with eye gaze associated with the cognitive state of the user. Different from the work related to heatmaps, the tool developed in this dissertation, uses the data stored in the database to build a new optimized interface, represented by the parts (from different GUI versions) where the user reacted more positively, aiming to improve the experience and usability of the end user, improving the interface design process.

Chapter 3 – Approach

In this chapter, the process of creating the tool is covered, from its architecture but also how the interactions between the tool and the user work. It will also be explained what equipment was used, why and how the EmoCam Panel works so that it detects facial expressions, to then estimate the users' emotional state, and finally, why BITalino was used, to collect EEG, ECG, and EDA data, also to estimate the users' cognitive state.

3.1. Equipment

After reading the related work, several devices were considered to read the physiological signals of the users and to see where the users were looking, but in the end, BITalino was used to read the physiological signals of users and Tobii 4c as an eye tracker. The reason why these devices were used will be explained below.

The device chosen to read EEG signals in this experiment is BITalino. One of the main reasons this device was selected is related to its accessibility. The other reason is that this device not only reads EEG signals, but also reads cardiac activity, related to ECG signals, and can read EDA signals. The last reason is because an EEG is not a cheap device (Appendix I).

As BITalino also reads ECG signals and it was already accessible, it was also used to read these signals. In the works of C. Martin et al. [31], the authors compare the two ECG devices found in the literature review, the Biopac and the BITalino. The idea of the experiment was to use three different types of electrodes (gel, paper, dry) and collect ECG signals with each type of electrode and then compare those signals. The classical acquisition methods used in clinical, or research studies, typically recur to gelled silver/silver chloride (Ag/AgCl) electrodes. As the Biopac has restricted operations and the electrodes used for data acquisition were always the pre-gelled Ag/AgCl, experimenting new customized solutions could damage the device, so they used BITalino, which gave them higher control over the system to try different experimental

setups, as the 3 types of electrodes. When both devices used electrodes with gel, it obtained the smallest SNR difference between devices, therefore its performance is very similar. The higher dispersion obtained was when BITalino’s electrodes were made of paper, due to higher noise presence in the signals. They also saw that the signals obtained from all three types of electrodes retain much of the waveform morphology. They concluded that “the heartbeat waveforms measured through the proposed approach are nearly identical to those obtained with the gold standard equipment”. For this reason and because BITalino is used to read EEG signals, BITalino was chosen, with gel electrodes.

Once again, the device that will read the EDA signals is BiTalino. Of the various papers read, several devices for reading EDA signals were used, all achieving similar performance to each other. For example, Microsoft Band 2, Empatica E4 and Biopac MP150. Since BiTalino will already be used to read both EEG and ECG signals, it was also chosen to read EDA signals.

The eye tracker that is going to be used in this dissertation is Tobii 4c. Yadati et al. [17] used an eye tracker named SMI RED 250, however this eye tracker is no longer sold [25] and the Tobii 4c device was already available.

The camera that will be used in this project is the camera that is integrated in the laptop. To detect facial expressions, the EmoCam program [5], that uses the Affectiva SDK, will be used, because this SDK, of all the SDKs found in the literature review, is the one that can read more emotions and expressions, as seen in table 3.1.

Table 3.1 - SDK for facial expressions

Name	Emotions	Expressions
Affectiva SDK [5]	8	20
Affectiva SDK [3]	9	21
FaceReader 6	7	-
eMotion	6	-

3.2. Architecture

The tool produced in this dissertation aims to help interface developers or designers to develop interfaces with a better design, a better user experience and a better usability, monitoring the user's emotional and attentional states.

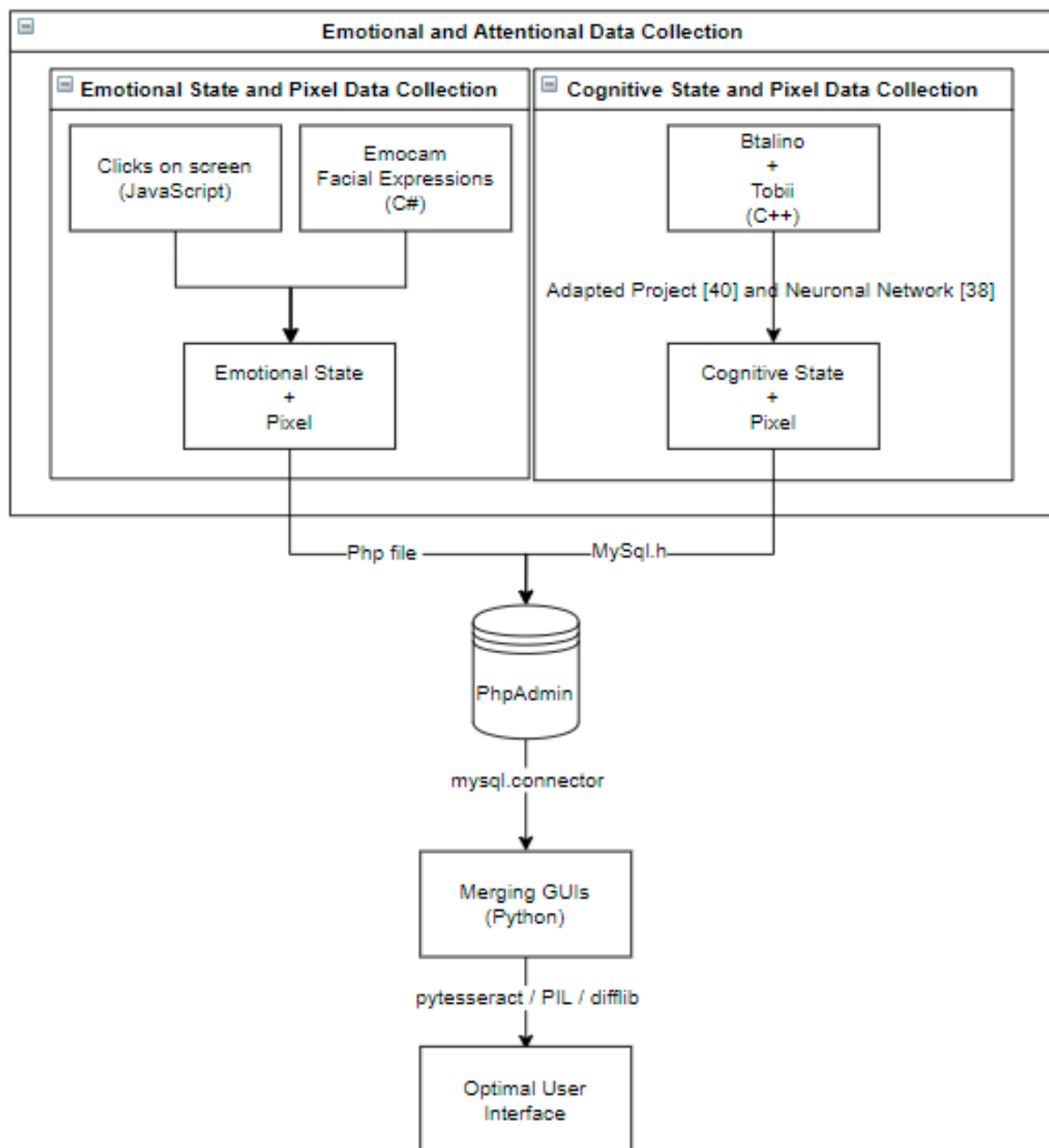


Fig 3.1 – General Diagram

Looking at Figure 3.1, it shows how the whole architecture was thought. In the beginning, the user opens three HTML interfaces (different design versions of the same). There are three interfaces because it was thought from the beginning to merge three interfaces. If it is to merge more or less interfaces, the code that merges those interfaces has to be changed. After opening the interfaces, the user's emotional and attentional data are collected. This tool is divided into two processes that are run in parallel. The first process, named “Emotional State and Pixel Data Collection” uses the EmoCam program to combine the emotional states (positive, neutral, negative), estimated by the facial expressions, with the user's clicks on the screen. The second process, named “Cognitive State and Pixel Data Collection”, uses the BITalino and the Tobii 4c to combine three cognitive states (rest, workload, and intense workload) with the user's eye

gaze. Then, all this data is stored in a *PhpAdmin* database to be used by a python file named “*ProcessImage.py*” that gives a merge between the three interfaces (Merging GUIs, Fig 3.1), where the final interface is the set of the parts where the user had no negative emotional or cognitive states.

Each part of the processes described above, “Emotional State and Pixel Data Collection”, “Cognitive State and Pixel Data Collection”, “Database (*PhpAdmin*)” and “Merging GUIs” will now be explained in more detail.

3.3. Emotional State and Pixel Data Collection

A tool named EmoCam was used to estimate users’ emotional states. Emocam is an existing tool that can capture eight different emotions (anger, contempt, disgust, engagement, fear, joy, sadness and surprise), five facial expressions and fourteen hand gestures. The goal of the authors was to develop a software toolkit that streamed the affective data to video games developed in Unity3D, allowing game developers to include emotional intelligence in their applications. This tool was discussed earlier in section 2. In this dissertation, the tool was adapted to only estimate users’ emotional states through facial expressions and leave hand gestures and the Intel RealSense related part aside. To estimate users’ emotional states through facial expressions, this software uses the Affectiva SDK, the one that detects more facial expressions, seen in Table 3.1.

So, while the user is navigating the interface without clicking on anything, the EmoCam is detecting emotions, however those emotions are not being saved in the *PhpAdmin* database. It only saves the user’s emotional state and pixel when he/she clicks in the html interface, because it is supposed to collect clicks associated with emotional states (the emotional state when the user does not click was not considered because it was not the objective of this process). The emotional states that are stored in the database are three, positive, neutral, or negative. These emotional states are a generalization of the eight different emotions that EmoCam can detect. For example, if EmoCam detected that the user was angry, this is a negative emotional state, but if the user on any other click was sad, this is also a negative emotional state. So, in both clicks, the emotional state saved on the database it’s negative. To save the pixels that were clicked in the interface on the database, a php file and a JavaScript file were used. The JavaScript file had a function to determine the pixel (x,y) that was clicked and the php file saves that pixel into the clickmap table. To save the three type of emotion on the database, a C# function is used. When the emotion is positive, the value 1 is stored, when it’s neutral the value 0 is stored, when the emotion is negative, the value -1 is stored. (See Fig 3.3)

3.4. Cognitive State and Pixel Data Collection

To estimate the cognitive states of the user, a device named BITalino was used. BITalino is a low-cost, open-source, single-board computer designed for education, prototype development and biomedical research. It was designed to be suitable for the acquisition of physiological signals. BITalino kits are equipped with sensors that are very useful to estimate user emotions. It comes with sensors for electromyography (EMG), electrocardiography (ECG), electrodermal activity (EDA) and electroencephalography (EEG). For the recording of physiological signals, it is possible to use BITalino with the OpenSignal software or with an API. APIs are available for the following platforms and programming languages: Python, Android, Matlab, Arduino, Bonsai Library, C++, Objective-C, LabView, Java, Raspberry Pi and Unity.

To get the cognitive states from the BITalino, the EEG, ECG and EDA signals were used and an already made project by Gagan [40] was also used. In the Gagan project, the author estimated three cognitive states of the user: rest, workload and stress; and also had the eye gaze of the user using the Tobii 4c device. In this dissertation, the three cognitive states that were estimated are rest, workload, and intense workload. These were the three cognitive states considered because the idea was to understand where, in an interface, the user reacted negatively, that is, with greater cognitive effort. An interface should be something easy and accessible to use, which does not force the user to increase his/her workload, except for some special cases like game interfaces or interfaces that have a video. To estimate these cognitive states, an adapted neuronal network made by António Tavares was used [38]. It was necessary to train this neural network. Then these estimations and eye gaze values were inserted in a table from the *PhpAdmin*, the “*bitalinotobiimap*” table. The rest state was inserted with value 0, the intense workload state was inserted with value 1 and the workload state was inserted with value 2.

For BITalino, to capture the EEG, ECG and EDA signals, the user needs to use electrodes in various positions, depending on the signal that is to be captured. Then, these electrodes are connected to the sensor that corresponds to the signal to be captured, through a cable, and this sensor is connected to the main board, also through a cable. BITalino is connected via Bluetooth to the computer.

To capture the eye gaze of the user, the Tobii 4c was used and, in the C++ program, made by Gagan, the Tobii gives the coordinates to where the user is looking at. However, when an interface with a scroll down option is being used, the Tobii does not work as it was intended, because if a user is using an interface and it scrolls down, the section of the interface to where

the user is looking at, passes the height of the screen and Tobii 4c only gives the coordinates of the dimensions of the screen monitor. The solution found was to make a JavaScript function that gives how many pixels the user scrolled (window.scrollY) and then it saves it to the *phpAdmin* database in a table called “*scrolloffset*”. So, every time the user scrolls, the pixels that were scrolled are saved on the table. At the same time, the eye gaze is being saved in the “*bitalinotobiimap*” table, but the y coordinate and the value of the scroll are added. It is always the last scrolled values that are fetched from the “*scrolloffset*” table.

3.5. Database - PhpMyAdmin

PhpMyAdmin Database is an open source and free tool to handle data, whether it is MySQL or MariaDB data. It is written primarily in php and it is used for web hosting services. For it to work on the local machine, a program called XAMPP had to be installed.

In this thesis, this tool was used to save the data in tables. Data that came from the facial expressions, the clicks, the BITalino and the Tobii. From the beginning, the use of *phpAdmin* was considered because it allows an easy way of communicating between programs that were written in different code languages, and in this tool, there are various programs that use different coding languages, as seen in the General Diagram (see Fig 3.1). However, there are other databases that also facilitate communication between programs that were written in different languages, such as Microsoft SQL, DBeaver, Snowflake, and more. The database model can be seen, in Fig 3.2.

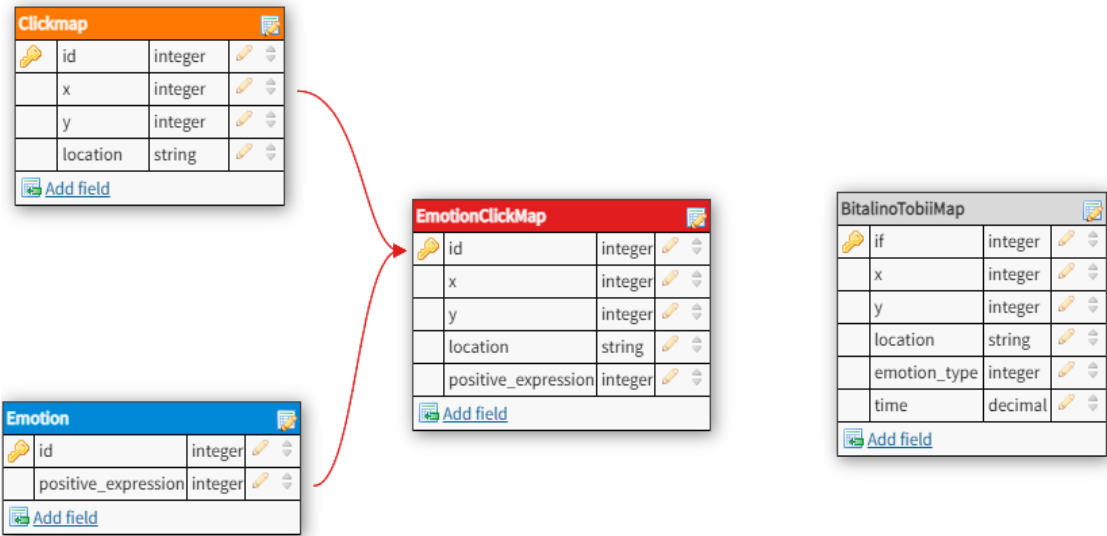


Fig 3.2 - Database model

For facial expressions, the table is called “*emotion*” (see Fig 3.3), and the process of how the emotional state is saved was explained in section 3.2.

The screenshot shows the phpMyAdmin interface with the 'emotion' table selected. The table has two columns: 'id' and 'positive_expression'. The data is as follows:

id	positive_expression
1	-1
2	-1
3	-1
4	-1
5	-1
6	-1

Fig 3.3 - "Emotion" table

For the clicks, the table is called “*clickmap*” (see Fig 3.4) and is composed by the “x” and “y” position of the pixel that was clicked and the location of the page that was clicked.

The screenshot shows the phpMyAdmin interface with the 'clickmap' table selected. The table has four columns: 'id', 'x', 'y', and 'location'. The data is as follows:

id	x	y	location
1	260	128	/C%3A/xampp/htdocs/clickmap/WebPage1_with_brakes.h...
2	313	132	/C%3A/xampp/htdocs/clickmap/WebPage1_with_brakes.h...
13	359	142	/C%3A/xampp/htdocs/clickmap/WebPage1_with_brakes_v...
7	357	143	/C%3A/xampp/htdocs/clickmap/WebPage1_with_brakes_v...
8	486	219	/C%3A/xampp/htdocs/clickmap/WebPage1_with_brakes_v...
3	346	315	/C%3A/xampp/htdocs/clickmap/WebPage1_with_brakes.h...

Fig 3.4 - "Clickmap" table

The tables described above had to be separated because the programming language used to get the clicks was not the same language used to get the emotional states. If both clicks and emotional states were inserted in the same table, it would not be possible for them to be in the same table entry, not even with the same attributes. So, the program that wrote the emotional states would always ask phpAdmin if there had been a click. If there was a click, this program would save the emotional state exactly with the same ID as the click, to then do a merge of the tables in a single one (See Fig 3.5). This merge is done by clicking on a button on the interface. This process is explained later (Page 44).

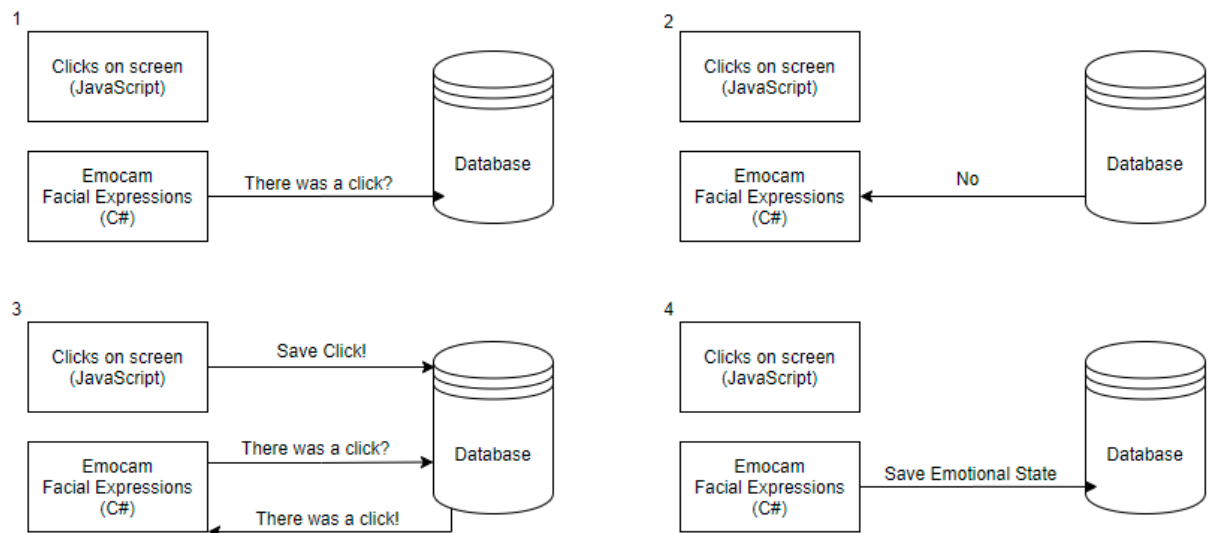


Fig 3.5 - Emotional States Process

For the Bitalino and the Tobii 4c, the table is called “*bitalinotobiimap*” (see Fig 3.7). It was not necessary to separate the BITalino from Tobii, as was done previously for facial expressions and clicks, because the data coming from BITalino and Tobii are from the same program, the Gagan program. It was enough to insert into the same entry of the table, with just one line of code, both the cognitive state of the user and to what pixels he/she was looking at (see Fig 3.6). The table is composed by the “x” and “y” coordinates, that is the eye gaze of the user, the “location” of the page to where the user is looking at, the “emotion_type” that is the cognitive states of the user, that can be 0 if it’s the Rest state, 1 if it is the Intense Workload state and 2 if it is the Workload state and finally the “time”, to know when the user had the interaction with the interface. Why the time is used will be explained in detail later (Page 44).

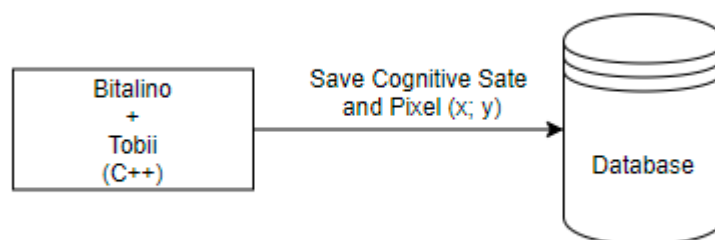


Fig 3.6 - Cognitive States Process

			id	x	y	location	emotion_type	time	
<input type="checkbox"/>	Edita	Copiar	Apagar	57011	626	15168	/C%3A/xampp/htdocs/clickmap/TestInterface_v3.html	1	349.609
<input type="checkbox"/>	Edita	Copiar	Apagar	56907	1932	15098	/C%3A/xampp/htdocs/clickmap/TestInterface_v3.html	2	244.963
<input type="checkbox"/>	Edita	Copiar	Apagar	56134	152	15074	/C%3A/xampp/htdocs/clickmap/TestInterface_v1.html	1	514.861
<input type="checkbox"/>	Edita	Copiar	Apagar	56999	153	15070	/C%3A/xampp/htdocs/clickmap/TestInterface_v3.html	0	337.526
<input type="checkbox"/>	Edita	Copiar	Apagar	57007	25	15034	/C%3A/xampp/htdocs/clickmap/TestInterface_v3.html	1	345.577
<input type="checkbox"/>	Edita	Copiar	Apagar	56149	19	15025	/C%3A/xampp/htdocs/clickmap/TestInterface_v1.html	1	530.977
<input type="checkbox"/>	Edita	Copiar	Apagar	56653	21	15024	/C%3A/xampp/htdocs/clickmap/TestInterface_v2.html	2	504.531
<input type="checkbox"/>	Edita	Copiar	Apagar	56845	1935	15018	/C%3A/xampp/htdocs/clickmap/TestInterface_v3.html	0	182.536
<input type="checkbox"/>	Edita	Copiar	Apagar	56660	0	15011	/C%3A/xampp/htdocs/clickmap/TestInterface_v2.html	0	514.599
<input type="checkbox"/>	Edita	Copiar	Apagar	56143	24	15009	/C%3A/xampp/htdocs/clickmap/TestInterface_v1.html	1	524.933
<input type="checkbox"/>	Edita	Copiar	Apagar	56144	24	15009	/C%3A/xampp/htdocs/clickmap/TestInterface_v1.html	1	525.939

Fig 3.7 - "Bitalinotobiimap" table

The data described above were saved so that later a call could be made to the database to give us the results in two distinct overlays, which cover the test interface. This call is made by two buttons present in the test interface, the "Display Click Map" button and the "Display Emotion Map" button.

When the "Display Click Map" button is clicked, a merge between the "clickmap" table and the "emotion" table occurs. A JavaScript and a php function were developed to do that. The merge results in a new table, the "emotionclickmap". This button also makes a call to get the values from this new table and shows them in an overlay, in front of the interface that is being tested. In the overlay, the user can see colored dots with three different colors: green, yellow and red. Green means that the user had a positive facial expression and corresponds to the value 1. A yellow dot means that, when the user clicked on that position, he had a neutral facial expression, corresponding to the 0 value. Finally, a red dot means that the user had a negative facial expression, corresponding to the value -1 (Appendix C).

The "Display Emotion Map" button is similar to the "Display Click Map" button, however, it does not merge two tables because it is not necessary. As described earlier, the cognitive states of the user and his eye gaze are already in the same table, so this button only makes a call to the "bitalinotobiimap" table to show the results in an overlay, in front of the test interface. This overlay is similar to the one described above, however, when the point is green, it means that the user was in a rest state and corresponds to the value 0. A yellow point means that the user, when he/she looked at that position, was in a workload state, corresponding to the value 2. Finally, a red point means that the user was in an intense workload state, corresponding to the value 1 (Appendix D).

In this specific overlay, it was developed a timeline widget for people who would analyze the user results (see Fig 3.8). This was developed so that analyzers could better understand why the user at a certain point was in a certain cognitive state. Physiological measurements ECG and EDA take a few seconds to manifest, so a dot appearing in red may mean that it was something that happened a few seconds ago. With this timeline, the analyzer can know where the user was looking seconds ago, thus discovering the reason why that point was red. It had two arrow buttons, one that represented “step back in time” and the other represented “go forward in time”. In the middle of these two arrow buttons, there was a text box where the analyzer saw the time in seconds and below this text box there was another that showed the jump that would be applied. This time jump could be changed manually or there were default buttons for each sensor that would be applied in the experience. The first one was the “ECG” button, and it altered the time jump by adding 0.5 seconds. The second one was the “EDA” button, and it altered the time by adding 2 seconds. Finally, the last button, called “Default”, altered the time by adding 1 second. The EEG button was not included because the time that the results of this physiological measure take to manifest is so short that it is considered "live", unlike the other two measures that have a delay. To change it manually, the analyzer just needed to write in the text box that showed the time jump that would be applied. Every time the analyzer clicked on one of the arrow buttons a call was made to the “bitalinotobiimap” table that asked for all points that had an associated time shorter than the time requested by the analyzer, that is, the time that was in the text box that showed the time. That is why the time column exists in the “*bitalinotobiimap*” table.

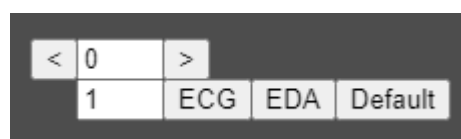


Fig 3.8 - Timeline

3.6. Merging GUIs

A python file, named ProcessImage.py, was developed to give a merge between the three tested interfaces, where the final interface is the set of the parts where the user reacted more positively. This python file makes queries to the tables described in section 3.5, to get the list of points where the user had a negative emotion state or a negative cognitive state.

Summarizing the process, the file cut the first tested interface from the top to where there was a negative emotion state or a negative cognitive state and pasted the cut interface in the

final interface. Then, passed to the second tested interface and did the same process, thus giving the best interface. The order in which the tested interfaces were processed was first, second, third and back to the first until the negative points were all processed. The process will be explained in more detail below.

For this to work the file first needed images of each one of the interfaces that were being tested. For that, a program called “ShareX” was used. This program is great for taking screenshots of interfaces that have a scroll down option. To open the images, the python file used a library named “PIL”.

After the interfaces prints, the program needed to know what text was present in these three interfaces and in which position. For that, two python libraries were used, one named “pytesseract”, that uses “OCR” and the other named “OpenCV”. This library runs through the print taken, detects text present in it and detects the box surrounding the text and saves this in a list (sentence, top left corner height, bottom left corner height). The reason why this was used will be explained later (page 46).

After these steps, the information present on the *PhpMyAdmin* database was important to know where the user clicked and looked at, and what were the emotion states and cognitive states of the user. For that, a library was used to connect the python program to the database named “mysql.connector”. When the file was connected to the database, various queries were made, one for each interface. The only dots/pixels that needed to be fetched from the database are those where the user had a negative facial expression (-1) or if intense workload (1) or workload (2) were detected. For that, a query like the one seen in Fig 3.9 was done:

```
query1 = mydb.cursor()
sql1 = "SELECT y FROM emotionclickmap WHERE positive_expression = '-1'
AND location = '/C%3A/xampp/htdocs/clickmap/TestInterface_v1.html' ORDER BY y ASC"
query1.execute(sql1)
```

Fig 3.9 - Query Example

The parameter “y” was the only coordinate of interest, since the image was cropped horizontally, so it was only necessary to know the height and not the width of the interaction. To understand the logic, just think that if a point with negative facial expression or a cognitive state of intense workload or workload was found, it meant that everything in the interface that was above that point, the user reacted positively, so the interface was cut from the top to that point and that portion was analyzed to see if it contained negative interactions or if the text that was present in that portion was already present in the final interface. If there were negative

interactions or repeated text, they were not inserted in the final interface. The parameter “location” was to know in which test interface the interaction was. In the query, the “y” is ascending so that the first negative point to get is the highest point on the interface.

With everything mentioned above, the boxes of the text present in each of the interfaces were found and the coordinates of the negative interactions were also found. The reason the interface text was needed is because the final interface could have repeated sections. If an interface has a section to fill in the name at the top of the interface and another test interface has a section to fill in the name at the bottom, in the final interface, these two sections could be present, if both were not associated to a negative emotion. Hence, the text of the interfaces was needed. However, “*pytesseract*” / “*OCR*” is not perfect, and some text may be poorly detected, for example, as seen in previous interfaces that were made to test all described above, “name” being detected as “nene”. The solution found was to see similarities between words. For this, another python library was used, named “*difflib*”. This library has a function, named “*SequenceMatcher*”, that tells how similar two strings are. After some tests, the best value to detect that the word was similar was 0.7, so when comparing two strings, if the result is bigger than this value, the strings are the same. With this, the same section could not be repeated in the final interface.

The coordinates of the negative interactions that were made, were saved in a list which was made to keep the interaction’s “y” coordinate and to which test interface that same interaction corresponded. For example, the list could be: [(45,1), (32,2), (78,3),...]. The first position of the list means that an interaction on the first test interface was made, because it has the value 1, and it was on a pixel with height 45.

Finally, an iteration was made through this list of the negative interactions until reaching the end of the list. Inside this iteration, another iteration was made by the list of words from the correct interface, to know which word that interaction was associated with. If the “y” value of the iterator was inside of one of the text boxes present in the list of words, it meant that the word associated with the negative interaction was found. As the list of negative points was made as (“y”, “interface”), it was possible to obtain the text list of the correct interface. After finding the word that corresponded to the negative interaction, the tested interface was cut from the top to the negative interaction (excluded) and another list was created to see the text that was present in this sub interface. This was made to see if in this sub interface there were other negative interactions in addition to the one being iterated and to see if there was repeated text between the final interface and the one to be cut. If there was a negative interaction or repeated text in this sub interface, those words were excluded from the text list of the sub interface. Then,

an iteration was made through this sub interface text list to cut the words and paste them in the final interface. In the final interface, to know from which test interface the word was cut, colored lines were around the cut, making a rectangle. If it was the first test interface the color was red, if it was the second test interface, the color was yellow, if it was the third test interface, the color was green (Appendix E).

In certain cases, a section of an interface may not be inserted into the final interface because, in all test interfaces, that section is associated with negative emotional states or negative cognitive states. Then, after the optimized interface is generated, another one is generated that contains the sections that were not inserted in the optimized interface, helping the designer to identify the parts that were not inserted, so that they can be more careful in the redesign (Appendix L).

Summarizing this chapter, the emotional state associated with clicks and the cognitive states associated with eye-gaze are saved in a database, so that a designer can later see these same data in two distinct overlays, to realize where the user clicked and in what emotional state, he/she is, where the user looked and with what cognitive state. This data is then also used by the tool to merge the interfaces, to give us the optimized interface in the end, formed by the parts where the user reacted more positively. In short, it can be concluded that the entire architecture of the project was described and explained, as well as each process that forms that architecture and the equipment that was used.

Chapter 4 - Case Study and Evaluation

This chapter will describe how the tests were carried out. According to the developed project, a case study was chosen, and an evaluation was carried out with designers from a specific IT company to evaluate the user's feedback. For data collection, both emotional and cognitive, it is important to note that due to the pandemic, caused by the Covid-19 virus, the users tested were known and familiar, so it can be said that the tests were carried out for convenience.

This study had two objectives. The first was to develop a tool based on user monitoring (emotions and attention), to provide optimal interaction, related to user emotions and attention. The tool development was already described in chapter 3. The second objective was to perform a user study to verify whether the developed solution brings improvements compared to the conventional method used by the company. That objective will be described throughout this chapter. The conventional method used by the company consists in developing an interface and then users test the interface and tell the designers what they think is good and what is not so good. In this use case, the user feedback will be the overlays (the clicks associated to the emotional state and the eye gaze associated to the cognitive state) and the designer will have to build an interface depending on the results of that same overlay. In the end, the designer's results will be compared to the tool results.

4.1. Design

In the testing phase, it was important to assess the following:

- If the neuronal network was well trained, to find out if it was a viable source to justify future conclusions.
- If the overlays (results from users) were easy to interpret, user friendly and if with only the overlays it was possible to build manually an optimal interface.
- If the tool developed was useful to be used by designers.

4.1.1 Neuronal network reliability

To assess whether the neural network was well trained, in the test interfaces, two tasks were developed that would stimulate either the cognitive state of workload or the cognitive state of intense workload.

One of them was to memorize a code at the beginning of the test and, when the user finished the tasks he had to do or the time he had to do those tasks ran out, the user was required to put the code memorized at the beginning of the test, at the end of the interface. This task would trigger the cognitive states mentioned above because to complete it, it is necessary to remember something, to think, which implies a minimum workload. The idea was, after the test is finished, to see the overlay (user results) and check if there was a yellow or red dot in that area (workload and intense workload, respectively). If so, it was possible to assume that the network was well trained.

The second task that would trigger the cognitive state of workload or intense workload would be an impossible task to complete. In the script presented to users, there was a task that the answer was not present in the interface. The idea here was to generate nervousness or irritation in the user, as they could not find the answer to the task and had time to do it. Once again, at the end of the test, the idea was to see the overlays (user results) and check if there were yellow or red dots while the user was solving this task.

With these two tasks developed in this way, it was possible to verify whether or not the neuronal network was well trained and could be used as a source of justification for future conclusions.

4.1.2 Overlays interpretation

To assess whether the overlays were easy to interpret, designers were asked to build an optimal final interface manually. This optimal interface should be a merge of the parts where the user reacted more positively. This process was recorded, with the consent of the designers, so that it could be reviewed later. The idea of this review was to understand if there were human errors and if they were due to a distraction or if they were due to the overlays being difficult to interpret. In the end, to also get the designers' opinions, a "USE Questionnaire: Usefulness, Satisfaction, and Ease of use" [39] inspired questionnaire was developed, because this type of questionnaire is used to measure the ease of use of a product or service, and the overlays ease of use needed to be measured. One of the questions in this questionnaire was "Is it user friendly?".

4.1.3 Tool usefulness

To verify if the tool was useful for the designers, the manual interface developed by them, through the users' overlays, already mentioned in section 4.1.2, was compared to the interface generated by the tool. If the results were equal to each other, it meant that the tool that builds

the optimal interface "automatically" was not useful, in the sense that, designers could only use overlays to build that interface. However, it was also necessary to understand if the tool is applying the rules that will be described in section 4.3.4. For this, all interfaces generated by the tool were revised, all cut sections were checked to see if there were any negative points in them. The recordings of the interface generated by the designers were also reviewed to see if this result might not be similar to the result generated by the tool because the designers misapplied the rules.

4.2. Experimental apparatus

To train the neuronal network, electrocardiographic (ECG), electrodermal (EDA) and electroencephalographic (EEG) data were collected with BITalino.

To collect the ECG data, three electrodes were used, one positive, one negative and the reference. The positive electrode was placed below the left clavicle, the negative electrode was placed below the right clavicle and the reference electrode was placed in the iliac crest (see Fig 4.1) [35]. There were other options of positioning the electrodes for this signal however these positions that were tested gave more stable values. [33]



Fig 4.1 - ECG electrodes position [35]

To collect the EDA data, two electrodes were used, the positive and the negative because in the manual of BITalino only these two were referenced. The negative electrode was placed in the left hand thenar eminence zone, closer to the thumb and the positive electrode was placed also in the left hand thenar eminence zone but under the negative electrode, closer to the wrist (see Fig 4.2). [3]



Fig 4.2 - EDA electrodes position [36]

To collect the EEG data, three electrodes were used: the positive, the negative and the reference. Both the positive and the negative were placed on the forehead and the reference electrode was placed behind the left ear. [3]

After the sensors were all prepared, the data was collected.

To collect the user's facial expressions, the camera's own laptop was used in conjunction with the EmoCam program. This process is described in chapter 3.

Tobii 4c eye tracker was used to collect the users eye gaze.

4.3. Procedure

The case study was divided into four steps. In the first step, two pilot tests were conducted to know what method was going to be used to train the neuronal network, if with an interface similar to the ones used in the final test with users or with tasks that increase the cognitive states of the user. The second one was to collect data to train the neuronal network, already knowing what method to train. Bear in mind that the activities carried out to collect data lasted the same time, 5 minutes. The third step was to test this trained neuronal network and to generate the optimal interface, generated by the python file. The final step was to send users results, that is the overlays, to see if the designers could replicate the optimal interface generated by the tool, to then compare it to the outcome of the tool. If they were able to replicate the same outcome as the tool, then there would not be much use for the same. In the end, the designers answered a questionnaire about their own experience.

4.3.1 Pilot Tests

In this first step, two pilot tests were made to see how the neuronal network was going to be trained. This was important to do because, for the neuronal network to work, it was necessary

to have user data in each of the cognitive states. If the files were very similar to each other, then the neuronal network would not be able to train, because, when it tried to learn, it would not be able to distinguish between one state or the other. In the first pilot test, tasks that increase the cognitive states of the user were carried out independently, which means, there was no interface, and in the other pilot test, a similar interface and a similar script of activities, used in the final step with users, was used.

In the first pilot tests, to collect data related to the cognitive state of Rest, the user was asked to relax for 5 minutes, doing nothing.

To collect data related to the cognitive state of workload, a digit-span task was presented to the user. This is a test of short-term memory. The user is asked to repeat a string of numerals spoken by an examiner. The string is made progressively longer in order to determine the numerals that can be recalled. In this case, the examiner was just a website that had this memory test implemented and the user needed to insert the numbers by the correct order to see if he/she memorized it or not. This activity had a duration of five minutes and the number of digits that the user needed to memorize was six. [34]

To collect data related to the cognitive state of intense workload, the same digit-span task was presented to the user, but instead of memorizing six digits the user needed to memorize eight. The task had a duration of five minutes.

In the other pilot test, that used an interface similar to the one used in the final tests, to collect data related to the cognitive state of workload, there was a code or ID, composed of six digits, that the user had to memorize for ten seconds. Then this code needed to be inserted in a text box at the end of the interface, simulating the digit-span task, after the user has completed some tasks with a maximum duration of ten minutes.

Similarly to the first pilot test, to collect data related to the cognitive state of rest, the user was asked to relax for 5 minutes, doing nothing.

To collect data related to the cognitive state of intense workload, the task was very similar to the task of collecting data for the workload state in the second pilot test, however, instead of having to memorize a six-digit number, it had to memorize an eight-digit number.

At the end of the first and the second pilot tests and after the neuronal network was trained in different ways (depending on the pilot test), users tested in an interface similar to the one they used in the second pilot test. The script used was different from the script used in the second pilot test. After the users finished the script tasks, the buttons that built the overlays were clicked and the estimated cognitive states were analyzed. In the end, the idea was to compare the overlays between the first pilot test and the second pilot test and see which one

predicted cognitive states that made more sense in certain areas of the interface. For example, the interface had a zone where a code was to be entered and, in that zone, it was assumed that the user should be in the cognitive state of workload or intense workload, since it was necessary to remember a code that was asked to memorize in the start of the script. Note that the users followed the same script at the end of both pilot tests this time, to distinguish what method to train the neuronal network was better.

The final results showed that the best method to train the neuronal network is the one that uses the digit span task. Using the other method, the method applied in the second pilot test, it was not even possible to train the neuronal network. This is because the output files given by each of the tasks corresponding to each of the cognitive states (rest, workload and intense workload), were very similar, so the network could not distinguish the states, meaning that the network could not learn. The graph corresponding to the tested users network overfits very early, that is, the network is not learning but rather memorizing the values of the files, so when new values are given to the network, it will not be able to predict the cognitive state.

With this, for the testing phase, the six-digit digit span was used to collect data associated with the workload state, the eight-digit digit span was used to collect data associated with the intense workload state and, to collect data associated with rest, the user had to be just quiet and still. All of these tasks lasted five minutes.

4.3.2 Collecting Data and Training the Neuronal Network

In this second step, a questionnaire (Appendix A) was made, and data were collected for later use in the training of the neuronal network. Only electrocardiographic (ECG), electrodermal (EDA) and electroencephalographic (EEG) data were collected, because, in order to train the neuronal network, it was not necessary the eye gaze of the user.

The questionnaire was the first thing to be done and it was done to collect demographic data, to find out if the user had emotional illnesses (yes / no answer) and to find out if the user considered himself/herself a person who uses computers (Appendix A).

After the questionnaire, the data was collected, using the method proved to be better, as seen in the pilot tests. The six-digit digit span was used to collect data associated with the workload state, the eight-digit digit span was used to collect data associated with the intense workload state and, to collect data associated with rest, the user had to be just quiet and still. All of these tasks lasted five minutes.

In the end of each task, a file was generated containing the values of HR, RMSSD, SDNN, LF, HF, SVB (these values are associated to the ECG); SCL, SD (these values are associated

to the EDA); and Alpha, Beta, Theta, Engagement (these values are associated to the EEG). So, in the end, a file was generated for the rest state, another was generated for the workload and another file was generated for the intense workload.

To train the neuronal network, a python program was used, made available by the author António [38] and only a few changes were made, such as reducing or increasing neurons per row, reducing, or increasing epochs, reducing or increasing queues, among others. Whenever a network was trained, two graphics appeared (seen in Fig 5.1 and 5.2). The most important thing was to be aware of the red line in both graphs. The red line refers to the validation data. The left graph showed the network accuracy, and the right graph showed the net loss. The graph of accuracy is the classification accuracy of the neural network, both in the training data and in the validation data (blue and red lines respectively). Naturally, the accuracy of the validation data must be lower, because it represents data that the network has never seen during training. In training, data accuracy is naturally high because it is the data that the network knows because it trained with them. The loss graph basically represents the cost associated with training the net, with low costs representing more accurate classifications. As in the accuracy graph, the accuracy is naturally high for training data, here the loss will naturally be low, because the network knows the data and classifies it with high accuracy, and therefore a low cost is attributed. In validation data, a decreasing cost means that the network is learning from the data, however if it reaches a point where the cost starts to rise, which is called the overfitting point, the cost starts to rise, meaning that the network is no longer simply recognizing the natural patterns of the dataset, but "memorizing" it and finding patterns that are too complex for the data it is given. When this happens, training should stop, as the overfitting point is the point at which the network is no longer able to faithfully recognize any data other than what it trained with. That's why the loss in training continues to go down, but it starts to go up in validation

To find out if the network was well trained, it was necessary to study the loss graph and see if it reached a point of overfitting.

4.3.3 Testing the Neuronal Network and Generate the Optimal Interface

In this third step, the neuronal network was already trained, and it could be used to predict the user's cognitive states as they use the test interfaces.

In this step, in addition to ECG, EDA and EEG data being collected, facial expressions, clicks and the user's eye gaze were also collected.

After the sensors and devices were all ready and in place, the user was able to test each of the three interfaces presented to him. While the user was using the three interfaces, a script of

activities was being read (Appendix B) and all the data was being collected, clicks (pixels) and facial expressions, eye gaze (pixels) and the cognitive states of the user. The prediction of the cognitive state was being given in real-time. All this data was saved in the *phpAdmin* database.

To later observe the data collected by each device, two buttons were created on each of the three interfaces to create an overlay on the interface itself (Appendix C; Appendix D). One button shows the pixels, collected by the user's clicks, associated with facial expressions and the other button shows the pixels, collected by the Tobii 4c, associated with cognitive states. When the user clicks on these buttons, a call to the *phpAdmin* is being made and the result of this call is the overlay with the points. To generate the optimal interface the python program was run.

4.3.4 Designers Manually Construct the Optimal Interface

In this last step, a team of designers was asked to design manually the optimal interface according to the results of the tests done to the users (*clickmap* and *emotionmap*), to then compare it to the outcome of the tool, the optimal interface. If the results were similar, it could be concluded that the developed tool was not useful. To design the optimal interface the designers had to follow the same rules that the tool, developed in python, applies. The rules were, the designers were obliged to start by analyzing the results of interface 1 simultaneously, that is, observing *clickmap_interface_v1* and *emotionmap_interface_v1* simultaneously. Starting at pixel 0 and then vertically going down, like a scanner, until the first negative point is detected (see Fig 4.3, 4.4, 4.5). In the *clickmap*, a negative point was just a red point, which was associated with a click and a negative facial expression, and in the case of the *emotionmap*, a negative point can be either a yellow or red point, workload and intense workload respectively, associated with the eye gaze of the user. After detecting that first negative point, the designer would cut everything that is up from that point, because everything that is above that point is positive.



Fig 4.3 - Designer process part 1



Fig 4.4 - Designer process part 2

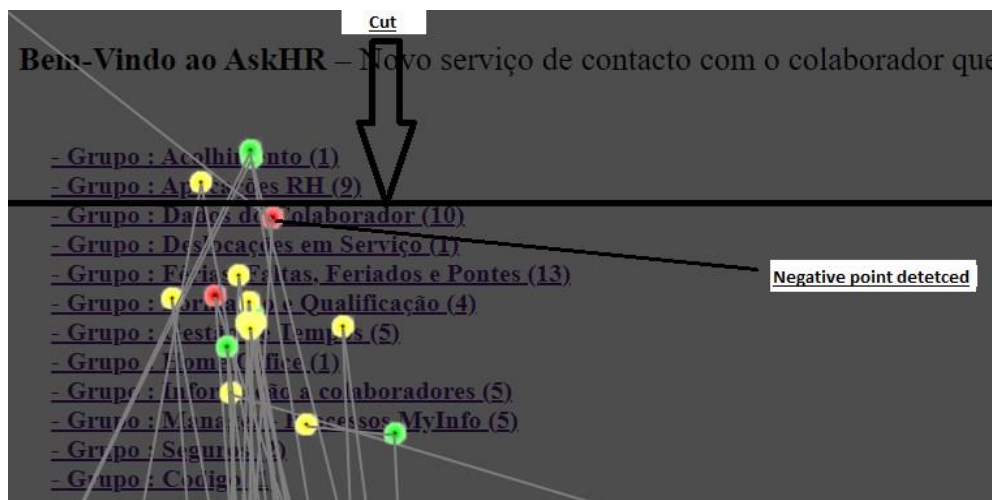


Fig 4.5 - Designer process part 3

After this cut, the designer pastes that section of the interface into *paint* and passes to the results of the interface 2. Does the same process and then pass to the results of the interface 3. Finally, while doing the same process, passes to the results of the interface 1 and repeat the whole process, until there are no more negative points in one of the users' results.

The designers were warned that when they were cutting out a section, they should pay attention to three more rules. The first one was to memorize where they were in the results of the interface they were observing, to then, when they returned to those results, know where to resume the analysis. The second one was for them to see if the section they were pasting was not already in the optimal interface. The last warning was for, when they were cutting a section, to see if, inside that section they were cutting, there were no strings or parts that were associated with negative points, because if there were, they could not insert those parts present in the section they were going to cut.

This entire process was recorded with the permission of the designers, because in the end all recordings were reviewed to see if the designers respected the rules which the tool applied. These recordings are completely anonymous.

As this process was complicated to explain and to maintain consistency among designers, a script was written so that they would all hear the same speech, and so that the announcer does not forget to say any important details, such as the rules that the tool follows (Appendix G). A video was recorded explaining all the process described above.

As the interfaces used in the tests with the users were quite large, only a small section of the user's overlays (results) was cut so that the tests with the designers were quicker (approx. 450 minutes). If instead of having given the designers the initial part and if they had given a part more in the middle or even the final part, and the designers started to cut through interface 1, as the rule says, when the interface generated by the designers was compared to the interface generated by the tool, the results would never be the same because the tool analyzed the interface from the beginning. So, when the tool reached the section that we gave to the designers, the tool might not be starting with the results of interface 1, but of 2 or 3. Therefore, the result would never be the same, and the result of the designers could not be compared with the result of the tool. Hence, the initial part was given.

After the designers developed the optimal interface, a questionnaire was answered (Appendix F). This questionnaire was based by the "USE Questionnaire: Usefulness, Satisfaction, and Ease of use" [39], because this type of questionnaire is used to measure the subjective usability of a product or service and the tool usability needed to be measured to understand if this tool was useful for the designers.

Summarizing chapter 4, two pilot tests were done to find out what is the best method to train the neuronal network in real tests. After knowing the best method, tests were made by each user to train the neuronal network. This network was trained and then tested by the same users who gave data to train it. After that, designers observed the results of the users and made

a new interface according to those results. To collect physiological data, BITalino was used, to know where the user was looking, Tobii 4c was used and to collect facial expressions, the camera integrated in the computer was used.

Chapter 5 - Results and discussions

In this section, the optimal interface generated by the tool and the results of the evaluation made with the users, who used and performed tasks in three interfaces, will be presented and discussed. These results are called overlays, which are constituted by the points that represent the emotional states associated with clicks, in the case of the *click_map* overlay, and the cognitive states associated with the eye gaze, in the case of the *emotion_map* overlay.

The results of the designers will also be presented, who were asked to manually generate an optimal interface according to the users' results, the overlays, comparing them to the optimal interface generated by the tool.

5.1 Users

Before the sensors were put in the users, they answered a demographic questionnaire (Appendix A). Twenty users were tested. Due to the covid-19 virus, it was not possible to test with more people, nor with randomly selected people, as it was meant to be. Thus, family members and acquaintances were tested. Of the 20 testers, 45% were female, making 55% male. The age range was 18-60 years old, as this age range is the age that has a professional work, and this thesis was developed to be used in cases where workers from an IT company use an interface and then designers from that same company evaluate the results with the overlays and the tool. The age group was mostly 18-30 years old (70%), 90% had at least higher education, none with emotional illness and 55% of users use the computer for more than 7 hours per day. Everyone accepted that their physiological data was collected and analyzed anonymously, and their face was filmed but not recorded in order to see their facial expressions.

5.2 Overlays

The metrics used to analyze whether the points of the overlays made sense or not, was the accuracy of the neuronal network, a zone in the interface where the user was meant to be in one of two specific cognitive states, and one of the tasks was impossible to resolve, already mentioned in section 4.1.1.

Whenever a neuronal network was trained, it was always attempted that its accuracy was above 70% and that there would be no overfitting, phenomena already explained in section 4.3.3. If this did not happen, the properties of the neuronal network would be altered, and the network would be re-trained.

The maximum accuracy achieved was about 97% and the lowest was about 68%. (see Fig 5.1 and Fig 5.2)

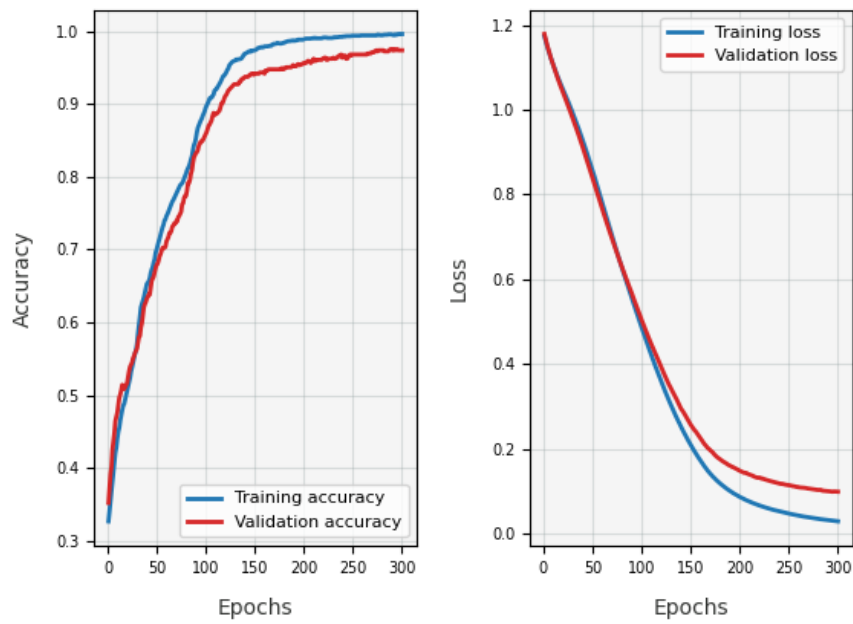


Fig 5.1 - User 5 neuronal network results

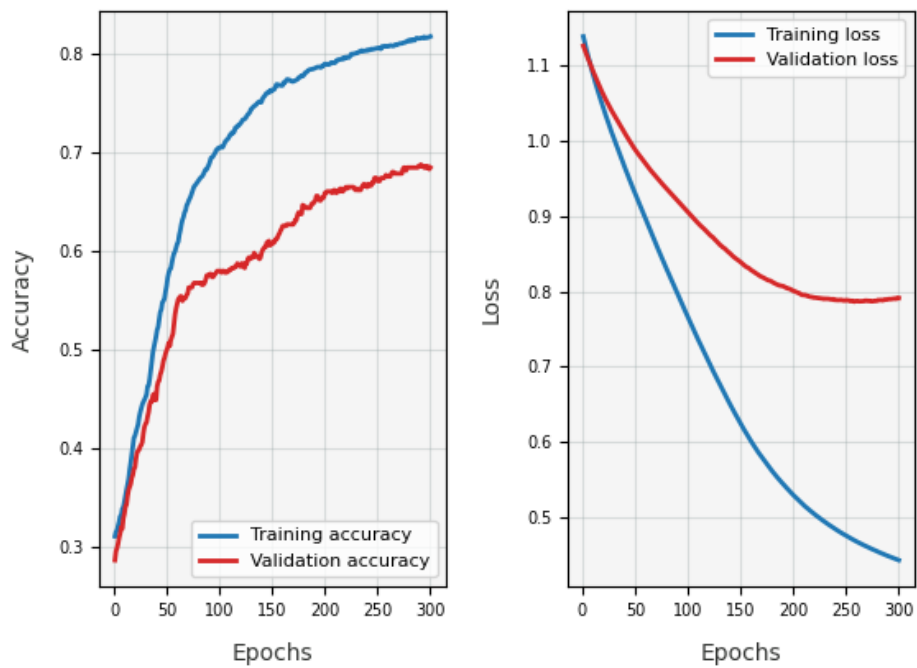


Fig 5.2 - User 20 neuronal network results

Analyzing the overlays themselves was not easy, given that you cannot find a right reason for the user to be in that cognitive state or having that facial expression, in that specific point,

because emotions are very subjective, something that differs from user to user. However, the interface was designed where at least, in one place, the user would be in one of two specific cognitive states, the area where the code was inserted (see Fig 5.3), already mentioned in section 4.1.1. In this zone, it was assumed that the user should be either in workload or in intense workload, as it was necessary for the user to remember a code that was asked to decorate at the beginning. In 60% of the users' results, a green dot, which represents rest, was not detected in the code area. The other 40% can be assumed that, as the neuronal network did not have an accuracy of 100%, the green dot that appeared was a false positive or that simply the user, in the zone of inserting the code, was in fact in the state of rest.

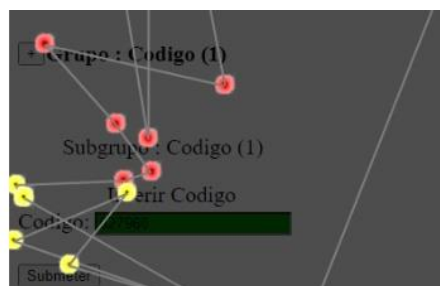


Fig 5.3 - User 18 Emotion Map, Code Zone

To also detect if the neuronal network was well trained, one of the tasks asked to the user was impossible to resolve, also mentioned in section 4.1.1. The idea was that when users reached this task, there could be no green dots. When users arrived at this task, it was expected they were very nervous because they had time to finish the task list. However, observing all the overlays, looking for a specific task and understanding which points are associated with that task, is not easy. Something that remained to be done in the testing phase was to point out the minutes when the user reached the impossible task to perform and then, when analyzing the results, use the timeline widget developed in section 3.5 to find out what cognitive state the user was in. So, this metric cannot be used to state with certainty that the network was or was not well trained.

Although few users have been tested, 60% of correctness in the code area is a good indication that the neural network was well trained, despite the accuracies, and the overlay can be used to show the monitorization of the user's attention and cognitive states. It is important to emphasize that analyzing user emotions is never an easy task since it is something very subjective and a user being in a state of rest and having certain EEG values, for another user, he can be in the workload state and the EEG values are the same as the previous user.

5.3 Optimal Interface

The optimal interface results were not what was expected. As mentioned in section 3.6, the tool should indicate which interface it is cutting from, drawing a rectangle around the cut section. If the rectangle was red, it was cut from interface 1, if it was yellow, it was cut from interface 2, if it was green, it was cut from interface 3. This functionality was well achieved. Another feature that was well achieved was the fact that the tool was only inserting the sections where the user had more positive emotional states or cognitive states, that is, it did not insert sections where negative points were present. However, many sections are repeated, and the flow of the optimal interface does not have context.

To analyze if the optimal interface was constructed only by the positive parts of the 3 interfaces, each section was analyzed and from which original interface they came from. In each inserted section, it was painted around, forming a rectangle, with the respective color to each of the original interfaces used by the user. If it was red it was removed from interface 1, if it was yellow it was removed from interface 2 and if it was green it was removed from interface 3. It was verified in the respective overlays for each of the sections, if in fact they were positive sections or not. This was confirmed. Therefore, it can be assumed that the rule of cutting only positive parts was respected and the colorization of each section was also respected.

However, it was detected that, in all the optimal interfaces, it had repeated sections. This problem tried to be overcome using a library already mentioned in section 3.5, however there were repetitions of sections. This is due to the fact that there are sections of the interface that are very similar. Take Fig. 5.4 and 5.5 as an example. Because the strings are so similar, the tool may find that both strings are the same section when they are not. The only difference between the two is a "+" and a "-" sign.

- Grupo : Acolhimento (1)

Fig 5.4 - String present in the interface

+ Grupo : Acolhimento (1)

Fig 5.5 - String present in the interface

The reason for the lack of context in the optimal interface it is because the test interfaces that were merged had the same elements but arranged differently. When a section was pasted at the final interface, in the next step, the tool analyzed another test interface, where the context of the previous interface that was cut was changed. For example, in an interface, a section related to "Insurance" was cut and pasted in the final interface. Then, the tool goes to the next

interface and cuts out a section related to "Home office" and pastes it into the final interface. Finally, the tool changes to another interface and once again cuts a section related to "Insurance" and pastes it into the final interface. So, in this example, it can be seen that, in the final interface, the "Home office" section is in the middle of the "Insurance" section, making no context in the final interface.

5.4 Designers Results and Discussion

The metrics used to demonstrate the results of the designers were the answers to each of the questions present in the questionnaire, inspired by "USE Questionnaire: Usefulness, Satisfaction, and Ease of use" [39], which was filled out right after the tests were done, the average of the value of the responses and their standard deviation. The questionnaire in the end had a section for the designers to write negative and positive aspects. To find out if the designer made mistakes, all recordings were reviewed.

The entire process of building the optimal interface by the designers was always accompanied in case they had any doubts.

Due to the SARS-COV 2 virus, all the tests done to the designers were done remotely through the Teams program and it is important to underline that the designers only had access to images of the overlays and not the overlays themselves, therefore they did not have the timeline of the points. It was not possible to give the overlays to the designers because, for the calls to the database to work, it is necessary to install the XAMPP program, already mentioned in section 3.5. The IT company from where the designers came from, had very strict rules with the programs that can be installed on the company's computers, computers that were used by the designers to carry out the tests. If they had the timeline, the results could possibly be quite different, given that a designer can see point by point, contrary to what was presented, a mesh of points, difficult to analyze.

The tests were done to 8 designers and they were divided in two groups, 4 per group. This division was made so that the designers did not analyze only one result of the users. Since there are 2 groups, only 2 results from the 20 users were analyzed. In the end, a questionnaire (Appendix F) was made to the designers and it had options from 1-7, being 1 "strongly disagree" and 7 "strongly agree", except for the first question of the questionnaire, where the designers were asked to evaluate themselves as designers. The question had options from 1-7, being 1 "without experience" and 7 "very experienced", and the average of those answers was 4 with a standard deviation of 1,187734939 (Appendix J).

There were 2 best rated questions (Appendix J). The first one was “It is useful.” and the second one is “I easily remember how to use it.” with the average of those answers being 5 with a standard deviation of 1,772810521 and 2,133909892, respectively.

Analyzing the answers to the most quoted questions and the designers’ comments, it can be said that the designers found that the fact that they are seeing the results of monitoring the user through an interface is quite useful because it allows them to understand where the user is having more difficulty in the interface. In that way, designers can change that section where the user is having more difficulty and improve it. As one designer wrote in positive aspects - "Helps better placing of elements, improves understanding user focus in elements" - and another designer wrote - "It can help me detect the better format to show something to the user and detect where the user normally focuses on the screen."

One of the designers also found it interesting to be merging the best parts of the interface even though it did not build an interface with context at the end - "It was an interesting idea to somehow merge the best parts of each interface (even though it takes no context) into a simpler one“.

The question with the worst average response was "It saves time when I use it" with an average of 1 with a standard deviation of 1,060660172 (Appendix J).

This question being one of the questions with the most negative answer was to be expected because the designers took between 30 and 60 minutes to analyze part of the total original optimal interface. If it was to analyze the interface that is given to the tool, which is all, the designers would take hours to analyze the points and paste all the sections. As one designer wrote - "Takes some time to get results out of it;" and another designer wrote - "Time consuming, easy to mix up". The fact that, to analyze a user's overlay, designers need to follow so many rules, having to analyze each point, analyze three interfaces and have a mesh of points that can generate confusion, may be the main reasons for this evaluation of the designers.

One of the questions asked to the designers was "It is user friendly" in which the average of the answers was 2 with a standard deviation of 1,685018016 (Appendix J). Many of the designers in the negative aspects supported this negative answer by writing - "A lot of points close to each other"; "Difficult achievement memorize the point, Line is difficult to see"; "It was very confusing when there were multiple dots in the screen even more when I had to switch between views (...)" ; "Too many points and it is difficult to spot some." ; "Too many points clustered, and even though there are some green points, they are irrelevant and only saturate the images themselves.". Given that *emotion_map*'s overlays had a large mesh of dots, it is normal

for designers to rate the overlay as poorly user friendly. In addition to each point, they were still connected by a line, which made it even more difficult to read and analyze the overlays.

To solve this problem, some designers suggested taking the positive points out. In the case of the *clickmap*, taking out the green and yellow points, and in the case of the *emotionmap* taking out the green ones - "As a suggestion, remove the green points as they are not important for the analysis."; "In order to generate less confusion in the analysis, it may not be shown the non-negative points in the *clickmap* and in the *emotionmap* (...)".

Because of the negative aspects talked about above, of all the designers' results (Appendix H), none of them gave the same results as the tool. This is associated with human error and its ability to analyze.

As already said, the sections given to the designers had a part that had a large set of points, all on top of each other and then had a more accessible part where the points were already more separated from each other (Appendix C and Appendix D).

In the most accessible part of analyzing, 75% of the designers made mistakes. Some forgot to see a point, others inserted a section that was associated with negative points and others were lost when they were passing overlays.

All designers, in the most difficult part to analyze, made mistakes because, as it was so difficult to analyze, they assumed that points that were close to each other were the same and given the large number of points there were also designers who forgot to analyze some points.

All these observations were made after analyzing the recorded videos of the designers building the optimal interface.

It is important to underline that the tool is also not perfect. It takes about 30 to 45 minutes to construct the optimal interface, when the whole interface is given, depending on the number of points that it needs to analyze. Despite being faster than the designers and being able to analyze the points perfectly, even if they are on top of each other, the tool repeats sections already inserted in the optimal interface. This is due to the fact that a library is being used that can give two different results for the same word, already described in section 5.1. and there are strings in the interface that are very similar.

Summarizing chapter 5, the overlays developed, which represent the monitorization of the users, can show where he/she clicked associated with the emotional state and can show where he/she looked associated with the cognitive state. A neuronal network is used to predict the user's cognitive state. The effectiveness of the neuronal network depends on the neuronal network precision, influenced by the way the network is trained. The optimal interface results show that it is formed by the sections where the user reacted positively but has two bugs. The

first one is that it repeats sections and the other one is that the interface has no context, the order of the sections does not make sense. However, the final results of the designers show that the tool is useful, as it is faster than designers to analyze the points, as it took approximately the same time to analyze the entire interface as the designers, who analyzed only about one tenth of the interface. The tool also analyzes with more precision each existing point as it will analyze them all, one by one and does not make the human mistakes that each designer made. Finally, the designers in the questionnaire stated that the tool was useful.

Chapter 6 – Conclusions

Finally, it is extremely important to discuss the conclusions that it was possible to reach, based on all the work described throughout this document. Thus, it is important not only to understand and discuss the most salient points, both positive and negative, to present the conclusions reached and also to talk about the limitations and possible future work.

6.1. Main conclusions

According to the final results, it can be concluded that both objectives proposed in chapter 1 were partially fulfilled.

The first objective, which was, aiming to improve the experience and usability of the end user, improving the interface design, was partially fulfilled because throughout this work a tool that monitors the users' emotions and attention was developed which selects the parts where the user reacted more positively, of three different interfaces, merging them into a new one, composed by those positive sections. However, this new interface has no context, that is, the order of the inserted sections does not make sense with each other and repeats sections. The tool improves the interface design but not as it was wanted, because it has these two bugs.

The second objective, which was, to perform a user study to verify whether the solution developed brings improvements compared to the conventional method used by the IT company, from where the designers are, was completed because, as the conventional method is very simple, where testers tell designers what, in their opinion, should change, the tool was able to select the positive sections of each interface indicating, by a colored rectangle, from what interface that positive section comes from, helping the designer. All this based on the user monitorization, something that the conventional method does not consider. Also, many users do not realize that they are in the workload state while they are in a section of an interface and will never say that this section is negative, while the tool indicates that.

In short, it can be concluded that most designers claim that this tool is useful, as can be seen in Appendix J, question number 3, where the average is 5 and the standard deviation is 1,772810521, which answers with success the research question proposed in chapter 1 – “Is a tool semi-automatically suggesting an optimized Graphical User Interface, useful for GUI designers, in relation to the physiological monitoring of users?”. The fact that they can monitor the user and then be able to see the results of that monitoring in the developed overlays was something that impressed the designers the most, but also the fact that the tool selected the positive parts of three different interfaces. However, there are not only positive points, but there

are also negative points. Most designers claim that the overlays are not user friendly due to the large amount of points very close to each other, which makes their analysis difficult and claim that, if they have to make the optimal interface manually, analyzing the data present in the overlays, they are not productive as they would take a long time, as can be seen in Appendix J, question number 5 and 9. Given that the tool was developed, this last negative point may have less weight, however, the tool has some bugs/flaws, namely that there are repeated sections and the final interface has no context. Therefore, it can be concluded that it appears, based on the qualitative results, that there is a tendency for the designers to consider the tool and the generated interface useful.

It is important to re-emphasize that the results must be carefully analyzed as the participants who tested the interfaces were known and the designers did not have access to the entire tool.

6.2. Limitations

This project was based on the development of a prototype of a tool that allows to merge the best parts or the parts where the user reacted positively of three interfaces in a single interface. But like all initial prototypes, this one has some limitations, that is, the tool only works for certain case studies. One of the limitations found during the tests is that the tool only works if the interfaces which are supposed to merge have the content sorted in the same way, but with different designs, to not generate an interface with no context. For example, a button on the right side in one interface and a button on the left side in another interface but at the same height. This was seen in the results, where interfaces with content sorted in a different way, in the end, gave an interface with no context. Another limitation is that the tool only cuts horizontally and not vertically, which limits a more accurate analysis of overlay results. It only cuts horizontally because it was developed this way. To cut vertically, more code needs to be added. Another limitation is that the tool only works if we always keep the same height across all interfaces, that is, the tool will not work with group boxes or expand buttons. It has to be that way to maintain a more accurate cutting precision. If there are group boxes that can be closed and opened during testing, if a user clicks inside a group box or looks inside a group box, the pixel value will be exactly the same if the user is clicking or looking at the group box zone, when it is closed. Thus, the analysis of the points will be wrong as it would not be possible to know if the user was clicking/looking inside or outside the group box.

Finally, and this being the most obvious limitation and one that requires a greater need for concern at an early stage, already mentioned in previous points, is the fact that in the final

interface there are repeated sections. This issue tried to be overcome using OCR as discussed in section 3.5 but there were some glitches.

6.3. Future Work

For future work, more investigation should be done, regarding applying Attentive User Interfaces and Affective Computing, to improve the interface analysis and help designers develop more user-friendly interfaces, based on monitoring the user physiology. This type of technology can be extended to many fields, can have many applications and can help designers do their work. Maybe different approaches can improve the final result of this thesis.

Regarding the tool itself, time may be invested in removing the green dots and lines as suggested by the designers. The positive points are not important for the analysis of overlays, since we are only detecting where are the negative points (on the *clickmap* overlay, it is a red point, on the *emotionmap* overlay, it is a yellow and a red point) and as the timeline widget is already available in an overlay, the one which shows the eye gaze associated with cognitive states (*emotionmap*), the lines connecting the points can also be removed. Time should also be invested in the timeline widget, to test it, as it was not possible to do so.

For the previous point to work even better, time could also be invested in joining the two overlays instead of separating them, having clicks and eye gaze in the same overlay, and adding a timestamp to the *clickmap* table.

Another possible future work would be to change the code that cuts and pastes sections to also cut vertically, instead of cutting only horizontally.

Time should also be spent investigating how to make the final interface have context, perhaps trying to divide the interface into context groups, then testing with users, monitoring their attention and emotion, and then giving these results to the tool, separately and in the end put them together. That way, maybe the interface will no longer lose context.

Above all, the main objective is that this tool can cover all types of interfaces, whether they are composed of group boxes or with different orders between each other, for example.

References

- [1] E. Horvitz, C. Kadie, T. Paek, and D. Hovel, “Models of attention in computing and communication: From principles to applications,” *Commun. ACM*, vol. 46, no. 3, pp. 52–59, 2003.
- [2] B. A. Huberman and F. Wu, “The economics of attention: Maximizing user value in information-rich environments,” *Adv. Complex Syst.*, vol. 11, no. 4, pp. 487–496, 2008.
- [3] G. Singh, S. Bermúdez i Badia, R. Ventura, and J. L. Silva, “Physiologically attentive user interface for robot teleoperation real time emotional state estimation and interface modification using physiology, facial expressions and eye movements,” *BIOSIGNALS 2018 - 11th Int. Conf. Bio-Inspired Syst. Signal Process. Proceedings; Part 11th Int. Jt. Conf. Biomed. Eng. Syst. Technol. BIOSTEC 2018*, vol. 4, no. Biostec, pp. 294–301, 2018.
- [4] R. Vertegaal, “Introduction,” *Commun. ACM*, vol. 46, no. 3, p. 30, 2003.
- [5] D. Freitas, J. E. Muñoz, and S. Bermudez I Badia, “EmoCam: capturing emotions using non-invasive technologies.”, 2017 [Online] Available: https://neurorehabilitation.m-iti.org/lab/wp-content/uploads/2017/07/Diogo_Freitas_-_Projecto__Esta%C3%ACgio_-_Sergi_Bermu%C3%ACdez_i_Badia.pdf
- [6] “Exploratory Research: Definition, Methods, Types and Examples | QuestionPro.” [Online]. Available: <https://www.questionpro.com/blog/exploratory-research/>. [Accessed: 09-Jan-2020].
- [7] “What are Research Paradigms? - Helping Research writing for student & professional researchers.” [Online]. Available: <http://researcharticles.com/index.php/what-research-paradigms/>. [Accessed: 09-Jan-2020].
- [8] “Qualitative Research: Definition, Types, Methods and Examples.” [Online]. Available: <https://www.questionpro.com/blog/qualitative-research-methods/>. [Accessed: 09-Jan-2020].
- [9] “Quantitative Research: Definition, Methods, Types and Examples | QuestionPro.” [Online]. Available: <https://www.questionpro.com/blog/quantitative-research/>. [Accessed: 09-Jan-2020].
- [10] F. I. Conference and I. V. Agents, “Affective Agents,” no. Iva, 2014.
- [11] C. L. Lisetti and F. Nasoz, “MAUI: A multimodal affective user interface,” *Proc. ACM Int. Multimed. Conf. Exhib.*, pp. 161–170, 2002.
- [13] B. Kerous, F. Skola, and F. Liarokapis, “EEG-based BCI and video games: a progress report,” *Virtual Real.*, vol. 22, no. 2, pp. 119–135, 2018.
- [15] D. Cernea, C. Weber, A. Ebert, and A. Kerren, “Emotion scents: a method of representing user emotions on GUI widgets,” *Vis. Data Anal. 2013*, vol. 8654, p. 86540F, 2013.
- [16] D. Chen and R. Vertegaal, “Using mental load for managing interruptions in Physiologically Attentive User Interfaces,” *Conf. Hum. Factors Comput. Syst. - Proc.*, no. January 2004, pp. 1513–1516, 2004.
- [17] K. Yadati, H. Katti, and M. Kankanhalli, “Interactive video advertising: A multimodal affective approach,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 7732 LNCS, no. PART 1, pp. 106–117, 2013.
- [18] V. Georges, F. Courtemanche, S. Senecal, T. Baccino, M. Fredette, and P.-M. Leger, “UX Heatmaps,” pp. 4850–4860, 2016.
- [19] K. Kucher, D. Cernea, and A. Kerren, “Visualizing Excitement of Individuals and Groups,” *Proc. EmoVis 2016, ACM IUI 2016 Work. Emot. Vis. Sonoma, CA, USA, March 10, 2016*, vol. 103, pp. 15–22, 2016.
- [20] M. V. Villarejo, B. G. Zapirain, and A. M. Zorrilla, “A stress sensor based on galvanic skin response (GSR) controlled by ZigBee,” *Sensors (Switzerland)*, vol. 12, no. 5, pp.

- 6075–6101, 2012.
- [21] M. Z. Poh, N. C. Swenson, and R. W. Picard, “A wearable sensor for unobtrusive, long-term assessment of electrodermal activity,” *IEEE Trans. Biomed. Eng.*, vol. 57, no. 5, pp. 1243–1252, 2010.
- [22] A. Nakasone, H. Prendinger, and M. Ishizuka, “ProComp Infiniti Bio-signal Encoder,” pp. 1–4.
- [23] R. N. Swamy, P. S. Mangat, K. Rajagopalan, and V. S. Parameswaran, “A discussion of the paper ‘A theory for the flexural strength of steel fibre reinforced concrete,’” *Cem. Concr. Res.*, vol. 5, no. 2, pp. 179–182, 1975.
- [24] T. Baltrusaitis, P. Robinson, and L. P. Morency, “OpenFace: An open-source facial behavior analysis toolkit,” *2016 IEEE Winter Conf. Appl. Comput. Vision, WACV 2016*, 2016, doi: 10.1109/WACV.2016.7477553.
- [25] “SMI RED – iMotions.” [Online]. Available: <https://imotions.com/hardware/smi-red/>.
- [26] R. Vertegaal, J. S. Shell, D. Chen, and A. Mamuji, “Designing for augmented attention: Towards a framework for attentive user interfaces,” *Comput. Human Behav.*, vol. 22, no. 4, pp. 771–789, 2006, doi: 10.1016/j.chb.2005.12.012.
- [27] J. H. Krantz, *Measuring behavior 2008*, vol. 41, no. 3. 2009.
- [28] J. A. Russell, “A circumplex model of affect,” *J. Pers. Soc. Psychol.*, vol. 39, no. 6, pp. 1161–1178, 1980, doi: 10.1037/h0077714.
- [29] R. D. Ward and P. H. Marsden, “Affective computing: Problems, reactions and intentions,” *Interact. Comput.*, vol. 16, no. 4, pp. 707–713, 2004, doi: 10.1016/j.intcom.2004.06.002.
- [30] V. Georges, F. Courtemanche, S. Sénécal, P.-M. Léger, L. Nacke, and M. Fredette, “The Evaluation of a Physiological Data Visualization Toolkit for UX Practitioners: Challenges and Opportunities,” *Work. Strateg. Best Pract. Des. Eval. Shar. Tech. HCI Toolkits (HCI Tools)*, no. 2, 2017.
- [31] A. P. Alves, J. Martins, H. P. Da Silva, A. Lourenço, A. Fred, and H. Ferreira, “Experimental study and evaluation of paper-based inkjet electrodes for ECG signal acquisition,” *PhyCS 2014 - Proc. Int. Conf. Physiol. Comput. Syst.*, no. January, pp. 275–281, 2014, doi: 10.5220/0004720802750281.
- [32] C. Stephanidis, “Foreword,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 9733, p. VII, 2016, doi: 10.1007/978-3-319-39513-5.
- [33] A. Němcová, L. Maršánová, and R. Smíšek, “Recommendations for Ecg Acquisition Using Bitalino,” *EEICT Conf.*, vol. 1, no. April, pp. 543–547, 2016.
- [34] C. Berka *et al.*, “EEG correlates of task engagement and mental workload in vigilance, learning, and memory tasks,” *Aviat. Sp. Environ. Med.*, vol. 78, no. 5 II, 2007.
- [35] G. Description, “Electrocardiography (ECG) Sensor Data Sheet Electrocardiography (ECG) Sensor Data Sheet,” pp. 5–6, 2020.
- [36] “Bitalino - Electrodermal Activity (EDA) Sensor User Manual.”
- [37] F. Courtemanche, P. M. Léger, A. Dufresne, M. Fredette, É. Labonté-Lemoyne, and S. Sénécal, “Physiological heatmaps: A tool for visualizing users’ emotional reactions,” *Multimed. Tools Appl.*, vol. 77, no. 9, pp. 11547–11574, 2018, doi: 10.1007/s11042-017-5091-1.
- [38] A. Tavares, “Physiologically Attentive User Interface for Robot Teleoperation,” Instituto Superior Técnico (IST), 2020.
- [39] M. Gao, P. Kortum, and F. Oswald, “Psychometric evaluation of the USE (usefulness, satisfaction, and ease of use) questionnaire for reliability and validity,” *Proc. Hum. Factors Ergon. Soc.*, vol. 3, pp. 1414–1418, 2018, doi: 10.1177/1541931218621322.

- [40] Singh, G., Bermúdez i Badia, S., Ventura, R., & Silva, J. L. (2018). Physiologically attentive user interface for robot teleoperation: real time emotional state estimation and interface modification using physiology, facial expressions and eye movements. In 11th International Joint Conference on Biomedical Engineering Systems and Technologies (pp. 294-302). SCITEPRESS-Science and Technology Publications.

Appendix

Appendix A – User's questionnaire

Dados demográficos

*Obrigatório

Sexo *

- Feminino
- Masculino

Faixa Etária *

- 18-30
- 31-40
- 41-50
- 51-60
- 61-70

Escolaridade *

- Ensino Básico (1º Ciclo)
- Ensino Básico (2º Ciclo)
- Ensino Básico (3º Ciclo)
- Ensino Secundário
- Ensino Superior
- Pós-Graduação (Mestrado, Doutoramento, Especialização)

Tem diagnóstico de alguma doença neurológica ou psiquiátrica? *

- Sim
- Não
- Prefiro Não Responder

Diariamente, quanto tempo (em horas) utiliza um computador? *

- 0
- 1-2
- 3-6
- >7

Aceita que os seus dados fisiológicos sejam recolhidos e analisados anonimamente e a sua face seja filmada mas não gravada para poder ver as suas expressões faciais? *

- Sim
- Não

Submeter

Appendix B – Script to users with tasks for them to do

Script para utilizadores

Neste teste vai ser pedido para procurar informações numa interface inspirada numa interface de recursos humanos de uma empresa real em 10 minutos. Se aparecer alguma informação seguida de "clique aqui" ou "clique no link" ou algo semelhante, não clique, não é suposto ter de abrir outra interface nem outro separador. Os únicos botões carregáveis são um botão "+" e um botão onde se lê "read more". A interface é composta por um índice, composto pelos Grupos existentes que podem ser clicados para um rápido acesso à zona do Grupo. O botão de "+" serve para expandir cada grupo, mostrando assim subgrupos e na mesma linha que um subgrupo, existe um botão "read more" que mostra a informação desse mesmo subgrupo. Quando encontrar a informação que anda à procura indique ao locutor sem tirar os olhos do ecrã.

No início irá ter de decorar um código/ID presente na interface por baixo do índice e tem 10 segundos para o fazer. Não o pode copiar. Vamos começar.

- Sou trabalhador da empresa e mudei de banco, como posso alterar o meu IBAN para receber o salário na conta nova?
- Como posso pedir para alterar o meu salário?
- Como posso comunicar uma falta?
- Onde e como me posso inscrever em cursos? Formações e Qualificação
- Quantos minutos tenho de compensar para o ano corrente?
- O que é o Performance Management Process?
- Qual o seguro da minha empresa?
- Para que numero devo ligar para saber os horários dos autocarros para o local de trabalho
- O que é o Mendix?
- Insira o código que decorou no final da interface/janela

Appendix C – Clickmap overlay example

Bem-Vindo ao AskHR – Novo serviço de contacto com o colaborador que pretende responder às dúvidas e necessidades relacionadas com os temas de Recursos Humanos

- Grupo : Acolhimento (1)
- Grupo : Aplicações RH (9)
- Grupo : Dados do Colaborador (10)
- Grupo : Deslocações e Serviço (1)
- Grupo : Férias, Faltas, Feriados e Pontas (13)
- Grupo : Formação e Qualificação (4)
- Grupo : Gestão de Tempo (5)
- Grupo : Horário de Trabalho (1)
- Grupo : Inovação e Incentivos (5)
- Grupo : MyInfo (5)
- Grupo : Segurança (1)
- Grupo : Saúde (1)

Grupo : Acolhimento (1)

Subgrupo : Bem-Vindo à Empresa (1)

O que precisa de saber?

Sessão de Acolhimento

Após a sua admissão, a Empresa irá ser convidado a participar numa sessão de acolhimento. Esta formação tem como objectivo dar a conhecer a empresa e as suas áreas de negócio, informações úteis, programas e benefícios.

Exames e Medicina do Trabalho

Os exames de Medicina do Trabalho são obrigatórios (cfr. Lei n.º 102/09, de 10 de Setembro). Como tal, no momento da admissão, será chamado ao Posto Médico a fim de realizar todos os exames e a necessidade de fazer a sua primeira (a)ra necessidade de realiza exames periodicos de 2 em 2 anos.

Grupo : Aplicações RH (9)

Subgrupo : Mailbox / Contas Funcionais (1)

Criação / Alteração Mailbox e/ou Conta Funcional

Como posso solicitar a criação de uma mailbox/conta funcional?

Para solicitar a criação de uma mailbox/conta funcional deverá preencher o template com as informações obrigatórias que se encontram a amarelo. Após o preenchimento do documento envio para a sua caixa de e-mail hr@askhr.com.

Como posso solicitar a alteração de informação de uma mailbox já criada?

Por favor enviar o pedido de alteração através do askhr@company.com. Não esquecer de sempre indicar o ID da mailbox / conta funcional bem como o do Sponsor (caso seja esta a informação).

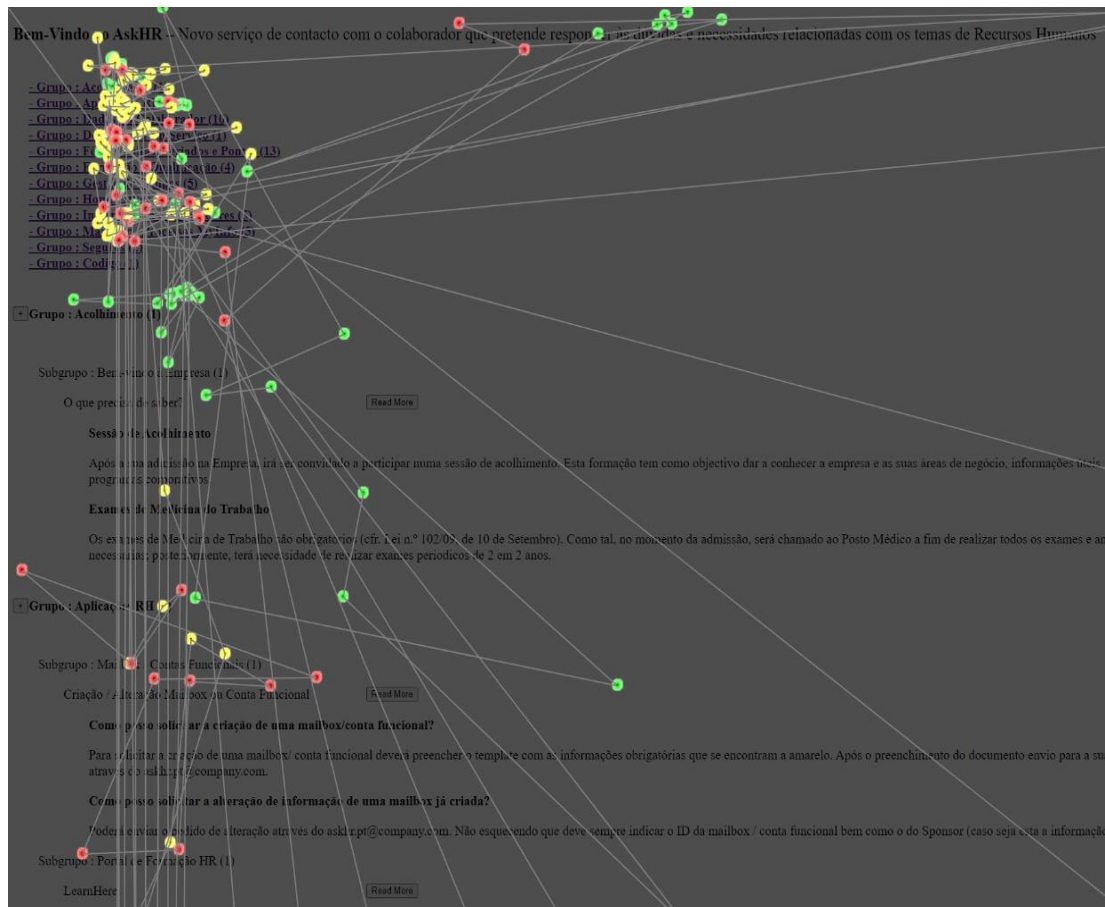
Subgrupo : Portal de Formação HR (1)

Learn Here

Learn Here

Para mais informações, por favor consulte a informação disponível na área de Formação e Qualificação – Portal de Formação HR.

Appendix D – Emotionmap overlay example



Appendix E – Final result example – Optimal interface

Bem-Vindo ao AskHR – Novo serviço de contacto com o colaborador que pretende responder às dúvidas e necessidades relacionadas com os temas de Recursos Humanos	
Bem-Vindo ao AskHR – Novo serviço de contacto com o colaborador que pretende responder às dúvidas e necessidades relacionadas com os temas de Recursos Humanos	
- Grupo : Aplicações RH (9)	
- Grupo : Home Office (1)	
- Grupo : Dados do Colaborador (10)	
- Grupo : Acolhimento (1)	
Subgrupo : Acidentes de Trabalho (1)	
Qual é a Seguradora?	
A seguradora de acidentes de trabalho é a Tranquilidade.	
O que fazer?	
No caso de acidente, em que o colaborador se possa deslocar, o mesmo deverá dirigir-se às unidades de cuidado de saúde associadas à seguradora Tranquilidade. Tenha sempre consigo o número da Apólice: - 0003703927 - company, SA - 0003704006 - company HEALTHCARE - 0003704024 - company POSTAL - 0004829713 - company MOBILITY, UNIPessoal LDA	
Onde me devo dirigir?	
Todos os sinistrados em estado grave, em risco de vida ou função (poli traumatizados, grandes queimados ou outros), devem ser enviados para o Hospital Distrital mais próximo. Na zona de Lisboa, as situações de urgência, sem risco de vida ou de função, que ocorram fora do horário de funcionamento do Centro Clínico, serão asseguradas através da urgência do Hospital da Luz Torres de Lisboa (ex British Hospital, Torre E, Piso 2). • Nas zonas de Lisboa e Porto, em caso de sinistros de menor gravidade, deve recorrer a: - Lisboa - Serviços Clínicos de assistência a sinistrados da Tranquilidade, no Hospital da Luz Torres de Lisboa. - Porto - Casa mesmo local, mas pelo serviço de urgência que depois darão seguimento para a companhia no dia seguinte.	
Tive gastos com o acidente, quem me reembolsa?	
Todas as facturas/recibos (medicamentos, táxi, bombeiros) que tenha associados ao acidente de trabalho deverá enviar diretamente para a seguradora Tranquilidade. Nota: As despesas de "Taxi" só serão indemnizadas caso exista indicação médica para o efeito, caso contrário deverá utilizar os transportes públicos.	
Como recebo o reembolso dos gastos?	
A Tranquilidade irá contactá-lo e fazer o pagamento através de transferência bancária ou através do envio de um cheque para a sua morada.	
Como informo os Recursos Humanos?	
localidade; • Em caso de hospitalização, qual o estabelecimento; • Descrição das circunstâncias em que se deu o acidente; • Natureza da lesão; • Parte do corpo lesionada: A sua assistente de RH irá preencher a declaração de participação do seguro, de acordo com as suas indicações.	
Nota: em caso de responsabilidade de terceiros, num acidente de viação, deverá entregar uma cópia da Declaração Amigável.	
Tive gastos com o acidente, quem me reembolsa?	
Todas as facturas/recibos (medicamentos, táxi, bombeiros) que tenha, associados ao acidente de trabalho, deverão igualmente ser enviadas para a seguradora.	
Subgrupo : Acidentes de Trabalho (1)	
Em caso de Acidente de Trabalho Read More	
O que fazer em caso de Sinistro?	
Tenha sempre consigo o número da apólice: company, S.A. Group 44523 policy (P002518161) company Mobility, Unipessoal Lda - group 44522 policy (P002518144) company Healthcare, Unipessoal Lda - group 44520 policy (P002518110) Sempre que exista algum acidente em Viagem de serviço no estrangeiro, os colaboradores (ou na sua impossibilidade, quem os possa representar) deverão ligar para a linha da Allianz disponível 24/7; Telefone: +353 9 876 5432 ou contactar através de; Email: client.services@company.com	
- Grupo : Manager - Processos MyInfo (5)	
Subgrupo : Mailbox / Contas Funcionais (1)	
Read More Criação / Alteração Mailbox ou Conta Funcional	
Como posso solicitar a criação de uma mailbox/conta funcional?	
Para solicitar a criação de uma mailbox/ conta funcional deverá preencher o template com as informações obrigatórias que se encontram a amarelo. Após o preenchimento do documento envio para a sua assistente HR através do askhr.pt@company.com.	
Como posso solicitar a alteração de informação de uma mailbox já criada?	
Poderá enviar o pedido de alteração através do askhr.pt@company.com. Não esquecendo que deve sempre indicar o ID da mailbox / conta funcional bem como o do Sponsor (caso seja esta a informação a alterar).	
Subgrupo : Portal de Formação HR (1)	
Read More LearnHere	
LearnHere	
Para mais informações, por favor consulte a informação disponível na área de Formação e Qualificação – Portal de Formação HR.	
Read More Fiori Mobile	
Como posso aceder ao CompanyPortal através do meu Tablet ou Smartphone?	
acesso ao CompanyPortal e um outro que explica como instalar a aplicação AirWatch: CompanyPortal Mobile - Guia de Acesso; AirWatch Manual para IOS Manual para Android Poderá igualmente assistir a um vídeo explicativo.	
Read More Mudança Organizacional	
Como posso solicitar uma mudança Organizacional?	

Appendix F – Designer’s questionnaire

Designers questionnaire

Please rate your agreement with these statements and try to respond to all the items.

*Obrigatório

How do you consider your experience as a designer? *

	1	2	3	4	5	6	7	
Without experience	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	Very experienced

Seguinte

USEFULNESS

1. It helps me be more effective. *

1 2 3 4 5 6 7

strongly disagree strongly agree

2. It helps me be more productive. *

1 2 3 4 5 6 7

strongly disagree strongly agree

3. It is useful. *

1 2 3 4 5 6 7

strongly disagree strongly agree

4. It makes the things I want to accomplish easier to get done. *

1 2 3 4 5 6 7

strongly disagree strongly agree

5. It saves me time when I use it. *

1 2 3 4 5 6 7

strongly disagree strongly agree

6. It meets my needs. *

1 2 3 4 5 6 7

strongly disagree strongly agree

Anterior

Seguiente

EASE OF USE

7. It is easy to use. *

	1	2	3	4	5	6	7	
strongly disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	strongly agree

8. It is simple to use. *

	1	2	3	4	5	6	7	
strongly disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	strongly agree

9. It is user friendly. *

	1	2	3	4	5	6	7	
strongly disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	strongly agree

Anterior

Seguinte

EASE OF LEARNING

10. I learned to use it quickly. *

	1	2	3	4	5	6	7	
strongly disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	strongly agree

11. I easily remember how to use it. *

	1	2	3	4	5	6	7	
strongly disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	strongly agree

12. It is easy to learn to use it. *

	1	2	3	4	5	6	7	
strongly disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	strongly agree

Anterior

Seguinte

SATISFACTION

13. I am satisfied with it. *

	1	2	3	4	5	6	7	
strongly disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	strongly agree

14. I would recommend it to others. *

	1	2	3	4	5	6	7	
strongly disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	strongly agree

15. I feel I need to have it. *

	1	2	3	4	5	6	7	
strongly disagree	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	strongly agree

Anterior

Seguiente

OBSERVATIONS

List the most negative aspect(s) (max 3):

A sua resposta

List the most positive aspect(s) (max 3):

A sua resposta

Anterior

Submeter

Appendix G – Designer’s script

Script para designers

Foram-te dadas 9 imagens, 3 delas são interfaces originais e 6 delas são os resultados de utilizadores que usaram essas interfaces para realizar x tarefas.

Destas 6, os resultados, 3 representam os cliques associados as expressões faciais do utilizador - clickmap e outras 3 representam o eye gaze ou olhar associado ao estado cognitivo do utilizador - emotionmap

Nos clickmap, o ponto verde é uma expressão facial positiva, o ponto amarelo uma expressão facial neutra e o ponto encarnado é uma expressão facial negativa

Nos emotionmap, o ponto verde representa o estado rest, o amarelo representa o estado de workload e o encarnado representa o estado de intenso workload.

O que te é pedido é que, observando estes 6 resultados consigas fazer um merge da melhor interface, um merge entre as três interfaces originais

Regras:

Deves começar sempre por observar os resultados da interface 1, ou seja, observar em conjunto o clickmap interface v1 e emotionmap_interface_v1 e depois passar para a 2 depois para a 3 e depois regressas a 1.

Mas como é que este processo todo funciona?

Começando em cima nos resultados da interface 1, canto superior esquerdo, verticalmente comesças a descer até encontrar um ponto negativo. Estes podem ser uma expressão facial negativa (encarnado no clickmap), ou workload ou intenso workload (amarelo ou encarnado no emotionmap, respetivamente)

Quando encontras um ponto negativo isto quer dizer que tudo o que está para cima estava positivo, logo podes recortar essa porção da interface (apenas recortas horizontalmente). Quando digo interface, é mesmo para recortar da interface original correta, ou seja, se estavas nos resultados da interface 1 vais recortar da interface 1 original. Recorta com uma ferramenta à tua escolha e depois cola no paint, pintando de lado (fazendo um risco vertical) de qual interface tiraste essa porção

Encarnado - Interface 1

Amarelo - Interface 2

Verde - Interface 3

Depois passas para a interface 2 fazendo o mesmo processo e depois para a 3 e depois para a 1 e assim sucessivamente, ate acabares em pelo menos uma das interfaces

É importante decorar que pontos já "viste" em cada um dos resultados das interfaces para saber onde começar a análise quando regressas a um resultado das interfaces já vistas.

Quando estás a recortar também é preciso de ter em atenção para não estar a repetir secções já coladas na interface final (merge das três interfaces)

PS: É importante, quando estás a ver se há secções repetidas, se por acaso a secção onde se esta a pegar tem pontos negativos

Duvidas?

No final vais responder a um questionário

Aceitas que estes testes sejam gravados, o que inclui o teu ecrã e a tua voz para uma futura análise?

Appendix H – Designers manual results

Bem-Vindo ao AskHR – Novo serviço de contacto com o colaborador que pretende responder às dúvidas e necessidades relacionadas com os temas de Recursos Humanos

- Grupo : Aplicações RH (9)

- Grupo : Dados do Colaborador (10)

- Grupo : Seguros (2)

- Grupo : Manager - Processos MvInfo (5)

- Grupo : Férias, Faltas, Feriados e Pontes (13)

- Grupo : Formação e Qualificação (4)

- Grupo : Seguros (2)

- Grupo : Código (1)

Grupo : Aplicações RH (9)

Subgrupo : Mailbox / Contas Funcionais (1)

Read More Criação / Alteração Mailbox ou Conta Funcional

Grupo : Acolhimento (1)

Como posso solicitar a criação de uma mailbox/conta funcional?

O que precisa de saber?

Read More

Sessão de Acolhimento

Grupo : Seguros (2)

Após a sua admissão na Empresa, irá ser convidado a participar numa sessão de acolhimento. Esta formação tem como objectivo dar a conhecer a empresa e as suas áreas de negócio, informações úteis, regras internas e programas corporativos.

Para solicitar a criação de uma mailbox/ conta funcional deverá preencher o template com as informações obrigatórias que se encontram a amarelo. Após o preenchimento do documento envio para a sua assistente HR

Exames de Medicina do Trabalho

Os exames de Medicina de Trabalho são obrigatórios (cfr. Lei n.º 102/09, de 10 de Setembro). Como tal, no momento da admissão, será chamado ao Posto Médico a fim de realizar todos os exames e análises clínicas necessárias; posteriormente, terá necessidade de realizar exames periódicos de 2 em 2 anos.

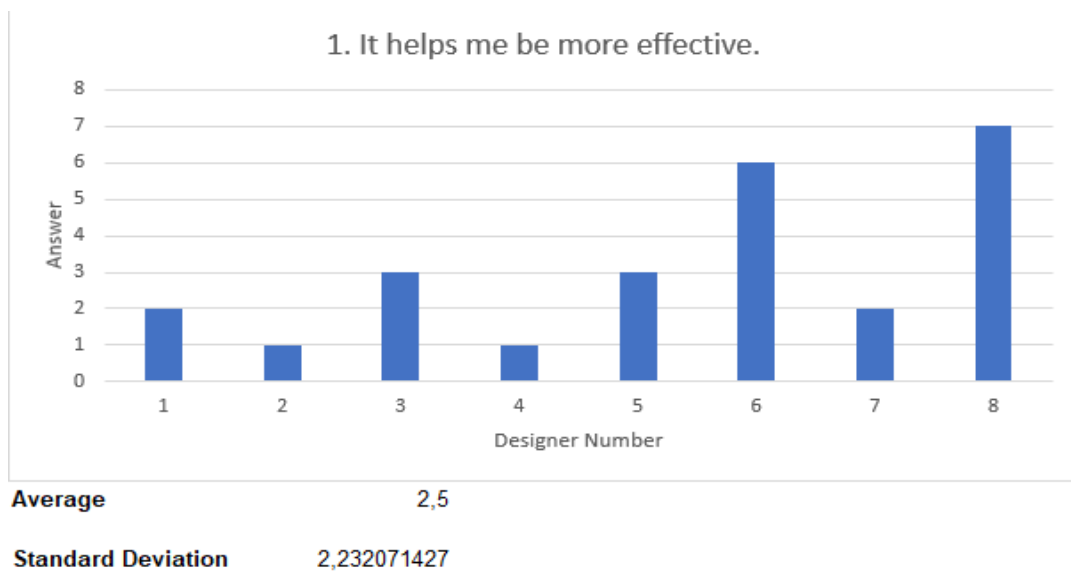
Como posso solicitar a alteração de informação de uma mailbox já criada?

Poderá enviar o pedido de alteração através do askhr.pt@company.com. Não esquecendo que deve sempre indicar o ID da mailbox / conta funcional bem como o do Sponsor (caso seja esta a informação a alterar).

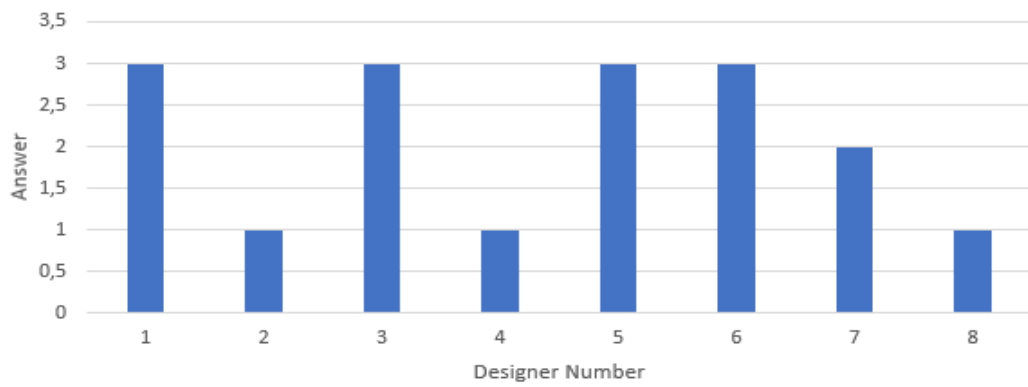
Appendix I – EEG Devices [21]

Device	Channels	ADC Bits	Sample Rate	Motion Sensors	Battery Length	Cost Jan 2017
Muse 2016	4-6	12	256 Hz	3 axis	5 hours	\$200
Epoc	14+2 ref	16	256 Hz	9 axis	6 hours Using BTLE	\$799
Insight	5+2 ref	15	2048 Hz	9 axis	4 hours Using BT	\$300
OPenBCI	Up to 16 Channels	24	230 Hz	3 axis	~26 hours	\$500(8c ha) \$946(1 6cha)
Neurosky Mindwave	1+1 Ref	12	512 Hz	N/A	8 hours	\$99.99

Appendix J – Designer’s Results



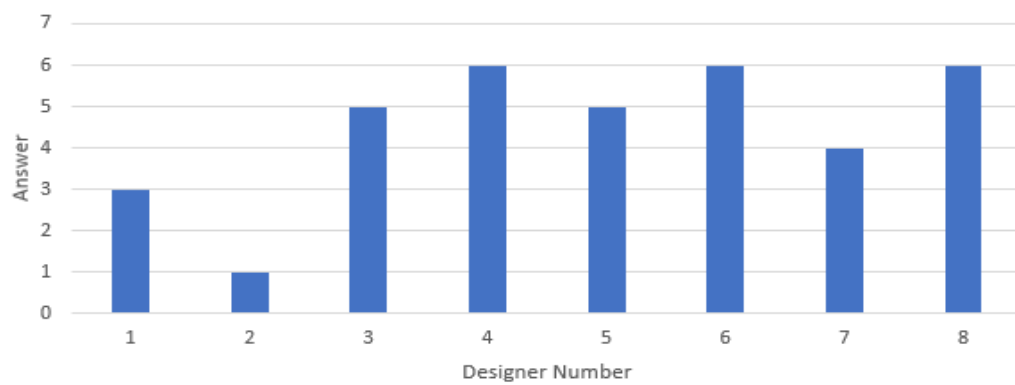
2. It helps me be more productive.



Average 2,5

Standard Deviation 0,991031209

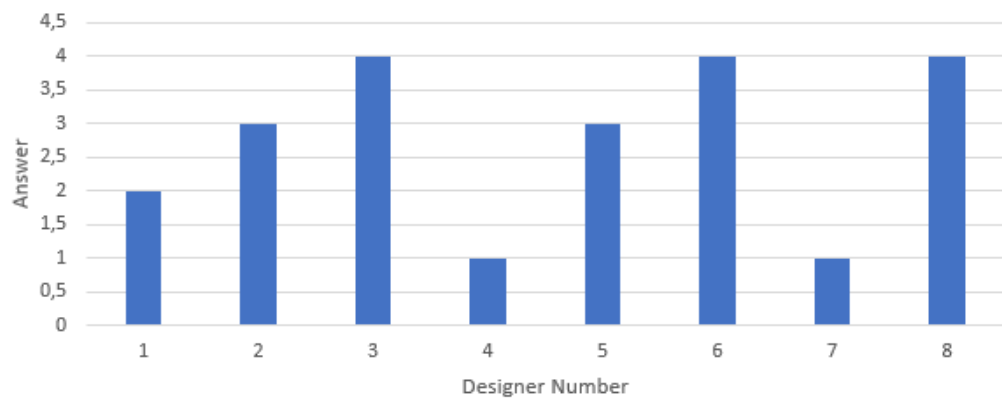
3. It is useful.



Average 5

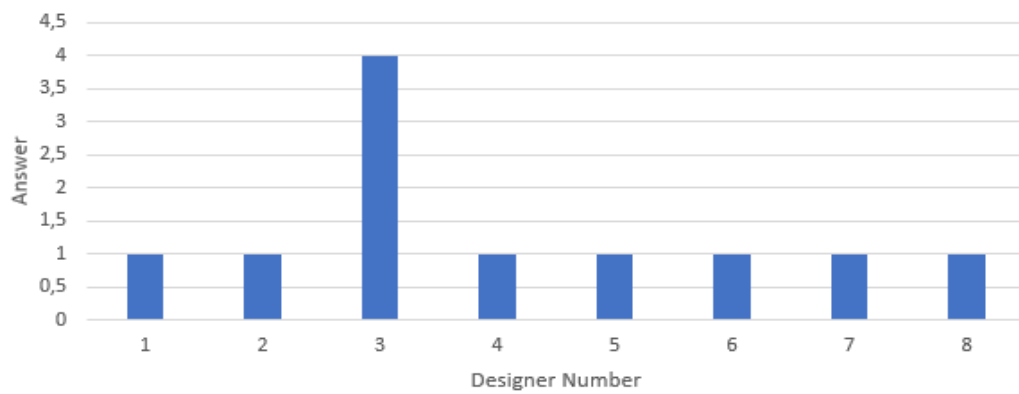
Standard Deviation 1,772810521

4. It makes the things I want to accomplish easier to get done.

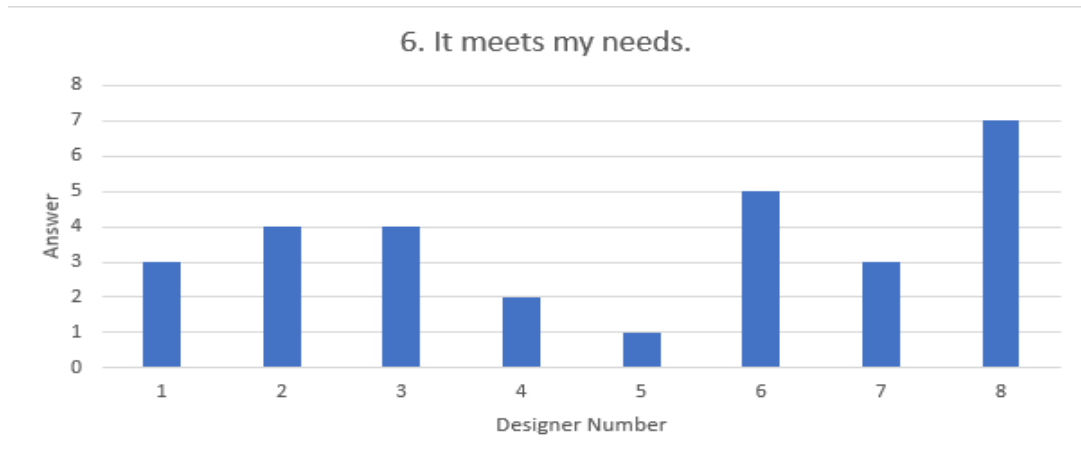


Average 3
Standard Deviation 1,281739889

5. It saves me time when I use it.

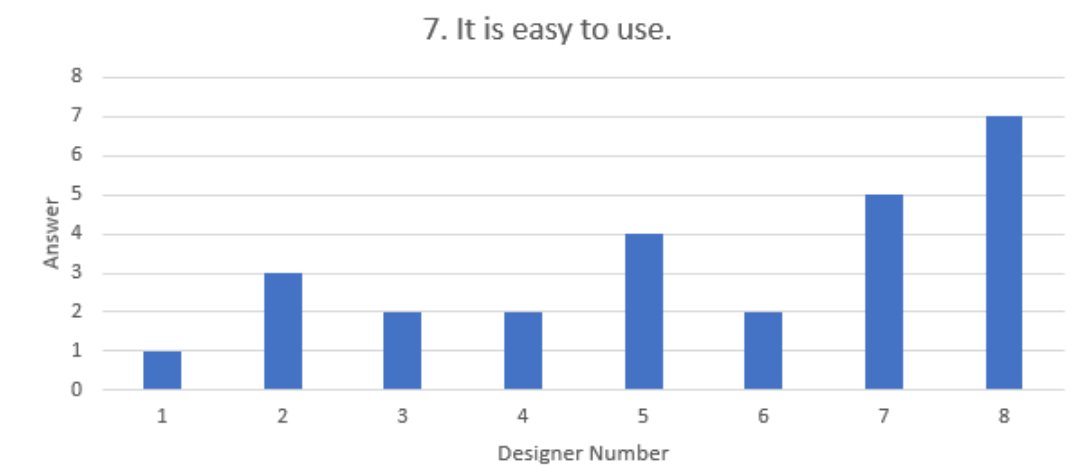


Average 1
Standard Deviation 1,060660172



Average 3,5

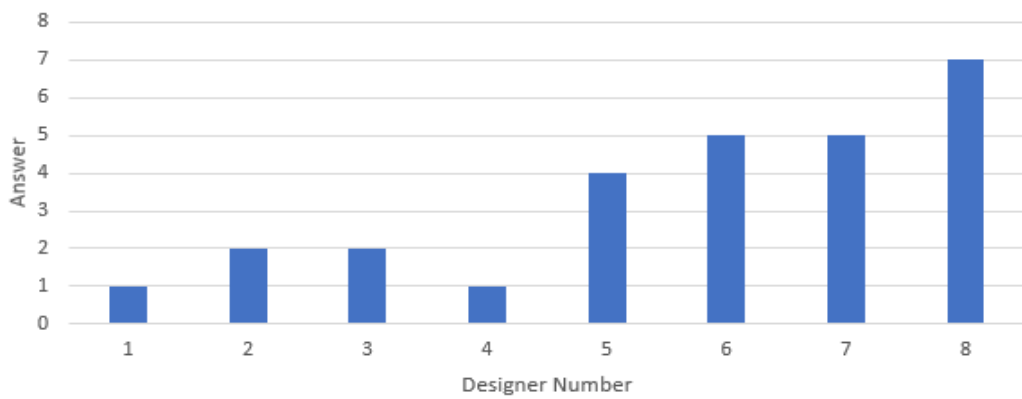
Standard Deviation 1,846811925



Average 2,5

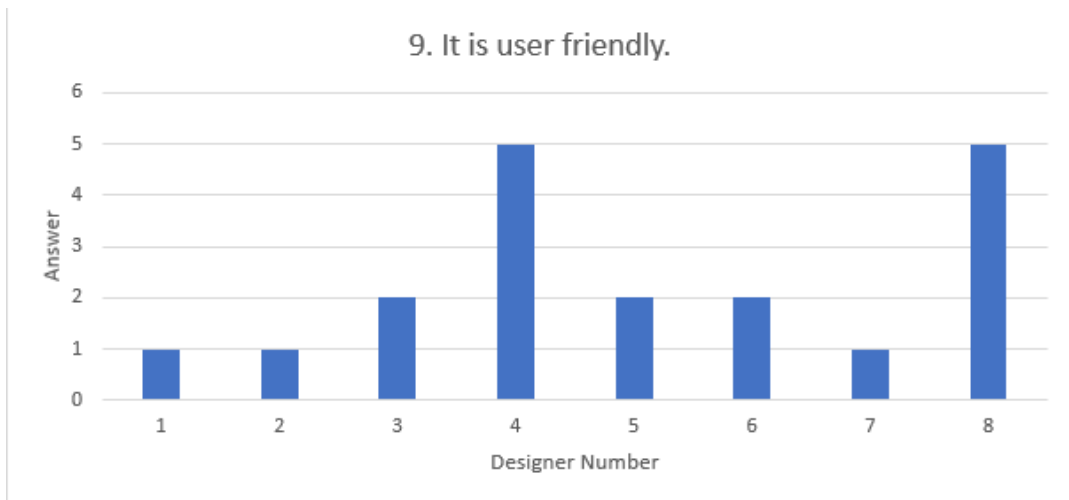
Standard Deviation 1,982062418

8. It is simple to use.



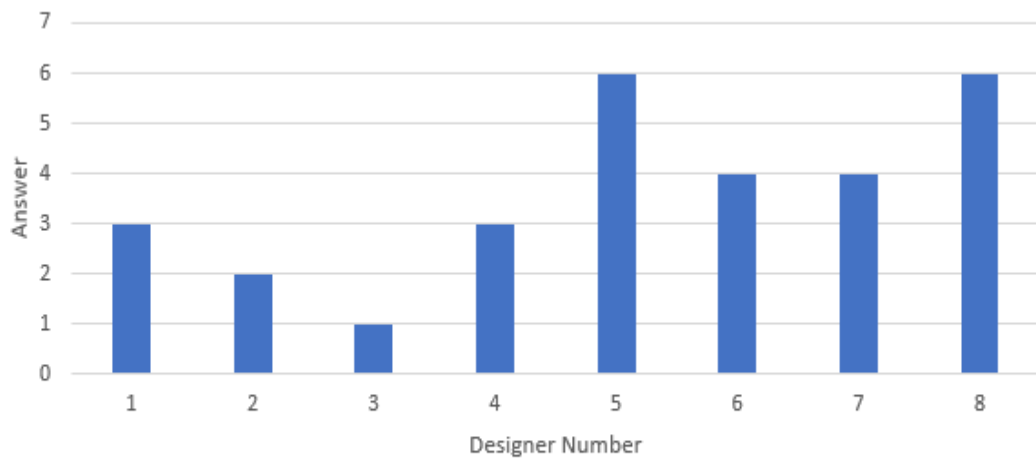
Average 3
Standard Deviation 2,199837656

9. It is user friendly.



Average 2
Standard Deviation 1,685018016

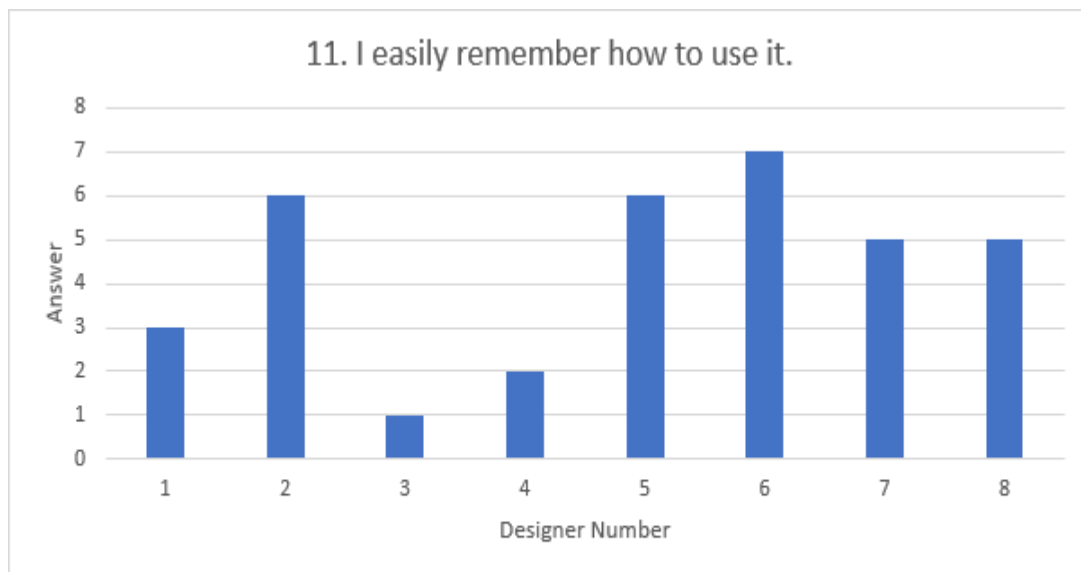
10. I learned to use it quickly.



Average 3,5

Standard Deviation 1,767766953

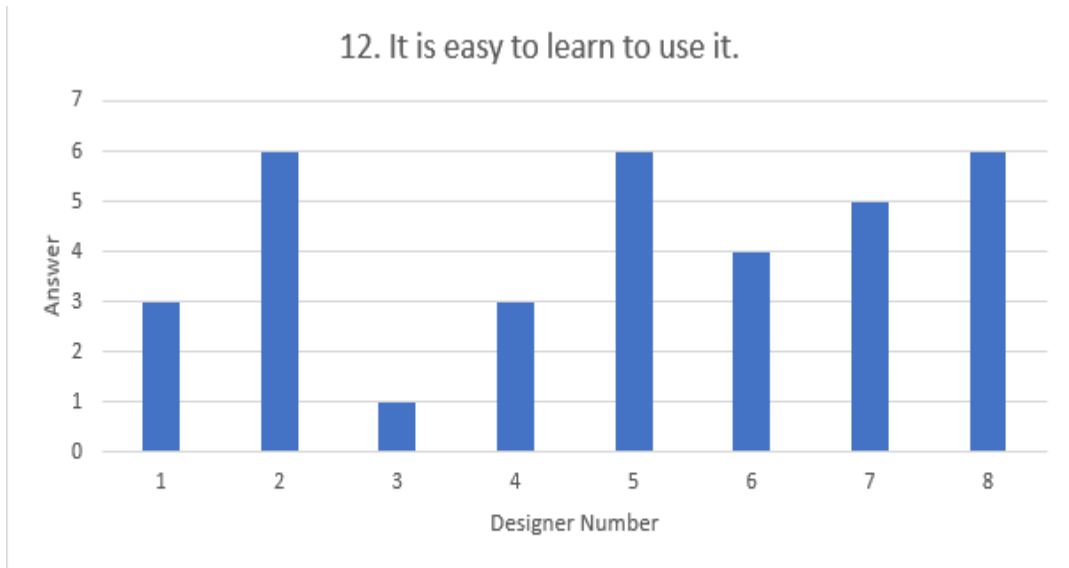
11. I easily remember how to use it.



Average 5

Standard Deviation 2,133909892

12. It is easy to learn to use it.



Average 4,5

Standard Deviation 1,832250763

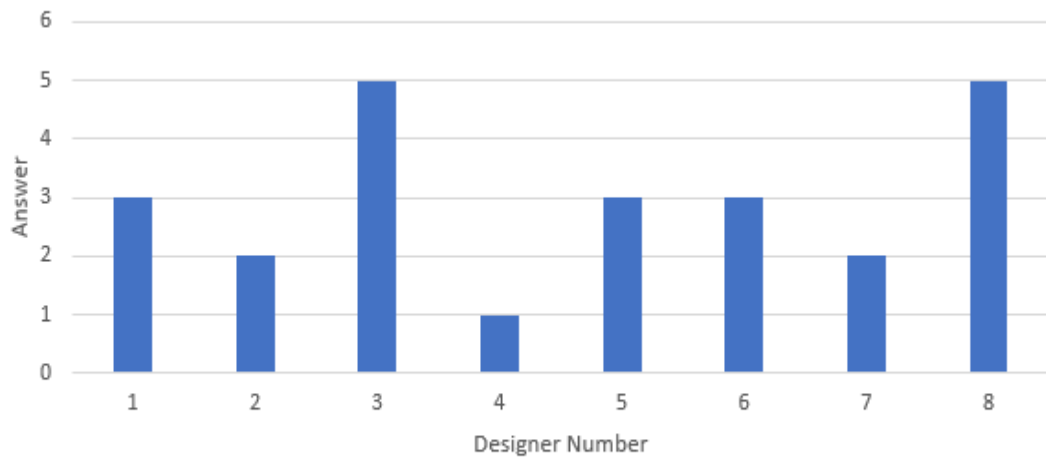
13. I am satisfied with it.



Average 3

Standard Deviation 1,922609833

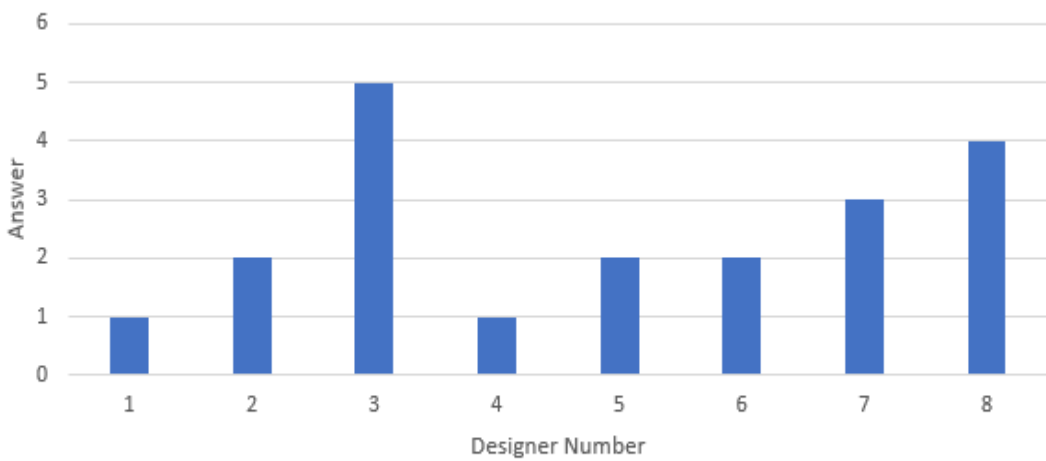
14. I would recommend it to others.



Average 3

Standard Deviation 1,414213562

15. I feel I need to have it.



Average 2

Standard Deviation 1,414213562

Appendix L - Non-inserted sections interface

[- Grupo : Férias, Faltas, Feriados e Pontes \(13\)](#)
[- Grupo : Férias, Faltas, Feriados e Pontes \(13\)](#)
[- Grupo : Férias, Faltas, Feriados e Pontes \(13\)](#)
[- Grupo : Férias, Faltas, Feriados e Pontes \(13\)](#)
[- Grupo : Férias, Faltas, Feriados e Pontes \(13\)](#)
[- Grupo : Férias, Faltas, Feriados e Pontes \(13\)](#)
[- Grupo : Férias, Faltas, Feriados e Pontes \(13\)](#)
[- Grupo : Férias, Faltas, Feriados e Pontes \(13\)](#)
[- Grupo : Férias, Faltas, Feriados e Pontes \(13\)](#)
[- Grupo : Férias, Faltas, Feriados e Pontes \(13\)](#)
[- Grupo : Férias, Faltas, Feriados e Pontes \(13\)](#)
[- Grupo : Férias, Faltas, Feriados e Pontes \(13\)](#)
[- Grupo : Formação e Qualificação \(4\)](#)
[- Grupo : Formação e Qualificação \(4\)](#)

[Read More](#) Informação Geral
[Read More](#) Informação Geral
[Read More](#) Funcionalidades do Portal - Chefias
[- Grupo : Férias, Faltas, Feriados e Pontes \(13\)](#)
[- Grupo : Férias, Faltas, Feriados e Pontes \(13\)](#)
[- Grupo : Férias, Faltas, Feriados e Pontes \(13\)](#)
[- Grupo : Férias, Faltas, Feriados e Pontes \(13\)](#)

[Read More](#) Comunicação das Faltas
Fiori Mobile [Read More](#)
[Read More](#) Feriados

O que são as Pontes?
[Read More](#) LearnHere
[Read More](#) EJE Travel@company - Portugal
[Read More](#) GS | New Bus Service
Pedido de Declarações [Read More](#)
[Read More](#) Ausências em Serviço

O sistema contabiliza 8 horas (mais os minutos de compensação para as "pontes") de trabalho por dia, até ao registo seguinte.
prémios disponíveis)

Como posso Comunicar?

- Grupo : Formação e Qualificação (4)

[Read More](#) Inserir Código
Codigo:

[Submit](#)

[Read More](#) Inserir Código
Codigo:

[Submit](#)

[Read More](#) Inserir Código
Codigo:

[Submit](#)

Consulte aqui, as regras sobre a prevenção e proteção de assédio no trabalho em complemento das Business Conduct Guidelines.

[Read More](#) Inserir Código
Codigo:

[Submit](#)

[Read More](#) Inserir Código
Codigo:

[Submit](#)

Em caso de dúvidas, quem devo contactar?
O sistema contabiliza 8 horas (mais os minutos de compensação para as "pontes") de trabalho por dia, até ao registo seguinte.
Salvo regimes especiais de horários já existentes, o regime de flexibilidade de horário em vigor na empresa considera o tempo de serviço efectivo de segunda a sexta-feira, efectuado no período de referência
Salvo regimes especiais de horários já existentes, o regime de flexibilidade de horário em vigor na empresa considera o tempo de serviço efectivo de segunda a sexta-feira, efectuado no período de referência
Salvo regimes especiais de horários já existentes, o regime de flexibilidade de horário em vigor na empresa considera o tempo de serviço efectivo de segunda a sexta-feira, efectuado no período de referência
Salvo regimes especiais de horários já existentes, o regime de flexibilidade de horário em vigor na empresa considera o tempo de serviço efectivo de segunda a sexta-feira, efectuado no período de referência

COVID-19 [Read More](#)
prémios disponíveis)

Tive gastos com o acidente, quem me reembolsa?
Como posso Comunicar?