FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Altycs: um componente visual de métricas de negócio

Lucas Martins



Mestrado em Engenharia de Software

Supervisor: Nuno Flores

29 de outubro de 2021

Altycs: um componente visual de métricas de negócio

Lucas Martins

Mestrado em Engenharia de Software

29 de outubro de 2021

Abstract

This thesis arises from the need to provide a component that represents in a way *user-friendly* the business metrics in order to support management and decision making actions.

The problem is embedded in a telecommunications business context in conjunction with the company Altice Labs. The company's platform ABC (*Advanced Business Communication*) presents a way of viewing the information generated by the user's business in a form of an unfiltered and decontextualized list, like a "printout"of data.

A list that prints out all the recorded data is an outdated way of representing business metrics. As the telecommunications environment generates large amounts of data, it is necessary that this data be organized so that it can then be more easily consulted.

To offer a more effective tool for data analysis, a dashboard-like solution was devised. In this way, users have at their disposal a control panel that allows them to more easily access the data and helps in the interpretation of the information presented.

To build the component's *front-end*, state-of-the-art technologies were surveyed and *Angular* was chosen.

SCRUM was adopted as the development technology (assisted by an ALM app, called CLICKUP), having produced a *product backlog* in form of *user stories* to define the functionalities that the component should offer. Also designed were *mockups* to have an idea of the visual impact that the component should follow and support the development.

To validate the solution, usability tests were carried out, where users answered questionnaires after coming into contact with the new component. The questions aimed at understanding if the component allows a good usability and if the information provided is well understood. In total, 16 users participated, and gave an average rating of 1.33 on a scale of 1 (easy) to 5 (hard) regarding the difficulty of using the component. A screening was also done to discard possible external threats to the results.

Keywords: Dasboard, business metrics, data visualization

Resumo

Esta tese surge da necessidade de fornecer um componente que represente de forma *user-friendly* as métricas do negócio, a fim de apoiar ações de gestão e tomada de decisão.

O problema está inserido num contexto de negócios de telecomunicações em conjunto com a empresa Altice Labs. A empresa tem a plataforma ABC (*Advanced Business Communication*) que apresenta uma forma de visualizar as informações produzidas pelo negócio do utilizador na forma de listagem sem permitir filtragem de dados e descontextualizada, como uma "impressão"de dados.

Uma lista que imprime todos os dados registados é uma forma desatualizada de representar as métricas de negócios. Como o ambiente de telecomunicações gera grandes quantidades de dados, é necessário que esses dados sejam organizados para que possam ser mais facilmente consultados.

Para oferecer uma ferramenta mais eficaz para a análise de dados, desenvolveu-se uma solução semelhante a um painel de controlo. Desta forma, o utilizador dispõe de um painel de controle que permite um acesso mais fácil aos dados e que auxilia na interpretação da informação apresentada.

Para construir o *front-end* do componente, tecnologias de ponta foram pesquisadas e o *Angular* foi escolhido.

O *SCRUM* foi adotado como tecnologia de desenvolvimento (auxiliado por um aplicativo ALM, denominado CLICKUP), tendo-se produzido um textit product backlog na forma de *user stories* para definir as funcionalidades que o componente deve oferecer. Também foram projetados *mockups* para se ter uma ideia do impacto visual que o componente deve acompanhar e apoiar o desenvolvimento.

Para validar a solução, foram realizados testes de usabilidade, onde os utilizadores responderam a questionários após entrarem em contato com o novo componente. As perguntas visam entender se o componente permite uma boa usabilidade e se as informações fornecidas são bem compreendidas. No total, participaram 16 utilizadores, que atribuíram uma nota média de 1,33 em uma escala de 1 (fácil) a 5 (difícil) quanto à dificuldade de uso do componente. Uma triagem também foi feita para descartar possíveis ameaças externas aos resultados.

Keywords: Dasboard, métricas de negócio, visualização de dados

Agradecimentos

O primeiro agradecimento vai para o meu orientador prof. Nuno Flores que fez um trabalho de orientação excecional. Foi desde o início do processo uma pessoa prestável, acessível e que me transmitiu conhecimentos que levo para a vida não só profissional como pessoal. Posso afirmar com toda a certeza que seria impossível ter um orientador melhor.

Agradecer á Altice Labs pela oportunidade de trabalhar com eles. Uma experiência fantástica graças a todos os profissionais que lá trabalham e que me fizerem sentir muito acarinhado e bem-vindo. Deram toda a ajuda possível e necessária para que todas as dificuldades fossem superadas. A todos os profissionais com quem tive contacto o meu obrigado. Agradeço em particular ao Célio Abreu por me ter recebido e introduzido numa equipa soberba. Nunca deixou que ficasse alguma questão por resolver e disponibilizou-se sempre para me ajudar a qualquer momento. Trabalhou sempre lado a lado comigo fazendo com que todas as dificuldades fossem ultrapassadas mais facilmente graças á sua experiência e conhecimento.

Aos meus pais que me deram todas as condições necessárias. Nunca deixando que nada me faltasse. Agradeço tudo o que fizerem por mim e espero que através destas palavras possam sentir todo o amor que sinto por eles. Á minha irmã pelo apoio e carinho. Á Joana Carvalho por não permitir que perdesse o foco e pelo seu suporte. Á minha família em geral por serem formidáveis comigo. Agradeço as tentativas de tentarem compreender o meu trabalho e falarem sobre ele, apesar de eu saber que continuam sem o perceber.

Não posso deixar passar a oportunidade para agradecer ao Nelson Novais que me tem dado conselhos preciosos e por me transmitir toda a sua experiência. Despertou o meu interesse por esta área e é sem dúvida graças a ele que hoje cheguei até aqui.

A todos os meus amigos que conheci durante o percurso académico. Formaram-se laços que ficarão para sempre. Agradeço á Micaela Pinto que me ajudou quando precisei da sua ajuda. Ao Rui Santhim por se ter arrastado comigo para as aulas de álgebra linear sem me deixar desistir. Aos restantes, Luís Rodrigues, Jorge Magalhães, Ricardo Sousa, Xavier Teixeira, Filipe Batista, Eduardo Machado, Telmo Cardoso, Paulo Barros etc. Ao maior, Rui Moreira.

A todas as equipas de arbitragem que estiverem comigo e fizeram com fosse mais fácil conciliar os estudos com o desporto. Uma relação que vai além das quatro linhas. A toda esta família obrigado.

A todas estas pessoas o meu enorme agradecimento.

Lucas Martins

"É difícil superar,
quem nunca desiste."

Lucas Martins

Conteúdo

1 Introdução **1**
- 1.1 Contexto . . . 1
- 1.2 Problema na visualização das métricas . . . 2
- 1.3 Novo componente de visualização . . . 2
- 1.4 Metodologia . . . 2
- 1.5 Estrutura do documento . . . 3

2 Visualização de informação **5**
- 2.1 Informação em tempo real e dados histórios . . . 5
- 2.2 Visualização dos dados . . . 7
  - 2.2.1 Impacto da visualização de dados na area dos negócios . . . 7
  - 2.2.2 Analise nos bastidores . . . 8
  - 2.2.3 Métricas diferentes significam diferentes tipos de visualização . . . 8
  - 2.2.4 Importância da personalização . . . 10
- 2.3 Interação Pessoa-Computador . . . 11
  - 2.3.1 Influência do contexto . . . 11
  - 2.3.2 Formas de interação . . . 12
  - 2.3.3 Ações implícitas . . . 12
  - 2.3.4 Formas de medir uma boa interação . . . 13
  - 2.3.5 Evolução e perspetiva futura . . . 15
- 2.4 Conclusão . . . 16

3 Visualização de informação na plataforma ABC **19**
- 3.1 Registo de dados . . . 19
- 3.2 Grupos de dados . . . 20
  - 3.2.1 SMS and Calls . . . 20
  - 3.2.2 Hunt Group . . . 20
  - 3.2.3 Waiting Queue . . . 20
- 3.3 Organização dos dados nas Base de Dados . . . 21
- 3.4 Problemas na visualização dos dados . . . 21
- 3.5 Visão de solução . . . 22
- 3.6 Validação . . . 23

4 Altycs **25**
- 4.1 Tecnologias . . . 25
  - 4.1.1 Data Binding . . . 25
  - 4.1.2 Performance e desempenho . . . 27
  - 4.1.3 Comparação de outros fatores . . . 28

4.1.4 Escolha da tecnologia . . . 28
4.2 Impacto visual . . . 30
4.2.1 Personalização . . . 30
4.2.2 User-Friendly . . . 31
4.3 Desenho do componente . . . 32
4.3.1 Product Backlog . . . 32
4.3.2 Mockups . . . 32
4.4 Metodologia para o desenvolvimento . . . 34
4.5 Divisão do problema . . . 35
4.6 Bibliotecas . . . 36
4.6.1 Gridster2 . . . 37
4.6.2 ApexCharts . . . 37
4.6.3 Node.js . . . 39
4.6.4 Knex.js . . . 40
4.7 Idealização da estrutura . . . 40
4.7.1 Passagem de dados . . . 41
4.7.2 Personalização da grelha . . . 41
4.7.3 Enquadramento de um gráfico na janela de widget . . . 42
4.7.4 Menu para adicionar um widget . . . 43
4.8 Dashboard . . . 43
4.9 Sumário . . . 44

5 Testes de usabilidade **47**
5.1 Conceção e planeamento . . . 47
5.1.1 Recolha de dados de usabilidade . . . 48
5.1.2 Inquérito de despistagem . . . 49
5.2 Resultados . . . 49
5.2.1 Análise do teste de usabilidade . . . 50
5.2.2 Avaliação da despistagem . . . 60
5.3 Sumário . . . 67

6 Conclusões **69**
6.1 Contribuições . . . 69
6.2 Trabalho futuro . . . 70

Referências **73**

Lista de Figuras

2.1 Possível amostragem dos resultado dos lançamentos de um dado 9
2.2 Possível amostragem de dados com centenas de variáveis 9
2.3 Ícone associado á abertura de menus 13
2.4 Exemplo de um *Eye Tracking* 14
2.5 Relação entre tempo-real,visualização de dados e HCI 17

4.1 Funcionamento *One Way Data Biding* 26
4.2 HTML com chamada da função *"myFunction()"* 26
4.3 Função no componente 26
4.4 Funcionamento *Two Way Data Biding* 27
4.5 Funcionamento *One Way Data Biding* e *Two Way Data Biding* 28
4.6 Comparação entre *React, Vue* e *Angular* 29
4.7 Mockup do menu do dashboard 33
4.8 Mockup para adicionar um widget 34
4.9 Mockup com opções de visualização 35
4.10 Mockup com opções de download 36
4.11 Gridster2 37
4.12 ApexChart 38
4.13 Exemplo do formato de dados utilizando ApexCharts 38
4.14 Exemplo da legenda utilizando ApexCharts 39
4.15 Exemplo de um gráfico utilizando ApexCharts 39
4.16 Exemplo de uma querie utilizando Knex.js 40
4.17 Opções de personalização da grelha 42
4.18 Área ocupada pelos constituintes de um widget 43
4.19 Menu para adicionar um widget 44
4.20 Dashboard 45

5.1 Linha orientadora para a execução da recolha de dados 48
5.2 Tempo médio e desvio padrão por pergunta em segundos 51
5.3 Dificuldade média por pergunta 51
5.4 Frequência da dificuldade na questão 1 52
5.5 Dificuldade na questão 3 53
5.6 Tempo de execução da pergunta 6 e 9 57
5.7 Tempo de execução da pergunta 4 e 12 58
5.8 Frequência de respostas á questão 1 de despistagem 61
5.9 Percentagem de respostas á questão 5 de despistagem 63
5.10 Condições de realização do teste 65

Lista de Tabelas

5.1 Média e desvio padrão dos tempos de resposta 50
5.2 Dificuldade atribuída por cada utilizador á questão 2 53
5.3 Dificuldade na questão 4 54
5.4 Respostas com nível de dificuldade superior a 3 55
5.5 Taxa de concretização por pergunta 59
5.6 Resposta esperada, resposta obtida e desvio padrão por questão de despistagem . 60
5.7 Número de respostas á questão 2 de despistagem 61
5.8 Respostas á questão 4 de despistagem 62
5.9 Respostas á questão 6 de despistagem 63
5.10 Tempos de resposta dos utilizadores com influências externas 64
5.11 Tempos de resposta do utilizador 6 que ultrapassam o tempo médio 64
5.12 Tempos de resposta do utilizador 9 que ultrapassam o tempo médio 65
5.13 Tempos de resposta do utilizador 15 que ultrapassam o tempo médio 66
5.14 Dados removendo as influências do User6, User9 e User15 67
5.15 Tempos de resposta dos utilizadores com agenda apertada 68

Acrónimos

ABC	Advanced Business Communication
ALM	Application Lifecycle Management
BD	Base de dados
HCI	*Human–computer interaction*
IBM	*International Business Machines*
km	Quilómetros
PC	Computador
SPA	*Single Page Application*
TI	Tecnologias de Informação
WWW	*World Wide Web*

Capítulo 1

Introdução

O ambiente corporativo, enquadrado nas telecomunicações e conjugado com o aparecimento de novos serviços e tecnologias é cada vez mais complexo e com mudanças/transformações mais aceleradas. Isto induz cada vez mais uma necessidade, quer nos clientes quer nas operadoras, de obtenção de informação agregada e compilada quer de negócio, quer operacional. Assim, torna-se possível tomar decisões sustentadas com uma maior celeridade, sendo por isso também uma mais-valia a sua integração nas atuais aplicações disponibilizada ao cliente.

1.1 Contexto

Atualmente no mercado, as redes fixas e móveis estão voltadas para ambientes IP[1] (que são mais baratos). Os negócios resumem-se a aquisição, instalação e manutenção, mas os custos de suporte são extremamente altos. A Altice Labs desenvolveu uma solução empresarial para comunicações convergentes, fornecendo um serviço "customer-oriented"com uma resposta flexível a uma maior abrangência de mercado, permitindo redução de custos. O sistema ABC é uma solução de comunicações, hospedado num ambiente baseado em *cloud*. Com base nas mais recentes técnicas de design, revela-se um portal personalizável que permite que provedores, empresas e assinantes configurem todas as caraterísticas disponíveis. Uma experiência do utilizador aprimorada, despesas operacionais reduzidas e um sistema baseado na web, o portal utiliza *self-provisioning*[2]. Com esta solução flexível e modular, os provedores do serviço serão capazes de implementar uma gama completa de serviços de comunicação empresarial fixo-móvel, minimizando os custos de comunicação dos utilizadores finais e melhorando a produtividade. Algumas das principais caraterísticas são:

• Restrição de chamadas efetuadas / recebidas;

[1] Internet Protocol

[2] também conhecido como *self-service* em *cloud*. É um prestador de serviço em *cloud* que permite aos utilizadores finais provisionar recursos por conta própria e configurar ou iniciar um serviço ou aplicação sem a intervenção de pessoal de TI dedicado ou dos próprios provedores de serviço.

- Números sempre disponíveis (para efeitos de escalamento);
- Encaminhamento de chamadas;
- Música de espera;
- Pré-atendimento;
- Fila de espera;

1.2 Problema na visualização das métricas

A plataforma ABC oferece uma forma de visualização dos dados no formato de listagem estática sem qualquer tipo de filtragem e contextualização. O utilizador tem dificuldades em perceber a listagem dos dados, porque nem sempre está familiarizado com todas as métricas, na forma como estas são apresentadas. Existe um número alargado de métricas, fora do contexto, que leva a erros de interpretação. A lista torna-se cansativa pelo seu tamanho, fazendo o cliente perder tempo na busca de informação. Por isso, é necessário que os dados sejam apresentados de acordo com as suas pretensões, em tempo real e de uma forma clara e apelativa, ajudando nas tomadas de decisão e ações de monitorização.

1.3 Novo componente de visualização

Para responder às pretensões referidas previamente, pretende-se oferecer um componente visual que mostre os dados de uma forma chamativa, percetível, em tempo real e dinâmica. Ir ao encontro da necessidade do utilizador e não obrigar o utilizador a "moldar-se"á aplicação. Tornar a plataforma mais *user-friendly* e personalizável, fornecendo dados em tempo real como suporte de monitorização e de sustentação a possíveis tomadas de decisão. Para tal, desenvolveu-se uma aplicação/protótipo, com uma representação em *dashboard* das diversas métricas de negócio e de gestão aplicacional em tempo real. Permitir um grau satisfatório de personalização e apresentar os dados de uma forma apelativa e útil estiveram na base da construção deste componente, com o objetivo de permitir ao utilizador encontrar os dados que pretende, interpretá-los e atingir as suas conclusões/decisões. Em suma, passar de dados meramente estáticos e listáveis para um componente dinâmico, personalizável e fácil de usar.

1.4 Metodologia

Para além do componente executar todas as funcionalidades pretendidas é importante que seja fácil de entender a informação que estamos a visualizar. Esta é a chave para que o componente seja efetivamente bom e ajude a acrescentar valor á plataforma. O cliente passou a ter ao seu dispor uma ferramenta que até ao momento não disponha, acrescentado assim valor e funcionalidades ao produto.

Para desenvolver a solução foi necessário:

- fazer um levantamento das soluções já desenvolvidas dentro da empresa que pudessem ser reutilizadas.
- fazer um estudo das tecnologias *front-end* que mais se adequam a este problema.
- explorar as frameworks existentes para a construção de *dashboard*s e de gráficos.
- perceber como é que a integração deve ser feita de forma que a chamada de dados demore o menos tempo possível.
- consultar o *back-end* para perceber o que já está feito e o que vai ser desenvolvido.
- contruir *mockup*s e o *product backlog* sob o formato de *user stories*.

O desenvolvimento foi feito utilizando a metodologia SCRUM. Para confirmar que o problema foi solucionado, esteve-se junto do utilizador para fazer uma avaliação do resultado, observando-se a utilização do componente e através do preenchimento de questionários, de forma a concluir em que medida o componente desenvolvido resolveu o problema.

1.5 Estrutura do documento

Este documento está dividido nos seguintes capítulos:

O presente capítulo, faz a contextualização, o primeiro levantamento do problema, como o resolver e como validar a solução;

No capitulo 2, é feito a exploração das áreas impactantes no desenvolvimento de um painel que controla as métricas de negócio. Visualização de dados, interação pessoa-computador e informação em tempo real. As várias formas de interação, de visualização e a importância de informação atualizada é explorada. Analisa-se a forma como estas áreas se relacionam e estão dependentes entre si para o bom funcionamento de um componente como um todo;

No capitulo 3 é explorado em maior detalhe o problema. O capítulo segue uma ordem cronológica. Começando pelo registo dos dados, como é que os dados são agrupados e quais são esses grupos, como se distribuem pelas bases de dados, como é que o cliente consulta os dados, como é que a consulta pode ser melhorada e como é que isso pode ser feito;

No capitulo 4, especifica-se em maior detalhe a solução. É aprofundado os aspetos a ter em conta para o desenvolvimento e como o desenvolvimento foi feito;

No capitulo 5, é explicada a validação. O planeamento, concessão e implementação dos testes de usabilidade e quais são os fatores de ameaça aos resultados obtidos bem como as conclusões que se podem tirar;

Finalmente no capitulo 6, apresentam-se as conclusões. Quais as contribuições e o trabalho futuro.

Capítulo 2

Visualização de informação

O mundo está a desenvolver-se rapidamente e a era da digitalização está a fazer com que a cada segundo existam grandes quantidades de dados a serem partilhados. O tráfico de dados aumenta a cada dia assim como seu volume.

As empresas estão a olhar para os dados que colecionam com mais atenção e atribuindo-lhes cada vez mais importância. As análises que fazem têm de ser desprovidas de erros e com grande precisão. Desta forma, dão respostas ao meio em que se inserem mais eficazmente e vão constantemente renovando-se de forma a prestarem serviços com qualidade.

A informação em tempo real assume um papel importante na coleta de dados e na sua análise. Quanto mais recente é a informação, melhor se entende as necessidades atuais e melhor se pode preparar o futuro de forma a responder a essas necessidades.

Mas, o quão importante são os dados em tempo real? Isto quer dizer que os dados históricos são descartáveis? De que forma se pode medir o valor da informação?

Neste capítulo serão abordados os tópicos de interação pessoa-computador, visualização de dados e a forma como se relacionam. Será explorado o ciclo de vida dos dados, quando é que são úteis e quando é que podem ser descartados ou guardados.

É também estudado a melhor fora de passar a informação dos dados ao utilizador de forma que este os perceba e tire todo o seu conteúdo de forma proveitosa.

2.1 Informação em tempo real e dados histórios

Existem constantes mudanças na informação. A cada segundo são gerados novos dados. É importante que a informação seja rapidamente disponibilizada para visualização. Um atraso de alguns segundos pode ter consequências negativas e com custos elevados associados. Quanto mais cedo a informação for tratada e disponibilizada mais segurança é oferecida aos clientes [9].

Informação em tempo real é a informação mais atualizada, isto significa que a informação em tempo real nem sempre é aquela obtida há segundos. Pode durar segundos, dias ou meses. O

contexto e a velocidade com que são gerados novos dados é que determinam o ciclo de vida em que os dados são uteis [22].

Por exemplo, há muitos anos atrás pensava-se que a forma da terra era plana. Durante um longo período de tempo, essa era a informação mais atualizada existente. Até que novos dados aparecerem e desvalorizaram por completo a informação anterior. Portanto, informação em tempo real pode ser informação com dias, meses ou mais.

Por outro lado, temos o mercado de ações em que o valor de cada ação muda constantemente. Os investidores precisam dos valores mais atualizados para considerarem investimentos. Neste caso o ciclo de vida da informação em tempo real é muito menor do que no caso anterior. Neste contexto os dados mudam constantemente e, apesar dos dados históricos serem importantes para fazer projeções e análises, são os dados mais atualizados que têm o maior valor [8].

Na área de negócios, para que as empresas possam ser bem-sucedidas precisam de ter a informação atualizada. Se é prestado um serviço que tem custos associados e que se espera que os clientes paguem para usufruir dele, qualquer incumprimento nos pagamentos deve ser reportado. Caso contrário, a empresa estará a perder dinheiro. O espaço entre o registo dos dados, o seu tratamento e a visualização deve ser o mais pequeno possível [8].

A informação deve ser transmitida o mais rapidamente ao utilizador para que este esteja ocorrente. Estes exemplos revelam que a informação atualizada nem sempre é aquela gerada há segundos e que o seu ciclo de vida vai depender do contexto.

Como foi visto o valor da informação está associado ao contexto. Os dados são úteis durante um determinado ciclo de vida até se tornarem dados históricos [4]. Consequentemente o seu valor também varia. Se fosse possível saber os números do Euromilhões antes do sorteio, estes teriam um valor enorme enquanto os números do sorteio de há um mês já não tem grande valor.

Concluindo, os dados em tempo real nem sempre são as informações geradas no presente, depende do contexto e da velocidade com que os novos dados são gerados. O fundamental é que, assim que surjam novas informações, estas devem ser transmitidas ao usuário para que ele esteja sempre bem informado.

A informação em tempo real é extremamente importante para as empresas. Mas, quando os dados em tempo real são transformados em dados históricos, são descartáveis? Já não são necessários?

Os dados passam a ser históricos a partir do momento em que aparecem novos dados para ocuparem a posição preenchida por estes até esse momento [18].

A nova informação pode eliminar o valor da antiga, como no caso de a terra ser plana. Neste caso, a informação anterior deixa de ter valor e passa a ser descartável. Contudo, dados históricos não significam que sejam descartáveis ou má informação [24]. As empresas investem cada vez mais em aparelhos capazes de guardar informação. Se á medida que o tempo passa os dados passarem a ser descartáveis, para que seria necessário fazer este tipo de investimentos? Antes de fazer os investimentos nos equipamentos, são realizados grandes estudos para perceber quais os dados que são importantes guardar e quais os dados que se podem "perder" [7]. Os dados históricos são normalmente muito importantes para ações de consulta [15]. Como quando se

quer consultar o valor pago em eletricidade no ano anterior, recorrendo às faturas dos 12 meses anteriores para fazer um balanço e possivelmente projeções.

No caso das redes sociais os dados históricos ajudam a rentabilizar o negócio [21]. Através dos dados históricos é possível traçar um perfil para cada utilizador. Sabendo os gostos do utilizador, a plataforma oferece conteúdo que este goste de ver. Vendendo espaços publicitários onde o utilizador tenha interesses em gastar dinheiro. Este tipo de empresas gasta grandes quantias em equipamentos para guardar os dados. São casos em que os dados históricos têm tanta ou mais relevância do que os dados em tempo real.

Percebe-se assim que os dados históricos têm a sua importância dependendo do contexto. É crucial fazer boas análises para se perceber aquilo que deve ser guardado e o que pode ser descartado de forma a otimizar os recursos e mantendo sempre bons desempenhos de performance. Mesmo guardando grandes quantidades de dados, as redes sociais mantêm uma velocidade de utilização excelente.

2.2 Visualização dos dados

A forma como os dados são apresentados aos utilizadores é fundamental para que este os possa perceber. O sucesso de boas análises de dados e projeções está dependente não só da qualidade dos dados, mas também da forma como eles são visualizados e acedíveis pelo utilizador. A seguir são explorados os fatores que devem ser tidos em conta para que os dados sejam bem transmitidos.

2.2.1 Impacto da visualização de dados na area dos negócios

A área de negócio está focada principalmente no dinheiro. As empresas sobrevivem com o que ganham ao vender os seus produtos. São necessários grandes estudos de mercado para oferecer algo útil e com valor por um determinado período de tempo. Para este ambiente, a análise de dados, a qualidade dos dados, a disponibilidade de dados, etc. são levados com muita seriedade e como vitais para o bom desempenho da empresa. A visualização dos dados e a forma como são exibidos é importante para fazer a análise [23].

Os dados por si só não significam nada. É necessário dar um tratamento para que ganhem valor. É necessário trazer os dados para a "linguagem comum". Os dados sem tratamento muitas vezes não são reconhecíveis pelos utilizadores [11]. Assim, para que os utilizadores compreendam o seu verdadeiro significado, é necessário que ambos falem a mesma língua. Isto é feito mostrando os dados da forma correta. Quando um utilizador vê os dados, o objetivo é que ele os compreenda e obtenha o maior valor para fazer a melhor análise possível [11]. A ideia a ter em mente é que os dados só terão importância e impacto se o recetor os visualizar de forma adequada e percetível. O efeito deste facto é como uma bola de neve no cume de uma montanha. Se o analista visualizar os dados de forma adequada, obterá o máximo de valor possível, o que permitirá fazer uma análise mais precisa que permite à empresa ter a melhor perceção do pretendido. O oposto terá efeitos devastadores.

2.2.2 Analise nos bastidores

Quando mostramos dados a um utilizador, não é apenas pegar nos dados recolhidos e exibi-los. Há todo um processo por trás. Por vezes temos dados que não são recolhidos diretamente, mas sim gerados (ex: duração média das chamadas). Todo o tratamento feito antes de mostrar os dados visa colocar os dados percetíveis para o utilizador.

Antes de o utilizador ver os dados, é necessário perceber se os dados recolhidos são percebidos pelo utilizador [13]. Em alguns cenários, os dados recolhidos são muito técnicos e específicos. Se não forem tratados, provavelmente não valem nada porque o utilizador não os compreenderá. Por exemplo, quando os dados são recolhidos numa medida europeia como o Km por hora e os clientes são americanos e querem vê-lo em milhas. É necessária a conversão antes de mostrar os dados. Este é um exemplo de um caso simples, mas o mesmo acontece para dados mais complexos.

Grande parte do valor que os dados podem ter depende da compreensão dos utilizadores [27]. Podemos recolher todos os dados possíveis, mas se o utilizador mais tarde não os compreender e não os utilizar, são inúteis [14]. Entregar uma pilha de dados recolhidos geralmente não é uma boa abordagem. É preciso refinar os dados. Coloca-los à disposição para que sejam percebidos e vejam o seu valor reconhecido. Basicamente, que "falam a mesma língua"[17]. Os utilizadores não necessitam ver todos os dados. Apenas irá tornar tudo mais confuso. Obviamente devem estar disponíveis porque podem precisar deles por qualquer motivo. Mas, tornar as coisas simples e organizadas normalmente é a melhor abordagem. Por exemplo: a categorização de dados ou a colocação de terminações em dados numéricos. Ajuda a entender os dados. Se for lida a frase "Comprei 9", não se sabe o que foi comprado. Mas com um simples final "Comprei 9 laranjas"já se sabe o que foi comprado. A informação faz sentido, percebemos e tem valor. Os detalhes fazem a diferença. Numa área extremamente competitiva tudo conta, estas pequenas diferenças podem fazer a diferença [26]. Uma boa análise acrescenta valor às ferramentas. Uma relação vantajosa para todas as partes interessadas.

2.2.3 Métricas diferentes significam diferentes tipos de visualização

As recolhas de dados podem conter vários tipos de dados. O tipo de dados varia, e mesmo dentro do mesmo tipo, temos que tratar cada dado de uma forma única. Cada métrica deve ter um tratamento isolado para obter todo o seu valor e apresentá-lo ao usuário da melhor forma [25]. Este é um passo particularmente importante e delicado porque requer muita atenção, criatividade e conhecimento do público-alvo.

Tratar cada métrica isoladamente demora muito tempo porque é uma tarefa muito específica. Claro que quanto mais dados são recolhidos, mais tempo este passo pode demorar [25]. É importante ter a ideia desde o início de que mesmo tendo dados do mesmo tipo (ex: numéricos) e que aparentemente podem estar dentro dos mesmos padrões, deve-se olhar para eles isoladamente. A ideia de que "ok estes dados são do mesmo tipo, vamos passar para outra etapa"ou "estes são praticamente os mesmos, por isso já está analisado"é errada. Claro que isto pode acabar por acontecer, mas é necessário chegar a esta conclusão depois de tratado cada dado isoladamente e não

da presunção. Cada métrica, num dado contexto, terá um tratamento especial que conduz a uma visualização única [30]. É aqui que os dados podem ou não ganhar valor. Este pequeno passo pode definir o sucesso ou o fracasso de todo um desenvolvimento. Pensando num dado de lançar. Seis resultados possíveis. Uma boa abordagem para mostrar os resultados do lançamento pode ser um gráfico de barras. Como se exemplifica na figura 2.1

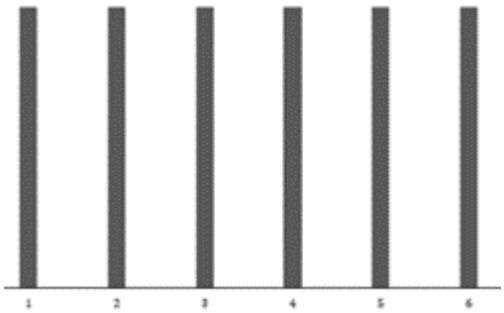


Figura 2.1: Possível amostragem dos resultado dos lançamentos de um dado

O eixo X é o número da face e o eixo Y é o número de vezes que o lançamento teve o número da face correspondente. Fácil de entender para qualquer utilizador e representado da forma mais simples possível.

Agora imagine-se outro cenário. Em vez de ter um dado com seis resultados possíveis, existem centenas de resultados possíveis. Utilizando a mesma abordagem (gráfico de barras) o resultado será algo semelhante ao da figura 2.2

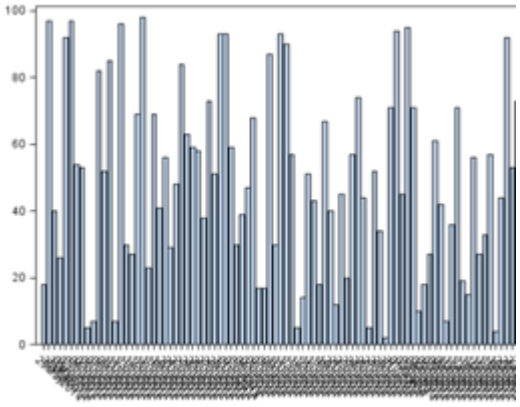


Figura 2.2: Possível amostragem de dados com centenas de variáveis

Analisando o gráfico, é difícil perceber os dados. Há muitas variáveis no eixo X e, consequentemente, muitas barras, tornando-o praticamente impossível de ler. A legenda não pode ser reajustada porque há muitas entradas para um espaço curto. Não se consegue descobrir a que número pertence cada coluna.

Apesar de termos o mesmo tipo de dados (nos exemplos, numéricos), não podemos tratá-los da mesma forma. No segundo caso, como temos vários resultados possíveis, a melhor abordagem é categorizar os dados. Desta forma será possível ter todos os dados organizados e legíveis. Como

temos várias entradas, a tabela pode ficar bastante grande, por isso podemos quebrá-la ou torná-la personalizável.

Estes exemplos são para números. Mas os tratamentos servem todos os tipos de variáveis. Há casos em que os booleanos têm de ser convertidos em colunas (ex: 0 - significa "não", 1 - significa "sim") para serem compreendidos pelo utilizador. Se o utilizador estiver à espera de ver um "sim"ou "não"como resposta e lhe for dado zeros e uns, ele ficará confuso e frustrado. O tipo de cenário que se quer evitar.

Cada dado tem características específicas. Depende muito da capacidade e criatividade dos programadores e da sua capacidade de ver os cenários do ponto de vista do utilizador final. O feedback do utilizador é importante. O que é percetível para os programadores pode nem sempre ser compreendido pelos utilizadores finais.

2.2.4 Importância da personalização

O que o cliente quer nem sempre é claro, quer para as equipas de desenvolvimento, quer para ele. Os clientes podem pedir algo bem planeado e desenhado, mas, quando veem o resultado, têm a sensação de que, afinal, não é bem o que querem. Para os componentes de visualização, é comum o cliente dizer “preferia isto mais à direita"ou "Se fosse maior, seria melhor". O melhor é oferecer personalização. Desta forma, o utilizador, tem a plataforma de acordo com os seus desejos [29].

Para componentes gráficos e de monitorização, não é fácil para os programadores ver o que o cliente quer. A ordem pela qual querem ver os dados, o tamanho dos gráficos, os que devem ser destacados, as cores, etc. é difícil de colocar na forma que o cliente realmente quer. A solução passa por permitir ao cliente selecionar o que quer. Funções como colocar um gráfico mais baixo, reajustar o tamanho, realçar um gráfico específico são o tipo de personalizações amigáveis [2]. A flexibilidade da plataforma permite-lhe também responder bem às mudanças no ambiente por um período de tempo mais longo. Por exemplo: imagine-se que estamos a controlar o número de sementes produzidas por certas espécies de flores. Num determinado momento, uma nova espécie aparece. Se a plataforma permitir a introdução de uma nova espécie para monitorizá-la, isso significa que se adaptou bem às mudanças no ambiente, evitando os custos de manutenção e poupando dinheiro. Imagine-se também que existe um certo painel. Este painel controla a produção de camisolas. Durante o decorrer do mês estamos interessados em ver quantas camisolas já foram produzidas, então no cimo do painel, pretende-se ter um contador com o número de camisolas produzidas. No final do mês, pretende-se um relatório com quantas peças foram produzidas por cada funcionário. Então, já não é preciso o contador de camisolas produzidas. Elimina-se o contador e introduz-se um novo gráfico com estes parâmetros. Este tipo de dinâmica e flexibilidade é muito bem recebida pelos utilizadores finais.

Um utilizador pode estar a olhar para um gráfico de barras, mas depois percebe que prefere ver os dados num gráfico circular. Permitir que estas alterações sejam feitas faz com que os clientes fiquem satisfeitos. Nunca esquecendo que o objetivo é que o cliente compreenda os dados. Assim, a personalização também ajuda a compreender os dados. Também é útil para um acesso mais rápido a informações específicas através de campos de pesquisa. Quando se tem várias entradas e

se quer ver uma em específico, ter a possibilidade de procurá-la em vez de perder tempo à procura dela aumenta a rentabilidade [19].

A personalização permite ao utilizador tirar o melhor partido da plataforma porque a plataforma está de acordo com os seus desejos. Qualquer ajuste que considerar necessário pode ser feito por ele, entregando um produto que modelará os desejos do cliente e que dará uma resposta eficaz durante um período de tempo mais longo.

2.3 Interação Pessoa-Computador

Interação Pessoa-Computador é uma área com muitos subcampos, multidisciplinar com uma relação intrínseca com a ciência da computação. Estuda a relação entre humanos e computadores em ambos os sentidos: Humano -> Computador e Computador -> Humano. É uma disciplina focada no projeto, implementação e avaliação dos sistemas interativos de uso humano [5]. Estuda como é que a comunicação pode ser feita, a natureza da interação, o contexto de utilização e como pode ajudar a melhorar os sistemas. Ajuda a aumentar a influência de equipamentos disponíveis (hardware e software) nos contextos humanos, estabelecendo regras para construir uma boa interface. Uma boa interface é crucial, é a forma como um utilizador contacta um sistema. O contacto pode ser motor ou percetivo. É o canal de comunicação por isso, necessita de estar perfeitamente sincronizado para que o ser humano possa ser satisfeito [7].

2.3.1 Influência do contexto

Cada ser humano é influenciado pelo contexto em que é inserido. Adaptam-se os comportamentos aos contextos em que cada um se insere. Estar numa aula é diferente de estar com amigos. O software é desenvolvido para um público específico. O público é inserido num contexto. Alguns *softwares* assumem que os utilizadores já têm algum conhecimento prévio para agir num contexto específico outros *softwares* são desenvolvidos para ensinar crianças.

O *software* destina-se a ajudar com uma determinada tarefa num contexto específico. Conhecer o contexto em que o *software* será inserido é crucial para adaptar algumas funções de acordo com o público-alvo [6]. Desenvolver uma plataforma de programação para crianças no primeiro ciclo é diferente de desenvolver uma plataforma para estudantes universitários. Aspetos como o conhecimento previamente adquirido, as metodologias de trabalho, as horas em que podem concentrar-se numa tarefa, são aspetos que influenciaram a fase de desenvolvimento e o produto final. Provavelmente para crianças a plataforma inclui mais cores e animações e para estudantes universitários será mais formal. A aceitação pelo público-alvo é o que define o sucesso de um sistema. Para isso, é necessário que os programadores se coloquem na posição de utilizador final. Em casos específicos, pode não ser fácil estar nesta posição, como por exemplo, desenvolvendo ferramentas para pessoas com dificuldades motoras. Para este caso, o aconselhável é testar a plataforma com utilizadores reais para recolher *feedback*.

Contextos diferentes requerem métodos diferentes. Compreender o público-alvo e o contexto onde a plataforma será inserida para que seja bem aceite. As interações entre humanos-computadores são influenciadas pelo contexto. Conhecimentos anteriores, idades, dificuldades motoras ou outras dificuldades são exemplos de aspetos a ter em conta para compreender o público-alvo [1]. Perceber o contexto leva à compreensão das necessidades reais dos utilizadores, o que leva a dar uma resposta mais eficaz.

2.3.2 Formas de interação

Há diferentes formas de interagir com o sistema. "Os computadores e dispositivos relacionados, têm de ser concebidos com a compreensão de que as pessoas com tarefas específicas em mente vão querer usá-los de uma forma que seja perfeita em relação ao seu trabalho diário"[1]. Um utilizador comum utiliza o computador sem se aperceber de todas as formas possíveis que um computador oferece para comunicar. Isto tem a ver com as necessidades de cada utilizador.

Ferramentas como a lupa ou os leitores de ecrã são ferramentas e formas de interação mais apropriadas para pessoas com deficiência. A maioria dos utilizadores comuns nem sequer sabe da existência destas ferramentas. Utilizando apenas o teclado e o ecrã também é possível ditar as ações desejadas para o computador e este segui-las [20]. Pode-se pensar que se trata de uma ação futurista, mas é atualmente utilizada por pessoas com necessidades especiais. Provavelmente nos próximos anos a interação passará por mais comandos de voz, mas neste momento os utilizadores comuns preferem os métodos mais tradicionais. Para computadores que incluem uma câmara é possível interagir com o sistema através dos olhos. A sua utilização não é muito comum, mas tem uma operação muito simples. Basicamente, os olhos do utilizador funcionam como um rato. Onde o utilizador olha, o rato aponta para lá, através de ações como apontar dois segundos para uma pasta e abri-la, é possível realiza todas as ações normais através dos olhos. O exemplo de Stephen Hawking é o melhor exemplo para entender como isto funciona. No início da sua doença, ele comunicava através de uma tela que emitia sons (palavras) onde quer que olhasse. Estes são os métodos que estão presentes num computador normal [10]. É claro que é possível ligar capacetes neuronais que detetam a atividade cerebral e leem os sinais recebidos, mas ainda carecem de maior exploração. Este método será muito útil no futuro e será capaz de alcançar e incluir mais pessoas na utilização das máquinas.

É possível utilizar uma combinação dos métodos listados para ter uma experiência mais personalizada e ligar outros métodos externos. Aqui está um registo das principais funcionalidades oferecidas pelos computadores para incluir todo o tipo de pessoas. Cada experiência de utilizador baseia-se nas características de cada um.

2.3.3 Ações implícitas

As tecnologias estão presentes na vida quotidiana de uma forma tão enraizada e crucial que se acaba por ter ações implícitas. A tecnologia já controla aspetos dos quais nem se dá conta, desde

ações básicas como parar no semáforo vermelho, até ações como controlar o acesso de pessoas autorizadas a certas áreas de um edifício.

É cérebro que comanda o corpo. Quando se aprende algo novo, o cérebro pode ter dificuldades em assimilar novas informações. Mas quando repetimos uma ação constantemente, a atividade começa a ser desenvolvida sem "pensar". Os cérebros processam informação e ações de forma tão automática que, se procede sem pensar no que se está realmente a fazer. Cada ser humano tem ações implícitas [16]. Desenvolver um sistema que engloba toda essa informação é impossível, mas como anteriormente referido algumas ações são tão básicas no dia-a-dia que são consideradas "ações normais". É importante perceber que em componentes de software o cérebro funciona exatamente da mesma forma. Por exemplo, quando se está a usar uma aplicação e se quer apagar algo normalmente procura-se ícones a vermelho. Associa-se a cor vermelha às ações de exclusão. É importante ter isso em conta para desenvolver uma boa interface. Estudar os padrões de comportamento do utilizador ajuda a desenvolver uma interface limpa. Interfaces limpas são boas para não assustar os utilizadores [12]. Desta forma, acham mais simples e fácil usar uma ferramenta e aumentar os índices de desempenho. Em alguns casos, é útil reduzir a fase de aprendizagem de um componente. As interfaces atuais são baseadas em símbolos que desencadeiam ações. O conhecimento do significado dos símbolos está implícito. Quando se vê o simbolo da figura 2.3 imediatamente associa-se a um botão de menu.



Figura 2.3: Ícone associado á abertura de menus

Uma boa interface não é sinónimo de uma boa plataforma para uma determinada tarefa, porém, se o efeito quando olhamos para um componente não é tão assustador como olhar para o painel de controlo de um avião, já se parte com alguma vantagem. A ação em si e o desempenho de um sistema é o mais importante. Apesar de um avião ter um painel de controlo assustador, o importante é permanecer no ar. As interfaces têm um grande impacto no desempenho e ajudam o utilizador a executar uma tarefa.

2.3.4 Formas de medir uma boa interação

Todos os tópicos previamente discutidos têm um enorme impacto no sucesso de um sistema. Tendo tudo isto em conta, está-se mais perto de uma construção bem-sucedida. No entanto, depois da construção, como se pode ter a certeza de que foi alcançado o objetivo e o público-alvo está satisfeito?

Depois de todo o trabalho de desenvolvimento, chega-se ao ponto em que é preciso validar o componente, através da aceitação do público-alvo ou alcançando os objetivos propostos. Este último passo é onde o trabalho realmente ganha valor. Qual é o valor de um componente que quando é posto em prática não cumpre o que deve? Provar que os componentes têm uma boa interação homem-computador é a chave de ouro depois da fase de desenvolvimento.

Existem muitos métodos para fazer este tipo medições. Desde a utilização dos dispositivos mais básicos, como o rato e o teclado até ao rastreio dos olhos, resposta galvânica da pele, volume sanguíneo e medições da frequência cardíaca etc. Para utilizar qualquer um dos equipamentos aqui mencionados, é necessário algum treino e conhecimento que vão além do HCI. "Ferramentas de rastreio de olhos que nos dizem para onde as pessoas estão a olhar num ecrã podem ajudar-nos a compreender os processos visuais envolvidos nas listas de opções de navegação. A resposta cutânea ou os monitores cardiovasculares podem fornecer informações sobre o nível de excitação ou frustração do utilizador. A informação rica e detalhada sobre as atividades e respostas dos utilizadores fornecidas por estas ferramentas pode ajudar a alargar a nossa compreensão do uso humano das interfaces de computador."[3]. Para validar o componente, depende do que se quer analisar. Para componentes visuais, o rastreio de olhos é uma ótima ferramenta, ajuda a compreender quais os componentes que se destacam e o impacto visual das cores, por exemplo. Por exemplo, é possível fazer testes, dizendo ao utilizador para fazer uma determinada ação e ver se ele necessitou de olhar para muitos componentes ou se foi fácil para ele encontrar o que queria.

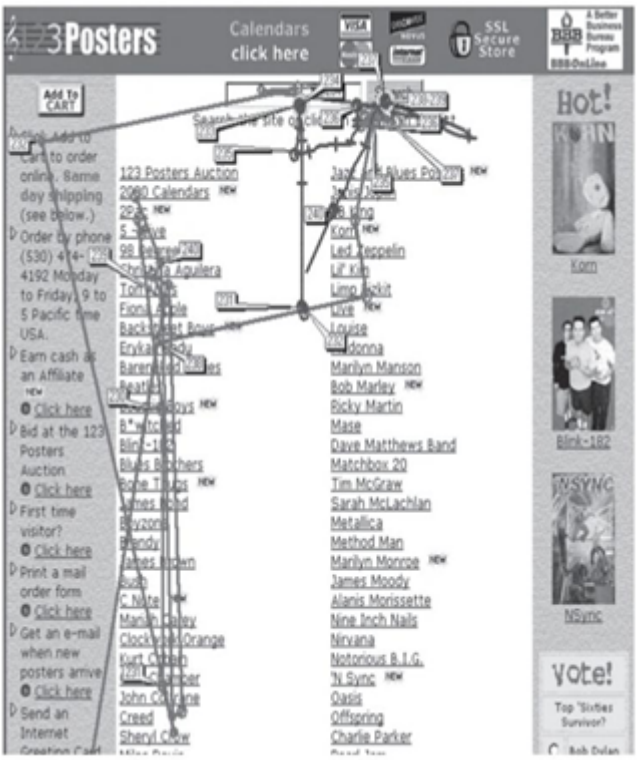


Figura 2.4: Exemplo de um *Eye Tracking*
Imagem retirada de [5]

No caso dos componentes visuais, como os painéis de controlo, é importante perceber se a informação está bem organizada e se as ações como adicionar um novo gráfico aos painéis é fácil de operar. No primeiro caso, uma experiência possível é pedir ao utilizador que forneça determinadas informações presentes nos painéis e analisar o tempo de resposta. No segundo caso, analisar através do rastreio dos olhos se o utilizador teve dificuldade em encontrar o botão para

essa ação, é um bom método de análise. Se o utilizador olhar para toda a página antes de encontrar o botão para adicionar um gráfico, provavelmente o melhor é reajustar a localização e o tamanho do botão.

Todas estas ferramentas querem enriquecer o valor do componente através da validação. O feedback é essencial. Só assim, se pode compreender se o que foi desenvolvido é bem utilizado e se concretiza as ações ao qual foi proposto.

2.3.5 Evolução e perspetiva futura

É uma área fascinante neste aspeto. Ao longo dos tempos muitos estudos foram produzidos, a tecnologia evoluiu e novos estudos surgiram. As tecnologias existentes há dez anos são diferentes das que temos agora. São necessários novos estudos para acompanhar novos desenvolvimentos para tirar o melhor partido da tecnologia e compreender porque interagimos num determinado modo em cada espaço de tempo [5]. Isto reflete a importância desta área nos componentes de software. Os estudos nesta área nunca terminarão até que o desenvolvimento das tecnologias também termine.

Historicamente, humanos têm interagido com máquinas desde o início. Foi, por exemplo, a partir das grandes guerras que se registaram grandes desenvolvimentos nas áreas tecnológicas. A necessidade de desenvolvimento conduziu a grandes feitos. Dos rádios básicos a telecomunicações de longa distância [1]. Interagimos sempre com máquinas como o cientista informático Jonathan Grudin escreveu no *Annals* em 2005: "As pessoas têm interagido com computadores desde início, mas demorou bastante tempo para que a interação homem-computador se tornasse num campo de investigação reconhecido". O estudo formal desta área aponta para a data de 1980. Impulsionado pela chegada do PC IBM [12].

Nos últimos anos temos assistido ao crescimento de dispositivos inteligentes. A comunicação entre pessoa e dispositivo passou da escrita para simples comandos de voz. A perspetiva para o futuro é que todos os dispositivos estejam conectados e comuniquem sem que o humano se aperceba. Os comandos de voz tornar-se-ão mais complexos e ações como a pesquisa escrevendo num motor de busca desaparecerá dando espaço aos comandos de voz. Este tipo de interação já começou a ser introduzido. Por exemplo, para saber o tempo é possível perguntar diretamente ao *smartphone* em vez de escrever "meteorologia"nos motores de busca WWW. Os seres humanos serão controlados pelos dispositivos sem se aperceberem (no bom sentido). Por exemplo: relógios inteligentes podem medir o estado espiritual, por isso quando uma pessoa se sentir ansiosa, numa casa inteligente as luzes podem automaticamente ajustar-se a tons mais acolhedores, a temperatura pode subir para aumentar o conforto e uma coluna de som pode começar a tocar música relaxante. A tecnologia cuidará dos humanos sem que eles deem conta. É inevitável.

Com o avanço da tecnologia surgem novos problemas. Como mencionado, os comandos de voz tornar-se-ão mais complexos em vez de apenas darem instruções simples. Este é um dos problemas a resolver nos próximos anos. Como é que os dispositivos obedecem a instruções mais extensas e complexas? Este será um tema que terá grandes desenvolvimentos nos próximos anos. Outro problema que surgiu no ano passado, especialmente devido à situação da pandemia, é o

reconhecimento facial. Quando um sujeito usa óculos de sol ou máscaras, os dispositivos têm dificuldade em reconhecer o utilizador. É previsível que nos próximos anos os dispositivos sejam capazes de reconhecer pessoas em qualquer cenário e com isso, outras aplicações podem aparecer. Estes são alguns dos problemas existentes, mas certamente outros irão aparecer com os próximos desenvolvimentos. Basta olhar para o caso dos carros. Há anos, a interação entre humanos e carros era mecânica, agora já existem carros inteligentes, criando novas formas de interação e novas tecnologias como o *autopilot*.

As tecnologias serão tão inteligentes e úteis que será inevitável não as deixar entrar. Permitir-nos-á manter-nos focados nos nossos empregos e passar mais tempo com amigos e familiares. O futuro da tecnologia passará pelo aumento do conforto e poupar tempo em tarefas. Dispositivos e casas inteligentes tornar-se-ão cada vez mais inteligentes e permitirão que o ser humano faça o que quer, sem que ele perceba que a tecnologia está a fazer o seu trabalho, como aspirar o chão. Se tanta tecnologia na nossa vida será boa? Só o tempo pode dizer.

2.4 Conclusão

Depois de analisados os tópicos anteriores, conclui-se que estes têm uma enorme importância no desenvolvimento de *software*. O desenvolvimento bem-sucedido destas três áreas garante o bom funcionamento de qualquer plataforma. São áreas que requerem especial atenção devido ao impacto que têm no desempenho do *software*.

Considera-se estas três áreas cruciais para desenvolver painéis de controlo na área de negócios. Esta é a principal razão pela qual se aprofundou estes temas, para construir uma ferramenta que permita ao cliente monitorizar, gerir e apoiar as decisões. O objetivo é oferecer uma ferramenta que realmente se adeque aos desejos do cliente, cumprindo com tudo aquilo que é explorado neste capítulo.

Depois de desenvolver cada tópico separadamente, o bom funcionamento de um componente depende do bom funcionamento dos três tópicos em conjunto. O não funcionamento de um dos tópicos tem consequências para os outros. A sincronização entre os três determina o sucesso da plataforma. Estes tópicos isolados podem não ter valor ou ver o seu valor reduzido porque os outros têm defeitos. A vitalidade de cada um é importante, mas é a operação conjunta que determina o funcionamento global da plataforma. A figura 2.5 resume como os diferentes tópicos se relacionam.

Conclui-se que, só é possível ter uma plataforma em pleno funcionamento se estes três tópicos estiverem sincronizados. A operação isolada ou a não operação de um deles tem consequências para o funcionamento global. Só o bom funcionamento dos três permite uma plataforma otimizada.

Funcional \ Não Funcional	HCI	Visualização de dados	Tempo-real
HCI	X	Consegue utilizar o equipamento mas não consegue visualizar os dados corretamente, o utilizador não conseguirá absorver as informações da plataforma e tarefas como gestão ou tomada de decisão são afetadas	O utilizador consegue usar o equipamento, mas não consegue aceder ás informações em tempo real
Visualização de dados	Caso o utilizador não consiga aceder ao sistema ou plataforma, por melhor que seja a visualização dos dados, não consegue visualizar as informações	X	O utilizador consulta e entende bem os dados, mas os dados que está a visualizar estão incorretos. As operações de gestão e tomada de decisão são afetadas. Apenas os dados históricos estão disponíveis para consulta
Tempo-real	As informações são enviadas e ficam disponíveis em tempo real, mas o utilizador não consegue usar o equipamento. Desta forma, não pode absorvera informação em tempo real	As informações disponíveis estão corretas, mas a forma como são exibidas é enganosa. O utilizador interpreta mal os dados.	X

Figura 2.5: Relação entre tempo-real,visualização de dados e HCI

Capítulo 3

Visualização de informação na plataforma ABC

A plataforma ABC apresentava o problema já anunciado na visualização da informação. Esta informação tem um valor alto para os seus clientes. A gestão do negócio, a tomada de decisão, projeções futuras etc. são ações cruciais para que uma empresa possa consolidar-se no presente e crescer no futuro.

Quando se oferece um produto da natureza do ABC aos clientes, estes querem ter uma visualização da informação personalizada, intuitiva, apelativa e clara. Quando se tem apenas listagem de dados estática e sem opções de filtragem isto não acontece. Imagine-se consultar o histórico de chamadas do telemóvel, e para ver as chamadas do dia de hoje ter que percorrer a listagem de chamadas desde o dia em que foi comprado. Ou então para consultar um número ter que procurar na lista de contactos em vez de procurar pelo nome diretamente.

Para não gerar confusão nos próximos pontos, dever-se-á ficar com esta ideia: um utilizador tem vários clientes. Ou seja, imagine que está a utilizar a plataforma ABC. O seu negócio tem vários clientes. Então você é o utilizador e os consumidores do seu negócio são os clientes.

3.1 Registo de dados

Todas as chamadas e SMS recebidos pelos clientes são registadas automaticamente pelo sistema e organizados em grupos. Estes grupos são: Sms and Call (sec. 3.2.1), Hunt Group (sec.3.2.2) e Waiting Queue (sec.3.2.3)

Cada chamada gera um conjunto de dados. Após o término da chamada, são registados os dados diretos (ex: duração da chamada) e atualizados os dados indiretos (ex: tempo médio das chamadas, total das chamadas etc.). O registo dos dados também pode ser feito parcialmente. Quando as chamadas passam dos cinco minutos, os registos passam a ser de 30 em 30 segundos até ao término da chamada. Quando as chamadas passam dos 10 minutos os registos passam a

ser feitos de 10 em 10 segundos. Por exemplo, quando uma chamada chega aos 5:30 minutos de duração é feito um registo parcial. Que entra nas BDs. Após terminar a chamada, os dados dessa chamada são atualizados nas BD.

As mensagens também são registadas. Nesta componente apenas é feito o registo do total de mensagens recebidas e não o do seu conteúdo.

3.2 Grupos de dados

Após o registo dos dados estes são agrupados em diferentes grupos. Os grupos podem ter métricas iguais. No entanto, o facto de pertencerem de grupos diferentes representa um contexto diferente.

3.2.1 SMS and Calls

Aqui entram as estatísticas relativas a chamadas de voz e SMS's. Pode agregar estatísticas dos outros grupos desde que sejam chamadas de voz ou SMS's.

O objetivo deste grupo é que seja possível ter uma visão global em que sejam incluídas todas as chamadas de voz e SMS. As chamadas de voz representam cerca de 80 a 95 por cento do meio de comunicação. Posto isto, foi necessário criar um grupo específico para que fosse possível visualizar as suas estatísticas. As SMS apenas têm um contador associado que contabiliza o total de SMS recebidas.Relativamente ás chamadas, este grupo permite entrar em maior detalhe sobre as mesmas.

3.2.2 Hunt Group

Este serviço é responsável pelas chamadas produzidas pelo grupo de atendimento. Por grupo de atendimento entendam-se como as pessoas que atendem o telefone quando precisamos de assistência. Como por exemplo, no caso de acidente, a assistência de viagem. Este serviço tem como responsabilidade distribuir as chamadas pelos múltiplos destinos tendo por base as diferentes funções.

Certamente já passou pela experiência de ligar para uma assistência e do outro lado receber a resposta: "(...) prima a tecla 1, (...) prima a tecla 2". Este serviço é responsável por essas funções e correspondente reencaminhamento.

As estatísticas deste grupo têm como foco saber se as chamadas foram recebidas e atendidas, quanto tempo em média demorou o atendimento etc.

3.2.3 Waiting Queue

Este serviço é responsável pelas chamadas que estão em fila de espera até serem atendidas. Permite á plataforma ter chamadas em espera enquanto outras são atendidas.

Enquanto esperam pelo atendimento é produzida uma música, e em espaços de tempo definidos e de igual intervalo é imitido a posição na fila de espera. Este serviço permite definir o

tempo máximo de espera na fila, até que seja desligado ou transformada. Esta transformação significa que assim que possível a chamada é retornada. Se a fila atingir um tamanho máximo (que é predefinido), é configurado se a partir desse ponto as chamadas são rejeitadas ou transformadas.

As estatísticas dedicam-se a saber quais as chamadas que são ou não atendidas em face das totais recebidas (que entram neste grupo), dentro das não atendidas a causa de não atendimento e as médias de tempo em espera e da duração da chamada (incluindo a espera e a conversação). Na seguinte tabela pode ver em maior detalhe as métricas e o seu significado.

3.3 Organização dos dados nas Base de Dados

Depois do registo dos dados, estes ficam armazenados em Base de Dados (BD). Esta secção visa perceber como é que são organizadas as várias bases de dados.

A primeira ideia que devemos excluir é que cada grupo tem uma BD específica. Isto não acontece. Aquilo que acontece é que os dados são etiquetados. Ou seja, os dados levam uma etiqueta que especificam a que grupo pertencem. Por exemplo o grupo Waiting Queue tem um identificador "wq".

Os dados não são distribuídos pelas BD's conforme o grupo, mas sim pelo seu tempo. Quer isto dizer que as BD são organizadas no seguinte formato:

- BD-15-minutos
- BD-30-minutos
- BD-1-hora
- BD-12-horas
- BD-1-dia
- BD-7-dias
- BD-15-dias
- BD-30-dias
- BD-historical-data

Conforme o registo temporal dos dados, são registados nas BD's. Conforme o seu registo vai avançando os dados são transferidos para a BD correspondente. Por exemplo, os dados que são registados neste momento estão na BD de 15 minutos. Amanhã estarão na BD de 1 dia.

3.4 Problemas na visualização dos dados

Na plataforma ABC a amostragem de dados é uma simples lista. Sempre que o utilizador quiser consultar dados dos clientes, tem que ir á lista e procurar o nome do cliente e os dados

que quer consultar. Para casos em que o utilizador tivesse um ou dois clientes esta abordagem até poderia ser aceitável, mas, para os casos em que os utilizadores tenham dezenas, centenas e milhares de clientes é impensável fazer buscas numa lista que não permite nenhum tipo de filtragem.

Para além do problema da listagem em si, existe o problema da descontextualização. Ou seja, a lista não tem qualquer tipo de ordenação. Nem por grupo, nem por cliente, nem por tipo de dados. Os dados que se pretendem encontrar podem estar tanto no início como no fim da lista, inclusive entre dados de outro cliente e de outros grupos. Basicamente, a sua ordenação é aleatória.

É, portanto, custoso consultar os dados, por conseguinte, pretende-se que os dados estejam facilmente acessíveis, para otimizar o processo de negócio do cliente. Todas as tomadas de decisão, ações de gestão, projeções etc. passam pela consulta de dados.

Estes não têm só que estar disponíveis de uma forma mais user friendly. Têm também que estar mais organizados. Oferecer filtragem aos dados e contextualizá-los. Desta forma o utilizador mais facilmente entende os dados e os manipula.

3.5 Visão de solução

Na área dos negócios o termo *dashboard*s é sinónimo de "painel de controlo": Estabelecer a conexão entre o cliente e tudo aquilo que está a acontecer apresentando os dados em formato de gráficos. Não adianta ficar a analisar milhares de linhas de listas e perder bastante tempo para ter uma visão geral do cenário.

O ideal é usar um *dashboard*. As informações atualizadas ficam consolidadas num único painel, otimizando o tempo e obtendo o resumo do desempenho das áreas do negócio. O principal objetivo é facilitar a visualização de dados, identificar problemas, analisar situações e acompanhar processos das empresas de forma eficiente, possibilitando uma visão clara dos dados e transparência corporativa na gestão da empresa. Um *dashboard* é uma ferramenta de gestão da informação que ajuda no acompanhamento e exibição de indicadores. São customizáveis para atender às demandas específicas de cada empresa e às suas áreas. Todas as informações que parecem confusas, se olhadas isoladamente, são entregues em gráficos, tabelas e imagens de forma simples e que possibilita a tomada de decisão bem como uma melhor tarefa de gestão.

Para por a solução em prática tal procedeu-se á realização dos seguintes pontos:

- Para o desenvolvimento do novo componente foi feito um levantamento das tecnologias existentes que melhor se adequam a este problema na componente de *front-end*.
- Foi feita uma análise de componentes que já foram desenvolvidos que podem ser reutilizados. Explorou-se quais as fremeworks que podem ajudar na construção dos *dashboard*s e na produção dos gráficos.
- Estudou-se a parte do *back-end* de forma a entender como se pode interligar com o *front-end*. De forma a perceber o que já está feito, o que está planeado desenvolver e o que falta desenvolver para que o novo componente funcione.

- Elaborou-se as linhas orientadoras para a definição do grau de customização do componente e as regras que este deve seguir para que seja *user-friendly*.
- A realização das etapas anteriores culminaram num *product backlog* composto por *user stories* e por *mockups*.
- O desenvolvimento foi feito seguindo uma metodologia SCRUM.

3.6 Validação

Depois da fase de desenvolvimento do novo componente foi necessário perceber se efetivamente o problema inicialmente enunciado estava resolvido ou mitigado.

Para ter esta perceção, foi pedido a um grupo de utilizadores que realizassem questionários com ações de utilização, de forma a perceber se a usabilidade e a compreensão dos dados apresentados estão no nível pretendido.

No final da realização do questionário de cada utilizador, houve um espaço para que dessem sugestões e *feedback* acerca da sua experiência de utilização.

Capítulo 4

Altycs

Para resolver o problema na visualização das métricas foi necessário desenvolver um planeamento detalhado para atacar o problema enunciado anteriormente.

Este planeamento começa pela escolha das tecnologias a utilizar que melhor se adequam ao problema.

Estudaram-se as melhores formas de passar informação. Estilos, cores, tipos de gráficos etc. O objetivo desta fase foi adquirir conhecimentos científicos para aplicar no aspeto do produto.

Depois desta fase, foi feito um *product-backlog* para definir as barreiras e funcionalidades específicas do produto.

Para o desenvolvimento do produto foi adotada uma metodologia SCRUM de trabalho de forma a gerir o projeto e manter os prazos controlados.

Por fim, todas as temáticas abordadas são postas em prática no problema específico.

4.1 Tecnologias

As tecnologias de *front-end* são muito discutidas, cada uma com as suas caraterísticas, vantagens e desvantagens. Atualmente as que assumem maior protagonismo são: React, Vue e Angular. De modo a clarificar melhor as comparações entre as frameworks (Angular e Vue) e a biblioteca (React.js) vão abordar-se os tópicos de: *Data Binding* para perceber o que poderá funcionar melhor para a personalização dos componentes por parte dos utilizadores, e a *performance* tendo em conta que é possível ter um alargado leque de clientes que produzem grandes quantidades de dados.

4.1.1 Data Binding

Data Biding é a conexão entre os dados e a visualização. Significa que os dados que definimos dentro do componente são anexados ao modelo HTML. Ou seja, aquilo que é definido ou que o componente vai buscar ás BD aparece na janela de visualização.

4.1.1.1 One Way Data Biding

One Way Data Biding significa que os dados fluem numa única direção. Sempre que os dados são atualizados no componente, a visualização também atualiza.

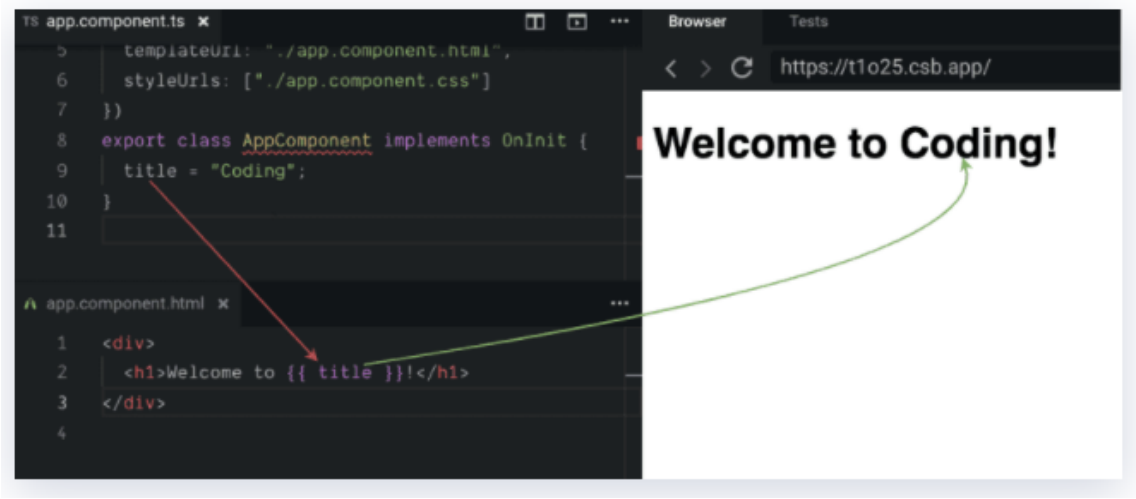


Figura 4.1: Funcionamento *One Way Data Biding*

Nesta imagem é possível visualizar que no componente é definido um *title* "Coding"que depois é chamado no HTML dentro do duplo parentese curvo. Sempre que o *title* for alterado, automaticamente o HTML passa a chamar a nova denominação e a visualização é atualizada. Isto no sentido do componente para a janela de visualização.

Para fazer atualizações da janela de visualização para o valor de uma variável dentro do componente é necessário usar uma técnica denominada de *event binding*.

```
<button (click)="myFunction()">Show alert</button>
```

Figura 4.2: HTML com chamada da função *"myFunction()"*

Considere-se que no nosso ficheiro HTML existe um botão que quando clicado chama a função *myFunction()* que está definida no componente.

```
myFunction(): void {
    alert('Show alert!');
}
```

Figura 4.3: Função no componente

A função quando ativada dispara um alerta na janela de visualização com a mensagem *"Show alert"*

4.1.1.2 Two Way Data Biding

Two Way Data Biding significa que os dados fluem em duas direções. Quer isto dizer que, quando ocorrer uma alteração na visualização, atualiza os dados e qualquer alteração dos dados atualiza a visualização.

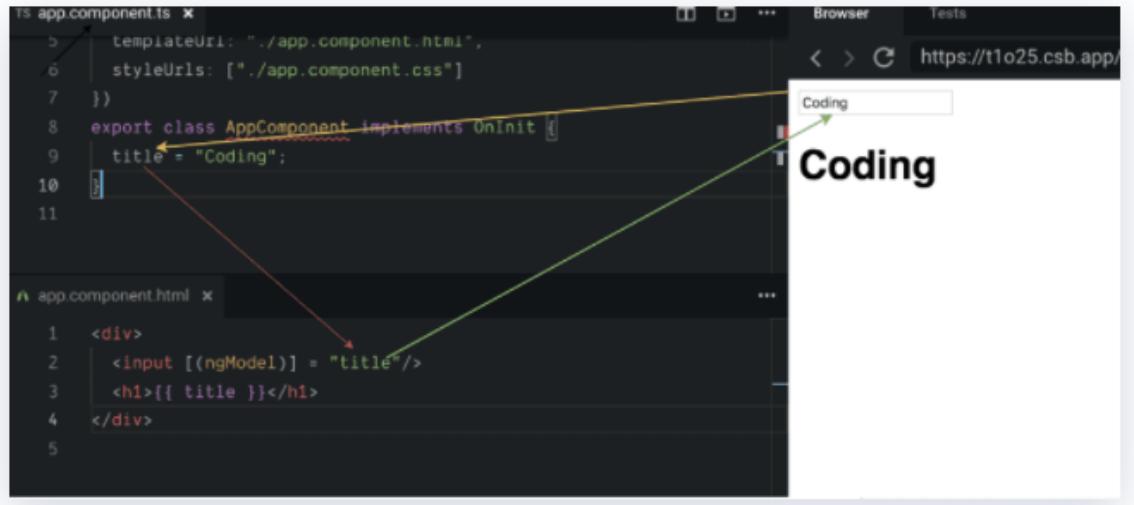


Figura 4.4: Funcionamento *Two Way Data Biding*

Nesta imagem, é definido um *title* inicial: "Coding". O ficheiro HTML tem um *input* que dá valor ao *title*. Em baixo irá aparecer o valor do *title*

Sempre que na visualização, o valor do *input* for alterado o valor do *title* também é atualizado. Se fosse escrito "Tese de mestrado"no *input* o valor do *title* passaria a ser "Tese de mestrado". Consequentemente em baixo do *input* apareceria "Tese de mestrado"em *negrito*.

Da mesma forma, sempre que atualizarmos o valor do *title* no componente a janela de visualização também será atualizada.

4.1.1.3 As tecnologias e o seu Data Biding

Uma grande diferença entre React e Angular é a utilização de *one-way* vs. *two-way* binding. Vue.js oferece suporte para ambas as ligações, one-way-binding e two-way binding.

Tendo em conta que a interação entre plataforma ABC e pessoa vai funcionar constantemente nos dois sentidos, por exemplo, o utilizador quer atualizar o título para a visualização de uma métrica (sentido visualização para componente) e a visualização do valor que as métricas apresentam (sentido componente para visualização), a abordagem de *Two Way Data Biding* é a mais apropriada.

4.1.2 Performance e desempenho

React e Vue utilizam V-DOM, que melhora o desempenho. Vue é conhecida por superar Angular e a React no desempenho, no entanto Angular pode oferecer o melhor desempenho em

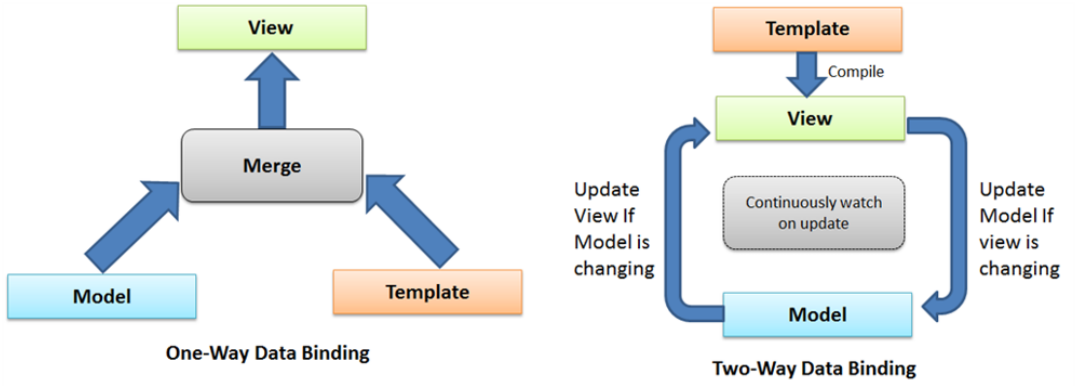


Figura 4.5: Funcionamento *One Way Data Biding* e *Two Way Data Biding*

aplicações de grande escala. Exatamente o pretendido neste caso em particular. ABC é uma aplicação que lida com muitos clientes que produzem grandes quantidades de dados.

A performance é crucial para o bom desempenho do componente. No entanto, a plataforma ABC pode atingir milhares de utilizadores. É essencial que esteja preparada para lidar com este cenário de larga utilização.

Angular perde em termos de *performance* para React e Vue. Mas, perante o cenário de utilização em grande escala é aquele que dá melhores garantias.

4.1.3 Comparação de outros fatores

Todas as tecnologias apresentam uma forte estabilidade no mercado, suportadas por enormes comunidades, sendo que a comunidade da Vue, não é enorme, mas grande o suficiente.

Todas apresentam boa documentação e são orientadas a componentes.

Relativamente à facilidade de aprendizagem, Vue acaba por apresentar uma curva de aprendizagem mais fácil, a aprendizagem de React é relativa, pois a linguagem utilizada é o conhecido JavaScript e Angular, acaba por requerer a aprendizagem de uma nova linguagem, o TypeScript.

Angular, permite a criação de aplicações *Single-Page-Aplicattion* (SPA), enquanto que React e Vue necessitam de integração com outras bibliotecas para "conseguirem" criar aplicações SPA.

Por fim, Angular e React são capazes de obter melhor desempenho em aplicações de grande escala, comparativamente com Vue.

4.1.4 Escolha da tecnologia

Perante o contexto apresentado, Angular revelou ser a tecnologia que melhor pode responder a este problema.

Angular é uma framework que permite desenvolver aplicações web e móveis. Ao nível da linguagem, as aplicações Angular podem ser desenvolvidas em TypeScript, que possui um foco muito forte na utilização de programação orientada a objetos, tornando a framework mais apelativa para programadores de *back-end*. No entanto, no momento da compilação, o compilador

Biblioteca/Framework	React	Vue	Angular
Estável	Sim	Sim	Sim
Suportado por uma grande comunidade	Sim, comunidade enorme suportada pelo apoio do Facebook	Não é enorme, mas é grande o suficiente e é suportada pela Laravel e Alibaba	Sim, comunidade enorme suportada pelo apoio do Google
Boa documentação	Sim	Sim	Sim
Fácil de aprender	Relativo	Sim	Requer aprendizagem de TypeScript
Baseada em componentes	Sim	Sim	Sim
Framework SPA	Não	Não	Sim
Bom desempenho em aplicações de grande escala	Sim	Não	Sim

Figura 4.6: Comparação entre *React, Vue* e *Angular*

compila os programas e emite para linguagem JavaScript, para que estes possam ser executados em diferentes ambientes de execução, visto que os browsers não "percebem" TypeScript.

A linguagem TypeScript, oferece vantagens como static typing, que permite a deteção de bugs à medida que os programadores escrevem scripts, sendo uma melhor opção para projetos de grandes dimensões, tal como a plataforma ABC, porque o refactoring com ferramentas TypeScript é mais fácil e mais rápido, oferece melhor colaboração através de type safety, que é uma funcionalidade que deteta erros durante o desenvolvimento de código e para além disso, o auto complete e o dynamic type podem ajudar a aumentar a produtividade dos programadores.

Esta framework que permite a criação de aplicações Single Page Applications (SPA), necessita de uma forma para "trocar de página". É possível criar um sistema de rotas, que quando solicitadas atualizam apenas partes da página, mantendo sempre a mesma base. Ou seja, em aplicações SPA, apenas parte da view necessita ser alterada, diminuindo assim o tráfego de requisições. Transpondo isto para o caso de um dashboard significa que, quando é necessário atualizar ou fazer alguma operação no dashboard não há necessidade de ter uma ação no dashboard completo, mas apenas no componente específico.

4.2 Impacto visual

É pretendido que a visualização das métricas seja de fácil acesso e compreensão. Para que o utilizador possa mais facilmente retirar o máximo de informação da visualização das métricas, é necessário melhorar o aspeto visual do componente.

Existe uma grande quantidade de utilizadores da plataforma ABC. Cada um quererá consultar métricas que não é possível prever quando é que elas serão úteis e descartáveis no espaço temporal. Como tal, o dashboard tem de permitir a personalização por parte dos utilizadores. O objetivo é que o componente se adapte conforme as necessidades. Não se pretende sair de uma configuração de listagem aleatória, e passar para uma configuração fixa de representação das métricas num dashboard através de componentes gráficos. As métricas apresentadas devem ser uma escolha do utilizador.

Para que o impacto visual faça sobressair o valor das métricas, foram estudados os aspetos a ter em conta na representação gráfica e funcionamento de forma a que componente seja *User-Friendly*.Um componente *User-Frienldy* é um componente que apresenta uma facilidade de intereção entre pessoa e computador a fim de realizar uma ação.

4.2.1 Personalização

A customização pretende oferecer ao utilizador a possibilidade de configurar a apresentação dos dados ao seu jeito. Pretende-se ir de encontro aos seus intuitos, oferecer uma visualização personalizada e a possibilidade de visualizar, para os mesmos dados, formas diferentes de apresentação como por exemplo, visualizar num gráfico de barras ou num gráfico circular.

A customização pode ser perigosa para os programadores. No fundo tudo pode ser personalizável, e se não for bem definido aquilo que será ou não personalizável poderá entrar-se numa espiral de desenvolvimento difícil de terminar.

Para além deste perigo, uma exagerada oferta de personalização do componente pode complicar a usabilidade. Se por cada clique o utilizador tiver receio naquilo que está a clicar, porque pode entrar em menus de configuração sem justificação, a usabilidade diminui.

Outro perigo associado á customização é que podem acontecer erros que os programadores do componente não previram. Quanto mais alargada a oferta de personalização, maior é o risco do ecrã ficar azul. Muitos dos possíveis erros podem ser limitados. Um exemplo prático é por exemplo, a escolha das métricas ser feita através de uma seleção e não colocar o utilizador a escrever. Aqui abre-se a porta para que seja escrito algo que não está previsto. Deste modo já conseguimos minimizar o aparecimento de erros.

Transpondo estes cuidados para o problema em específico conclui-se que a customização oferecida será bebida de configuração. O que o utilizador poderá configurar:

- Tamanho do gráfico. O utilizador pode aumentar o tamanho do gráfico se quiser visualizar melhor um componente ou vê-lo destacado.

- Movimentação e ordenação dos gráficos. A disposição no dashboard é controlada pelo utilizador.
- Visualizar para os mesmos dados, um tipo de gráfico diferente.
- Adicionar novos widgets com as métricas que pretender. Selecionar os dados (através de uma combo box, de forma a evitar erros e disponibilizar apenas os dados que podem ser apresentados) e, consoante os dados, apresentar os gráficos possíveis. Por exemplo, se o utilizador criar um novo gráfico e quiser ver o total de mensagens a única opção que vai ter para visualização será um counter. Não faz sentido ver o total de mensagens num gráfico de barras sendo que apenas tem uma variável.
- Eliminar widgets

4.2.2 User-Friendly

O componente dos dashboards tem de ajudar o utilizador a simplificar e facilitar a análise dos dados. A usabilidade e a fluidez com que se navega na plataforma é essencial para que as tarefas não sejam um quebra-cabeças.

Existem várias regras que se podem seguir no desenvolvimento de forma a garantir que o componente é *User Friendly*. O conjunto de regras mais conhecidas e seguidas são as regras de Jakob Nielsen. Jackob Nielsen propôs um conjunto de 10 regras a seguir para a construção de qualquer interface. O objetivo é que a experiência de utilização seja aprimorada e menos cansativa.

Adaptando as regras de Nielsen ao contexto do problema conclui-se que existem seis linhas orientadoras que devem prevalecer em todo o desenvolvimento. São elas:

- **1. Ponto de correspondência entre o sistema e o mundo real:** O sistema tem de comunicar na linguagem do utilizador. Utilizando palavras e conceitos com os quais o utilizador está familiarizado. Seguir as convenções do mundo real, fazendo com que as informações apareçam numa ordem natural e lógica.
- **2. Controlo do utilizador:** Muitas vezes os utilizadores enganem-se naquilo que querem fazer. É preciso garantir sempre uma ''saída de emergência" claramente sinalizada para deixar o estado indesejado. Suporte para desfazer.
- **3. Consistência e standards:** Os utilizadores não devem ser confrontados com palavras, situações ou ações diferentes, que significam a mesma coisa. Ou seja, palavras diferentes para chegar ao mesmo resultado
- **4. Prevenção de erros:** Melhor do que boas mensagens de erro é um design cuidadoso que evite a ocorrência de problemas. Eliminar as condições sujeitas a erros e verificá-las. Apresentar aos utilizadores uma opção de confirmação antes de se comprometerem com ela.
- **5. Design estético e minimalista:** Os diálogos não devem conter informações irrelevantes ou raramente necessárias.

- **6. Ajudar os usuários a reconhecer, diagnosticar e recuperar de erros:** As mensagens de erro devem ser expressas em linguagem simples (sem códigos), indicar precisamente o problema e sugerir uma solução de forma construtiva.

4.3 Desenho do componente

Os pontos anteriores refletem toda a informação necessária a ter em conta para a construção da solução. Antes de a fase de desenvolvimento ter início, é importante construir um guião para dar suporte a esta fase. Definir os limites do componente para não existirem dúvidas do objetivo e daquilo que o componente vai ser capaz. Para além das funcionalidades bem definidas é fulcral ter atenção ao aspeto visual.

Para definir as funcionalidades do componente foi feito um *product-backlog* sob a forma de *user stories* onde foram definidas todas as funcionalidades do componente.

Para o aspeto visual do componente elaborou-se *mockups* para ter uma noção do estilo do componente.

4.3.1 Product Backlog

As *user stories* seguem a seguinte estrutura: **Como** <utilizador> **Quero** <capacidade> **Para** <benefício>. Através desta estrutura será definido todas as funcionalidades da solução. São elas:

1. **Como** utilizador, **Quero** ter um header, **Para** facilitar a personalização do dashboard
2. **Como** utilizador, **Quero** redimensionar o tamanho de um widget, **Para** conseguir ver melhor destaca-lo
3. **Como** utilizador, **Quero** colocar widgets noutra posição do dashboard, **Para** ter o dashboard organizado
4. **Como** utilizador, **Quero** criar um novo widget, **Para** visualizar a métrica pretendida
5. **Como** utilizador, **Quero** ter opções de criação de vários tipos de gráficos associados ás métricas, **Para** visualizar dados da forma que desejar
6. **Como** utilizador, **Quero** ter a opção de fazer download de um gráfico, **Para** utilizar essa informação fora da plataforma

4.3.2 Mockups

O objetivo das *mockups* é projetar a aparência de componentes de *software*, dispositivos etc. Passar para o papel a aparência que está idealizada na cabeça para depois passar á fase da conceção tendo por base aquilo que foi projetado.

No que a *software* concerne, é uma fase importante para que os *developers* e os clientes projetem aquilo que está nas suas mentes. É conhecida a dificuldade existente de perceber aquilo que o cliente quer exatamente e de o projetar.

O que sai da construção das *mockups* servirá de itinerário para os *developers*. A partir daqui será traçado um caminho que deverá ser seguido para chegar ao final. As mudanças que possam ocorrer depois de projetar o componente podem implicar mudanças profundas que atrasam a fase de desenvolvimento. Os efeitos das alterações podem ser variados sendo que por norma existe um "efeito bola de neve".

É difícil de cumprir na integra todo aquilo que está projetado, mas, esta fase requer uma atenção cuidada e redobrada. Uma boa projeção do componente pode revelar-se crucial para que todas as fases posteriores sejam bem suportadas e se desenrolem melhor.

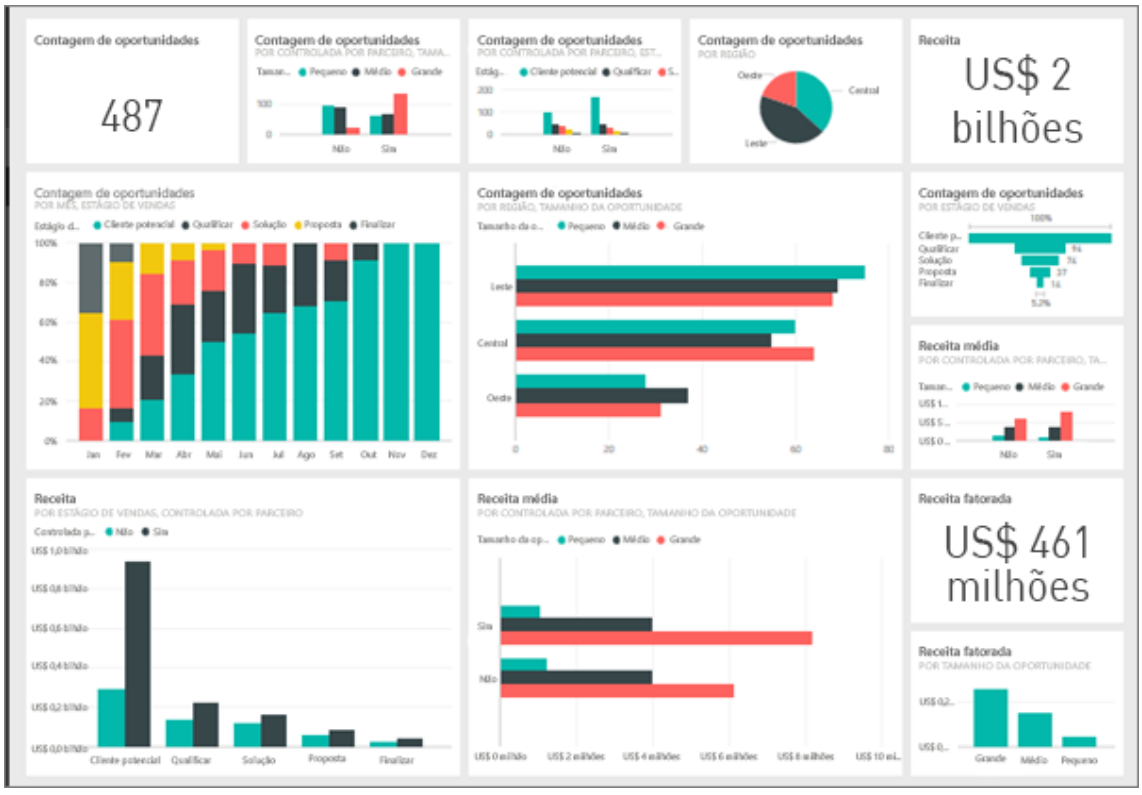


Figura 4.7: Mockup do menu do dashboard

Esta *mockup* na figura 4.7 é uma perspetiva do menu inicial do *dashboard*. Múltiplos gráficos, com várias métricas representadas e com diferentes tamanhos. O utilizador constrói assim um dashboard dentro das suas pretensões. Tem a opção de expandir ou reduzir os tamanhos das janelas que contêm os gráficos. O objetivo é que tenha o seu dashboard com informação que quer visualizar numa determinada forma de apresentação, no tamanho que achar mais apropriado para a visualização e na conjugação com as outras janelas.

Na figura 4.8 é possível visualizar a perspetiva dos campos e das opções para adicionar um widget. O utilizador irá dar um nome, selecionar a métrica, escolher se quer os dados em agregação

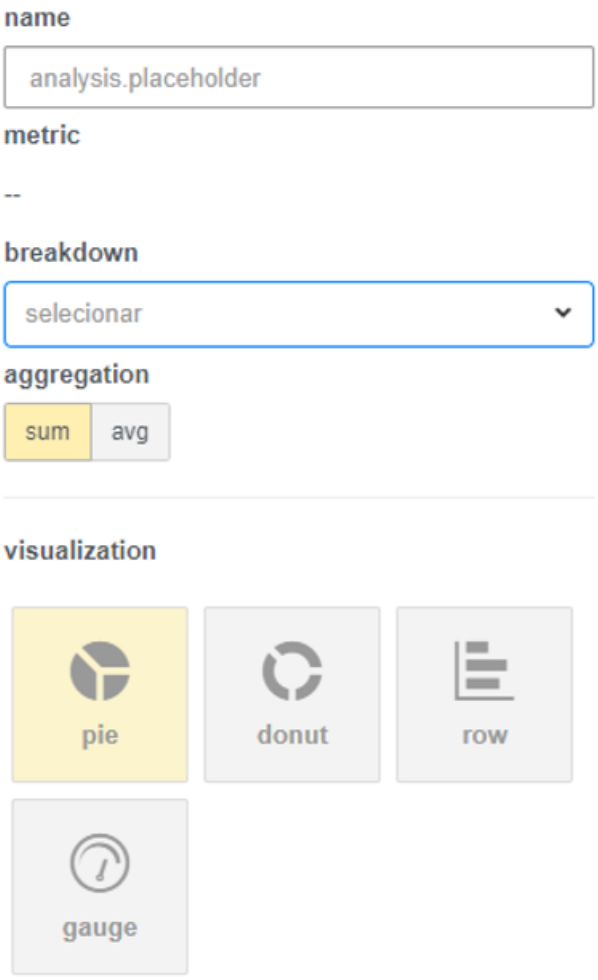


Figura 4.8: Mockup para adicionar um widget

ou percentagem e selecionar o tipo de visualização.

A *mockup* da figura 4.9 pretende mostrar as opções de *zoom in* e de *zoom out*. De forma a cumprir uma das regras definidas em 4.2.2 o icon *home* deverá colocar o gráfico novamente na visualização *standard*.

Na *mockup* 4.10 são apresentadas as seguintes opções de download que devem ser possíveis para descarregar a informação. O utilizador pode ter necessidade de utilizar a informação fora da plataforma.

4.4 Metodologia para o desenvolvimento

Para o desenvolvimento da solução será utilizado a estrutura metodológica SCRUM. Uma metodologia ágil para gestão e planeamento de projetos de software. No SCRUM, os projetos são divididos em ciclos chamados de Sprints - desenvolvimento iterativo. Cada sprint pode variar

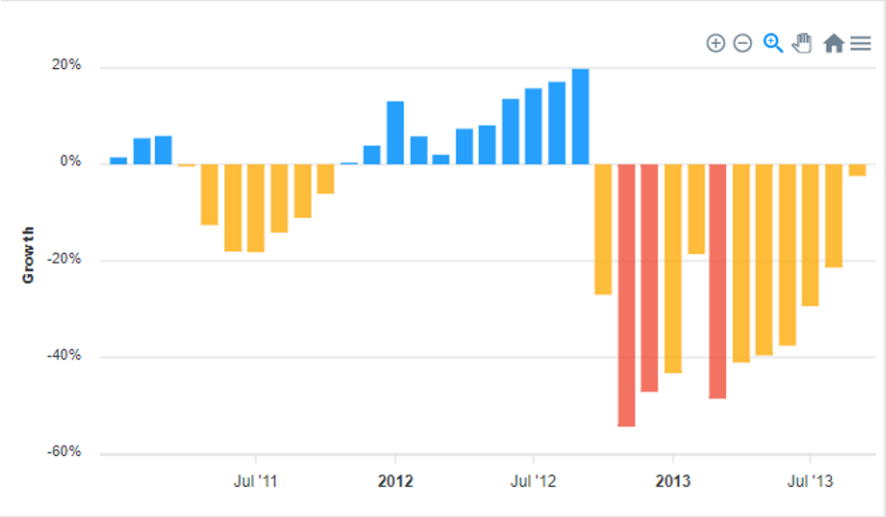


Figura 4.9: Mockup com opções de visualização

entre duas ou três semanas. A cada iteração, pretende-se mostrar o estado atual, para que caso algum desenvolvimento não vá de encontro ao pretendido a mudança e o esforço associado seja menor. Para ajudar a adotar esta metodologia foi utilizada a ferramenta web CLICKUP. De forma a ter tudo bem documentado, planeado e um suporte ao desenvolvimento.

4.5 Divisão do problema

A construção de componentes de software envolve vários estágios de resolução de pequenos problemas que no fim culminam na resposta a um problema maior.

A fase de conceção pode ser comparada á construção de um puzzle. Temos uma imagem na capa, que é o nosso objetivo final. Normalmente começando pelos rebordos por serem as peças mais fáceis de encaixar. Depois vai-se colocando as peças que vão encaixando sucessivamente até não restar nenhuma peça fora do puzzle.

Os rebordos são as fronteiras do componente. O que está dentro do puzzle são as suas funcionalidades. Tudo o que ficar de fora não deve entrar mais. Com este exemplo é possível perceber a dificuldade de, após a fase de definição das barreiras e das funcionalidades do componente, acrescentar novas peças. Uma nova peça na fronteira implica novas peças no meio para que o puzzle fique completo e equilibrado, sem buracos no meio.

Para que a construção seja mais organizada e rentável, as peças são divididas por "áreas". São identificadas as caraterísticas semelhantes como as cores, desenhos etc. Se uma parte do puzzle é o céu, junta-se as peças com a cor azul. A divisão de um grande problema em pequenos problemas. Vai-se agrupando as peças por grupos onde se vão encaixar. Encarando pequenos problemas de cada vez.

Á medida que os pequenos problemas vão sendo resolvidos, são encaixados no puzzle. Ou seja, os componentes de software vão comunicando entre si de forma harmoniosa, tal como as

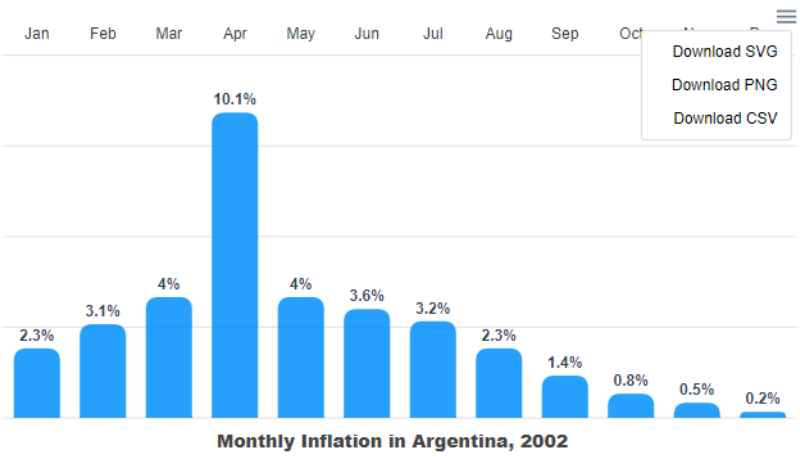


Figura 4.10: Mockup com opções de download

peças se encaixam. Funcionando como um todo. Não é possível ter todas as peças encaixadas entre si, o importante é estarem encaixadas no local certo encostadas nas peças vizinhas corretas. Os componentes constituintes do software não comunicam todos entre si. Comunicam apenas com os componentes com os quais necessitam.

Por vezes acontece de algumas peças virem encaixadas, ou por magia encontrarem-se dentro da caixa e quando as retiramos estão encaixadas, resolvendo assim uma parte. Quer isto dizer que alguns dos problemas que são enfrentados para a construção do componente podem já estar resolvidos. Existem frameworks e bibliotecas que se adequam ao problema e que podem ser utilizadas no componente. Poupando tempo e esforço.

Para o problema específico é possível identificar dois componentes gráficos fundamentais:

- Uma grelha que permita aumentar e diminuir o tamanho dos widgets, posicionar os widgets de forma dinâmica e adicionar/eliminar.

- Os gráficos que ficam dentro das janelas da grelha.

4.6 Bibliotecas

As bibliotecas são extramente úteis. Ajudam a poupar tempo e esforço na implementação de funcionalidades. Com a digitalização do mundo e com a quantidade de informação a circular é inevitável não olhar para esta ferramenta tão útil.

A roda já foi inventada á um par de anos. A forma como é utilizada é que pode variar. Pode ser usada para mover veículos como para içar materiais. O seu formato pode variar em tamanho, peso, material de construção etc., mas o conceito é o mesmo.

Na programação acontece o mesmo. Grande parte dos problemas já foram discutidos em fóruns e é possível encontrar a sua resolução na *web*. Claro que existe sempre a adaptação da biblioteca áquilo que queremos que ele faça exatamente. Tal como a aplicação da roda aos diferentes contextos.

4.6.1 Gridster2

Para ajudar na parte da grelha para que o utilizador posicione os widgets e os redimensione conforme as suas pretensões existe a biblioteca *Gridster2* para ajudar.

Já preparada em *Angular* a sua integração é simples. O código encontra-se disponível *online* em `https://tiberiuzuld.github.io/angular-gridster2/`. Tem várias opções para configurar a grelha como tamanho das colunas e das linhas. Permite acrescentar e eliminar os widgets, redimensionar e reposicionar.

Uma ferramenta ótima para que o utilizador possa personalizar o dashboard.

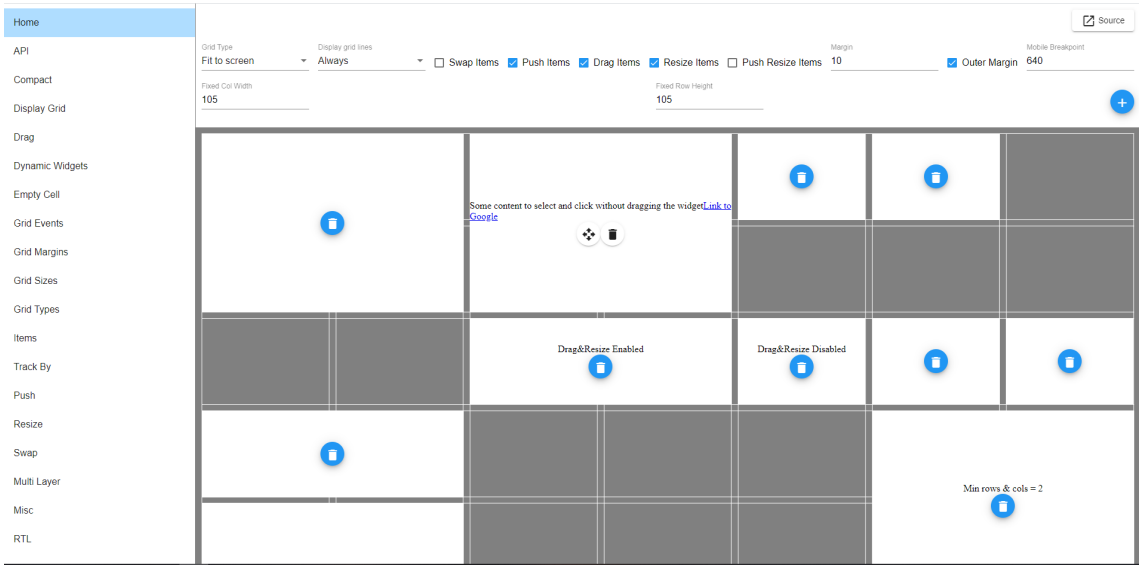


Figura 4.11: Gridster2

4.6.2 ApexCharts

Para o utilizador visualizar informação são necessários gráficos para fazer a sua leitura. *ApexCharts* é uma biblioteca que construi vários tipos de gráficos. Estes gráficos entram dentro das janelas de *widgets* da grelha.

Permite fazer integrações com tecnologias *front end* como o *React*, *Angular* e *Vue*. Também é possível fazer integração com o *back end*. Suporta tecnologias como *PHP*, *Django* e *ASP.NET*.

Para *Angular* tem 15 opções de gráficos simples como: gráfico de barras, colunas, circular etc. Pode consultar-se todas as opções em `https://apexcharts.com/angular-chart-demos/`.

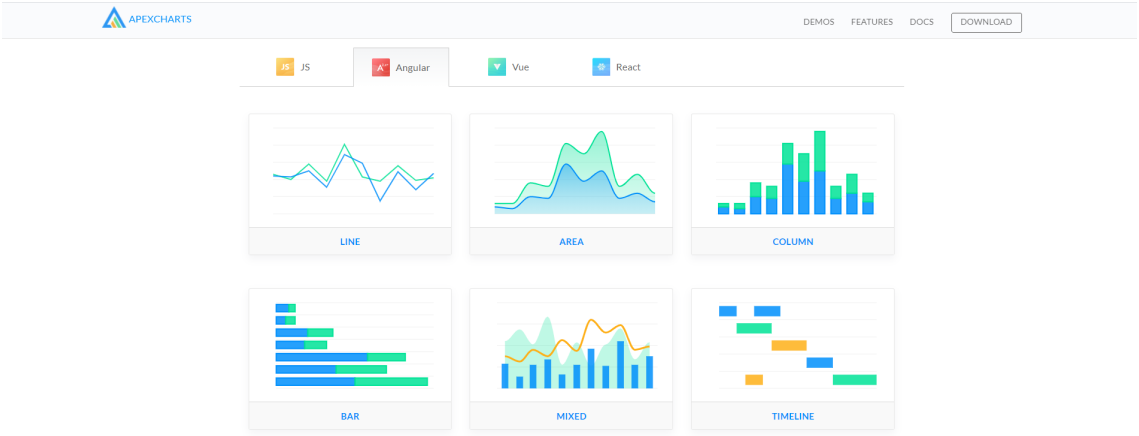


Figura 4.12: ApexChart

4.6.2.1 Dados nos gráficos

Os gráficos têm que receber dados. Após o recebimento destes dados é que os gráficos são construídos. São alinhadas as diferentes cores, a escala, os números no eixo dos X e Y etc.

Os dados são registadas em bases de dados distribuídas em PostgreSQL. É necessário ir buscar os dados ás BDs e colocá-los nos gráficos na forma correta. Com ApexCharts apenas é necessário colocar os dados que queremos no gráfico e as legendas (eixo dos X e título).

O formato para preencher o gráfico é um vetor com dados numéricos. Este formato é transversal a todos os tipos de gráficos e entra sempre na secção *series*. A imagem 4.13 ilustra um exemplo.

```
series: [
  {
    name: "Inflation",
    data: [2.3, 3.1, 4.0, 10.1, 4.0, 3.6, 3.2, 2.3, 1.4, 0.8, 0.5, 0.2]
  }
],
```

Figura 4.13: Exemplo do formato de dados utilizando ApexCharts

Após a inserção dos dados, falta a colocação de uma legenda nos eixo do X que identifique os dados numéricos bem como um título para identificar o contexto. É possível configurar a escala, os valores que aparecem na escala, os espaçamentos dentro da escala etc. No entanto é pretendido encontrar uma solução geral para um problema específico. Como existe uma grande quantidade de tipos de dados que podem variar entre as unidades, dezenas, centenas e milhares no valor, o foco é em ter os dados no formato correto e uma legenda que os identifique. O título fica a encargo do utilizador. A legenda é sempre inserida na categoria de *xaxis* também transversal a todos os tipos de gráficos.

O gráfico vai pegar na legenda e distribuí-la no eixo dos X conforme ela tenha sido inserida. Mantem a ordem, assim como faz com os dados do gráfico. Ou seja, aquilo que seja inserido em

```
xaxis: {
  categories: [
    "Jan",
    "Feb",
    "Mar",
    "Apr",
    "May",
    "Jun",
    "Jul",
    "Aug",
    "Sep",
    "Oct",
    "Nov",
    "Dec"
  ],
```

Figura 4.14: Exemplo da legenda utilizando ApexCharts

primeiro lugar irá aparecer no gráfico em primeiro. Na imagem 4.15 podemos ver o resultado final após a inserção dos dados e da legenda.

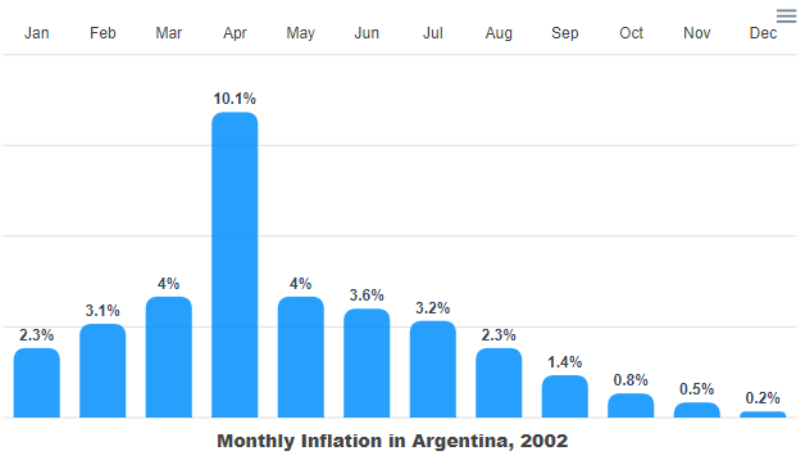


Figura 4.15: Exemplo de um gráfico utilizando ApexCharts

4.6.3 Node.js

Para fazer a ligação entre as BDs e o componente *front end* a tecnologia mais famosa é o Node.js. Com uma execução JavaScript assíncrona orientada a objetos, está desenhado para aplicações escaláveis.

Esta tecnologia irá ajudar os gráficos a terem os dados no formato correto para fazerem a apresentação ao utilizador. Fazendo a transição dos dados nas BDs para a sua visualização nos gráficos.

O objetivo é construir as queries para que todas as requisições de dados pelo utilizador sejam respondidas num formato de dados que se encaixam nos gráficos ApexCharts.

Assim, quando for necessário entregar dados o componente Angular comunica com o node.js enviando-lhe os dados que pretende. O node.js irá responder com os dados (dados + legenda) no formato para encaixar nos gráficos.

4.6.4 Knex.js

O Knex.js é uma tecnologia que permite construir queries para as base de dados em Postgres, MSSQL, MySQL, MariaDB, SQLite3, Oracle, e Amazon Redshift. Suporta TypeScript (a mesma tecnologia usada em Angular).

Apesar de não ser o prepósito do Knex.js, apresenta um conjunto de caraterísticas que ajudam a fazer um projeto de queries quando já existe uma BD desenvolvida e com uma dimensão volumosa. Não requer um grande estudo das BDs para fazer uma construção das queries.

Apresenta um suporte de documentação excelente que pode ser consultado em `https://knexjs.org/`. Toda a construção de queries encontra-se bem documentada e contempla uma panóplia de condições alargada. Qualquer ação pretendida está documentada.

```
knex('users').where({
  first_name: 'Test',
  last_name:  'User'
}).select('id')

Outputs:

select `id` from `users` where `first_name` = 'Test' and `last_name` = 'User'
```

Figura 4.16: Exemplo de uma querie utilizando Knex.js

Na imagem 4.16 fica um exememplo da construção de uma querie. Seleciona-se a tabela, assim como as características que se pretende com as respetivas condições.

4.7 Idealização da estrutura

Para ser possível o utilizador visualizar as métricas existe todo um mecanismo por detrás. A forma de organização padrão da tecnologia Angular é colocar os componentes como os gráficos, a grelha e outros numa pasta chamada de *components* e os componentes que vão buscar a informação numa pasta chamada de *services*. Cada componente chama o serviço que quer subescrever para obter informação. Cada componente pode subescrever mais do que um serviço para obter informação.

Estruturam-se os serviços conforme os grupos existentes de *"SMS and Call"*, *"Hunt-Group"* e *"Waiting Queue"*. Ou seja, cada grupo tem um serviço isolado dos outros. Caso um falhe, o outro não depende disso e pode continuar a funcionar.

Os serviços vão ser subscritos por cada tipo de gráfico, que solicitarão a informação pretendida pelo utilizador. Os serviços atendem ao pedido e respondem com os dados.

Ao efetuar alterações nos tamanhos dos widgets que contêm os gráficos, é necessário que os gráficos atualizem o seu tamanho, ocupando o novo tamanho da janela do widget. Para isto acontecer a grelha dispara um evento que enviara para o gráfico correspondente para que este atualize o tamanho de acordo com a nova configuração.

4.7.1 Passagem de dados

Visto existirem outras prioridades no desenvolvimento, que não passam por desenvolver as APIs para os dados dos gráficos, a alternativa foi temporariamente colocar este componente a ir buscar os dados através de um projeto *Node.js*. Assim que a equipa conseguir desenvolver as APIs com a estrutura de dados que os gráficos necessita, os serviços passam a chamar essas APIs ao invés de o projeto *Node.js*.

Conectam-se á API do projeto *Node.js* e enviam a métrica da qual querem receber os dados, bem como o id do cliente que o utilizador pode selecionar. Caso o utilizador não selecione nenhum cliente a informação que receberá será um agregado. Por exemplo se selecionar as chamadas recebidas, e não selecionar nenhum cliente, vai ter a informação de todas as chamadas recebidas de todos os clientes.

O projeto *Node.js* faz a conexão diretamente á BD. Recebe a informação da métrica e executa a querie necessária para obter os dados. A resposta segue sempre o mesmo formato. Uma legenda e os dados. Desta forma os dadas já se encontram no formato pedido pelos gráficos.

Os serviços recebem a informação, e passam-na para os componentes dos gráficos. Estes componentes pegam na informação recebida e colocam-na nos gráficos que já se encontra no formato correto.

4.7.2 Personalização da grelha

Com a grelha o utilizador arrasta os itens para diferentes posições e aumenta e diminui o tamanho conforme o desejado. Esta é a base para conseguir personalizar o dashboard e colocar as janelas na disposição que achar mais conveniente.

Além desta personalização o utilizador tem ao seu dispor mais opções que pode alterar. A figura 4.17 ilustra as opções que o utilizador pode alterar.

Tem ao seu dispor a opção de colocar linhas na grelha permanentemente, sem linhas ou com linhas apenas quando movimenta widgtes ou faz alterações no tamanho.

Formatar o valor da margem entre os widgtes escrevendo o valor.

Alterar o tamanho das colunas e das linhas da grelha em pixéis.

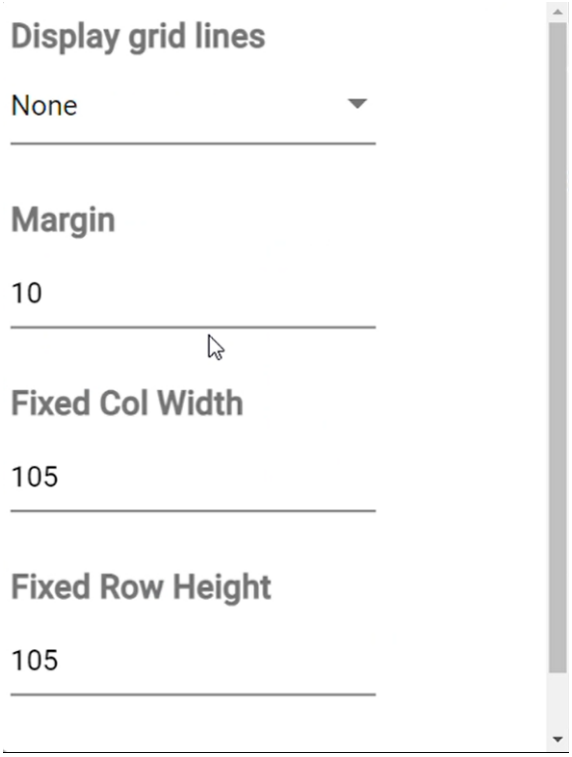


Figura 4.17: Opções de personalização da grelha

4.7.3 Enquadramento de um gráfico na janela de widget

Os gráficos têm que aparecer dentro das janelas de widgets da grelha. Ao enquadrar um gráfico dentro desta janela, o espaço fica praticamente todo preenchido. Isto levanta um problema na movimentação.

A janela de widget está preparada para que em caso de clique na janela e deslize (clicar + movimentação), a janela acompanha a movimentação do rato, até que solte o botão e a janela pousa na posição em que o rato estiver nesse momento.

Colocando gráficos nessa janela, a janela fica preenchida quase na totalidade por esses gráficos. Estes gráficos apresentam funcionalidades de interação. Por exemplo se o utilizador fizer *over* com o rato numa zona de informação, o gráfico destaca essa zona e sombreia as outras.

Ou seja, a zona de clique para movimentação fica muito reduzida. Dificultando que o utilizador movimente os widgtes.

A solução deste problema passa por colocar um ícone que funciona exatamente da mesma forma. Quando o utilizador quiser movimentar o widget clica nesse ícone, movimenta e larga na posição desejada. Deste modo não anda a clicar ao calhas na janela á procura da área que permita a movimentação.

A figura 4.18 ilustra a área ocupada por cada elemento presente numa janela do widget. A restante área é aquela que permite clicar e movimentar. Se fosse excluído o ícone de movimentação que está no canto superior direito seria muito difícil o utilizador conseguir mover as janelas.

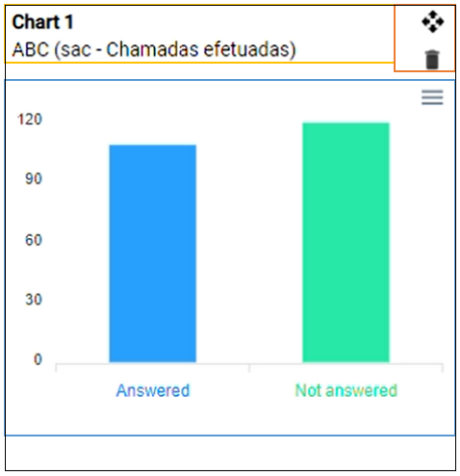


Figura 4.18: Área ocupada pelos constituintes de um widget

4.7.4 Menu para adicionar um widget

Para que seja possível adicionar widgets ao dashboard é necessário um menu que apresente as opções ao utilizador. Uma forma visual que o próprio consiga facilmente entender e interagir.

As opções que deverá ter acesso para personalizar são:

- Colocar um título á sua escolha
- Selecionar o grupo de métricas e a mética
- Escolher (ou não) um cliente
- Selecionar o tipo de gráfico onde quer ver representado as métricas que selecionou.

A figura 4.19 apresenta o menu onde o utilizador percorre cada uma das opções para introduzir novos widgets ao dashboard.

4.8 Dashboard

Após a análise de todos os componentes necessários e da elaboração do mecanismo que está por detrás de toda a passagem de informação para os gráficos, o resultado é o dashboard que pode ser consultado na figura 4.20.

Na parte superior esquerda do dashboard temos a opção para adicionar novos widgets (pode consultar em maior detalhe em 4.7.4). Ao meio o logotipo da Altice Labs. O ícone de definições da grelha (consultar 4.7.2), que abre o menu para o utilizador fazer alterações. O símbolo de ajuda onde o utilizador pode descarregar o manual de informação, com toda a explicação necessária para trabalhar com a nova ferramenta e o ícone para fazer *log-off* da plataforma.

Os widgets são constituídos por um título que se encontra a *negrito* para o salientar. Por baixo do título está o cliente selecionado (caso não selecione aparece *overall*) e a métrica. No canto superior direito estão colocados os ícones para movimentar os widgets e o ícone para apagar.

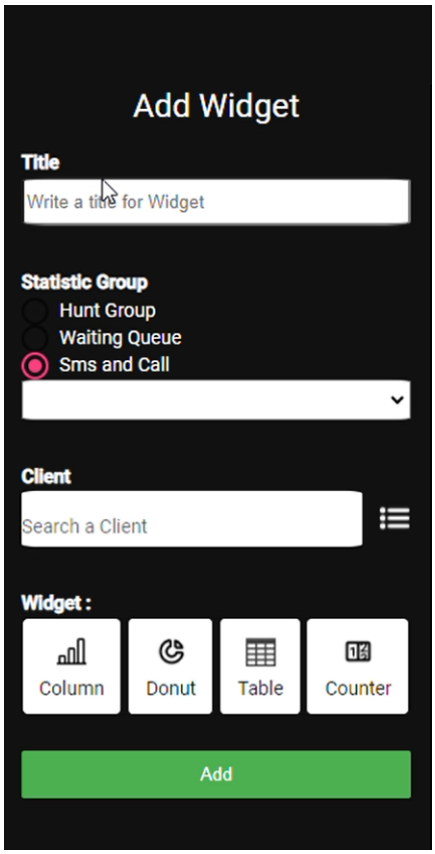


Figura 4.19: Menu para adicionar um widget

Ocupando a maior área está o gráfico que pode ser um gráfico de colunas, *donut*, tabela e um contador.

4.9 Sumário

Neste capítulo são abordadas as diferentes caraterísticas das principais tecnologias de *front-end*, tendo em conta as propriedades que melhor se encaixam no contexto da solução pretendida. São abordados os tópicos de *Data Binding* (4.1.1), a performance e o desempenho (4.1.2) e outros fatores importantes para a escolha da tecnologia. Analisando e comparando os tópicos foi escolhida a tecnologia Angular.

Depois da escolha da tecnologia foi explorado a melhor forma para que visualmente o componente fosse bem recebido pelos utilizadores. Foram definidas as linhas orientadoras de personalização (4.2.1) e as regras a seguir para ter um componente *user-friendly* tendo por base as regras de Jakob Nielsen.

Após a definição das linhas orientadoras, procedeu-se ao desenho do componente (4.3). Foi elaborado um *product backlog* sob a forma de *user stories* definindo todas as funcionalidades que o Altycs deve cumprir (4.3.1). Na secção 4.3.2 foi feito um primeiro desenho do componente com base nas funcionalidades anteriormente expostas.

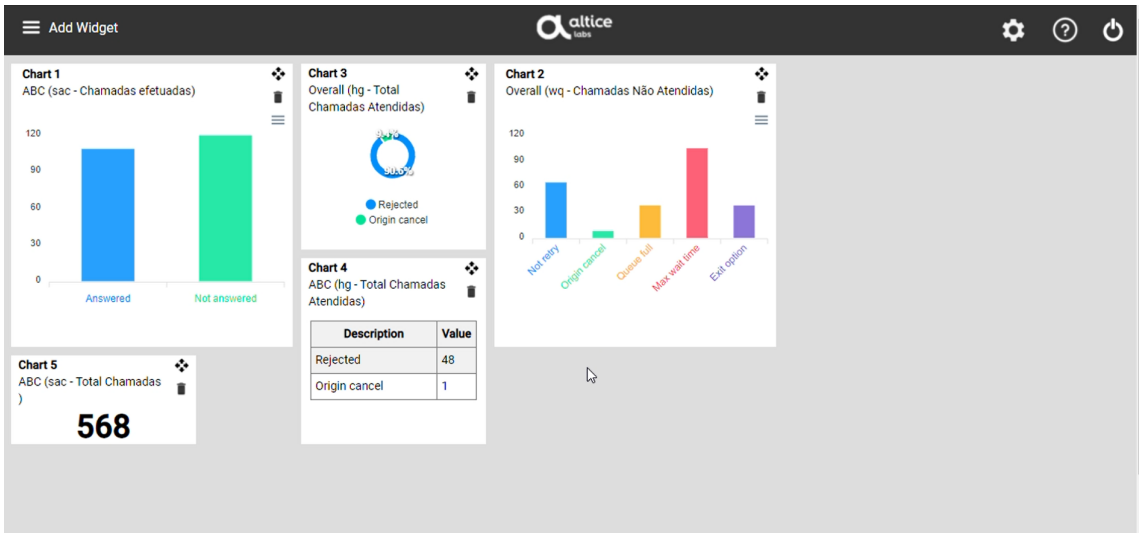


Figura 4.20: Dashboard

No subcapítulo 4.4 é explicada a metodologia que foi seguida para a fase de desenvolvimento. Adotou-se a metodologia SCRUM e utilizada a ferramenta web CLICKUP para estruturar a documentação e dar suporte ao desenvolvimento.

Foi dividido o problema como todo, em problemas mais pequenos que se interligam (4.5). Explorou-se o material que podia ser utilizado (4.6) para resolver esses problemas e explicado como iriam funcionar em conjunto.

Clarificou-se a estrutura da plataforma (4.7). Como é que a plataforma vai buscar os dados e os coloca á disposição do utilizador (4.7.1). O mecanismo que permite a personalização da grelha, os enquadramentos necessários para que tal mecanismo funcione e os menus para que seja possível adicionar novos widgets ao dashboard.

Por fim é exposta a solução e os seus menus na secção 4.20.

Capítulo 5

Testes de usabilidade

Após a conclusão da fase de desenvolvimento foi necessário por o componente na prática. Colocar os utilizadores a interagirem com a nova funcionalidade. Fazer o registo das interações e concluir se o componente resolve o problema ou parte dele e aquilo que ainda pode ser melhorado.

Para os utilizadores pouco importa aquilo que está por detrás do componente. Se foi utilizada a tecnologia A ou B e porquê. O importante é que funcione e resolva o anterior problema.

Para fazer os registos foi necessário desenvolver um teste que passa-se pelas funcionalidades do componente. Perceber se houve dificuldades na interação, fazer o registo e ouvir os utilizadores para perceber, do ponto de vista do utilizador, aquilo que querem ver melhorado.

Após uma primeira avaliação sobre a plataforma foi necessário despistar interferências que possam ter ocorrido. Se os dados de um widget não carregaram talvez a ligação internet estivesse em baixo e não tenha permitido que os dados tivessem sido carregados. Pretendia-se perceber se o ambiente em volta do teste era adequado e se, os possíveis maus funcionamentos do componente, têm a ver com o seu funcionamento ou com outros fatores. Consistiu em fazer uma despistagem às respostas do primeiro teste.

5.1 Conceção e planeamento

Uma fase que determina o sucesso ou insucesso do trabalho desenvolvido tem de ser cuidadosamente pensada e posta em prática. Desde que o problema foi abordado e se pensou numa solução, que esta fase foi contemplada no plano. Obviamente teve de ficar para o fim, após o desenvolvimento, para concluir que aquilo que foi estudado e idealizado como solução é efetivamente competente.

Para a realização dos testes foi necessário haver uma adaptação face ao estado atual da pandemia. Inicialmente pretendia-se que os testes fossem feitos de forma presencial. Como nesta fase esse cenário não era possível, foi necessário pensar numa solução que não passa-se pelo regime presencial, mas que contempla-se todos os testes inicialmente pensados.

Como tal chegou-se á conclusão que a melhor maneira seriam criar *slots* de 20 minutos para cada utilizador interagir com o componente e fazer a despistagem. As sessões, feitas remotamente via *Microsoft Teams*®, em regime de videochamada, foram divididas em 15 minutos dedicados á usabilidade do componente e 5 minutos dedicados á despistagem. A imagem 5.1 ilustra a linha orientadora que foi seguida para execução desta fase.

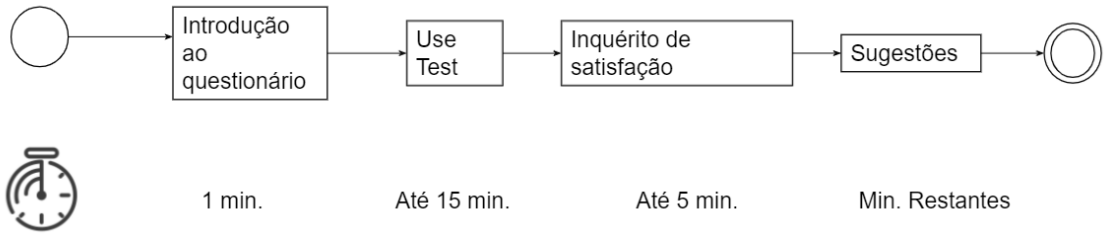


Figura 5.1: Linha orientadora para a execução da recolha de dados

Foram convidados o maior número possível de utilizadores sabendo de antemão que nem todos iriam aceitar. Dezasseis utilizadores aceitaram o desafio. As sessões foram pensadas e elaboradas para ter um equilíbrio entre demorar o menos tempo possível, percorrer todas as funcionalidades, ouvir sugestões e fazer a despistagem.

5.1.1 Recolha de dados de usabilidade

O objetivo deste teste foi perceber se a interação entre o componente e o utilizador é positiva. Avaliar se a informação presente nos gráficos passa bem a informação, sem gerar confusão ou más interpretações. Entender se conseguem facilmente adicionar novos gráficos e se as mensagens de erros ajudam efetivamente os utilizadores a resolverem problemas que possam aparecer.

Aquando da realização do teste de usabilidade a API que dava a listagem dos clientes para que, o utilizador escolhe-se os clientes que pretende, encontrava-se em manutenção. As perguntas que seriam para testar esta componente tiverem de ser suspensas.

No cenário ideal o utilizador percebe imediatamente como interagir com o componente. Não tem dificuldades na interação e abraça esta nova funcionalidade de imediato.

Num cenário obscuro o utilizador não percebe as informações nos gráficos, não sabe onde adicionar novos *widget* e tem dificuldades para personalizar o dashboard.

Para que o teste seja eficaz na avaliação era necessário passar por todas as funcionalidades e componentes. Como tal estas são as áreas que foram avaliadas:

- Perceção dos dados nos gráficos (gráfico de barras, gráfico em donut e tabela)
- Personalização do dashboard (mover gráficos, aumentar/diminuir o tamanho, eliminar)
- Adicionar novos widgets
- Personalização da grelha onde assentam os widgets

Como referido em 5.1, a realização do teste foi forçadamente remota. Como tal, a ideia foi que nas *slots*, seria partilhado o ecrã onde do utilizador que teria a plataforma á sua frente. Depois de estabelecida esta conexão em devidas condições, foi feito o teste como se de um ditado se tratasse. Foi ditado ao utilizador que fizesse determinadas ações (que contemplassem os pontos acima referidos) e foi registado o tempo que ele demorou a realizar a ação. Quando terminava a tarefa foi lhe pedido que classificasse a dificuldade que teve na realização, numa escala de 1 a 5. Sendo que 1 é muito fácil e 5 extremamente difícil.

5.1.2 Inquérito de despistagem

Existem fatores que podiam influenciar negativamente a realização do teste. Ligação á internet em baixo, ouvir as questões do teste com cortes, a partilha do ecrã congelar momentaneamente etc. Foi necessário perceber que as condições da realização do teste foram adequadas. Porque em caso negativo, isto poderá ter influenciado tanto o tempo das respostas no teste de usabilidade como a dificuldade que os utilizadores atribuíram a cada questão. Para fazer esta avaliação serão abordadas as seguintes influências:

- stress no trabalho
- agenda apertada
- conexão internet
- funcionamento da VPN
- fatores externos como mensagens no skype, carteiro a bater á porta etc.
- o ambiente ser apropriado. Ex: dimensões do ecrã, claridade etc.
- pressão pelo teste estar a ser supervisionado

O utilizador respondeu numa escala de 1 a 5. Sendo 1 Discordo Totalmente e 5 Concordo Totalmente. Desta forma foi possível aferir se os resultados estavam ou não influenciados.

5.2 Resultados

As respostas obtidas foram registadas na ferramenta *Microsoft Excel®*. Desta forma, foi possível ter os dados organizados, fazer as operações necessários e tirar as devidas conclusões. O registo segue a ordem cronológica de resposta. O primeiro utilizador a responder, corresponde ao primeiro registo tanto no teste de usabilidade como na despistagem. Portanto, o primeiro registo corresponde ao mesmo utilizador e assim sucessivamente.

Foram também registadas e organizadas as sugestões dos utilizadores. A intenção foi analisar e consequentemente priorizar o que deve ser corrigido ou melhorado.

5.2.1 Análise do teste de usabilidade

Avaliando a seguinte tabela 5.1, relativa ao tempo médio e desvio padrão do teste de usabilidade, verificamos que a primeira questão foi aquela que demorou mais tempo a ser respondida. O impacto de os utilizadores visualizarem um novo componente e não o conhecerem á partida influencia o tempo de resposta da primeira questão. É também possível constatar que apenas 3 questões das 15 no total têm um desvio padrão inferior a 10 segundos. O que permite concluir que os comportamentos de utilizador para utilizador são muito destintos. Não existindo um comportamento modelo.

Questão	Média	Desvio Padrão
1	67,00	52,94
2	13,13	22,65
3	32,13	35,82
4	37,06	32,81
5	31,50	18,30
6	50,75	39,80
7	27,67	17,52
8	8,13	6,05
9	35,92	37,08
10	10,75	11,67
11	14,81	15,44
12	39,43	35,82
13	25,18	31,74
14	5,56	0,96
15	3,38	1,62

Tabela 5.1: Média e desvio padrão dos tempos de resposta

Somando a média do tempo de todas as respostas obtemos uma valor arredondado de 7 minutos. O facto de ser bem abaixo do tempo previsto (15 minutos) é positivo e abriu a janela para que as observações e apontamentos de melhoria por parte do utilizador fossem feitos com mais tranquilidade.

1. **Colocar linhas na grelha**

   Por ser a primeira pergunta, pretendia-se perceber se o ícone para aceder às definições da grelha está facilmente acessível. Entender se o utilizador chega com facilidade às opções de personalização da grelha e se entenda as opções.

   É possível concluir que a primeira questão é aquela que demorou mais tempo a ser respondida. Como é possível observar em 5.2 a primeira questão é a única que passa da barreira dos 60 segundos a ser respondida. A conclusão a tirar daqui é que no primeiro impacto, como os utilizadores não conhecem o novo componente, a tendência foi para carregar em

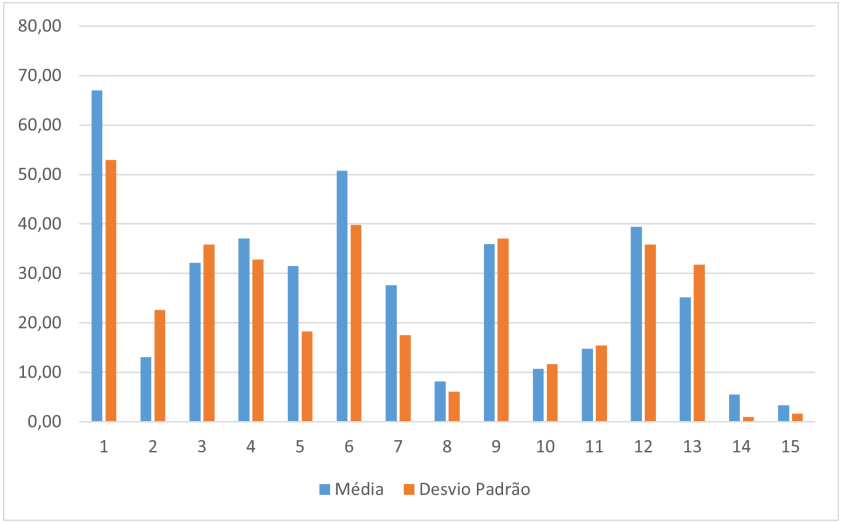


Figura 5.2: Tempo médio e desvio padrão por pergunta em segundos

todos os botões para ver o que acontece, até encontrar a solução. Aproveitam a primeira questão para fazer a "fase de exploração".

Quando se olha para a dificuldade média atribuída pelos utilizadores a esta questão nota-se que se obtém o segundo valor mais elevado. Como é possível observar em 5.3, a primeira questão teve um valor de dificuldade média de 2,06. Um valor relativamente baixo tendo em conta a escala de 1 a 5. No entanto, um dos valores mais elevados no cômputo geral.

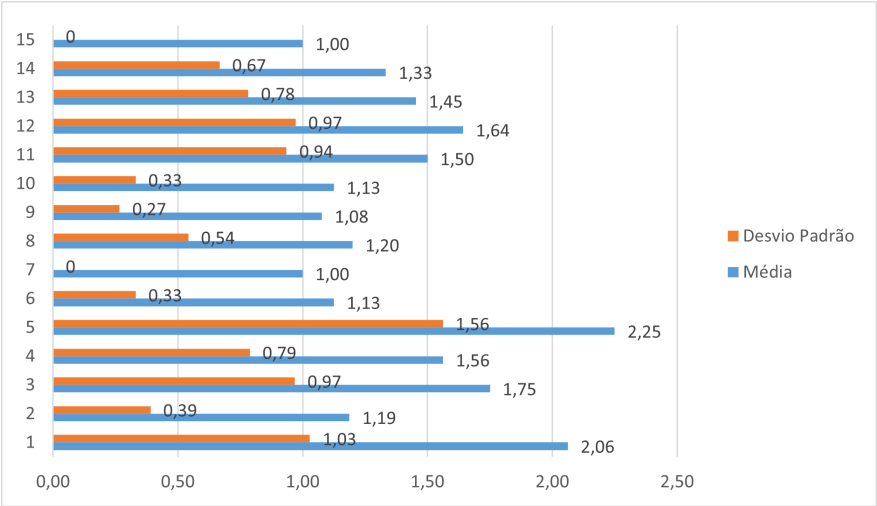


Figura 5.3: Dificuldade média por pergunta

As análises dos utilizadores foram também muito variadas. Os comentários variavam entre: "É fácil de lá chegar. Eu é que não estava a ver bem". "Demorei porque não conhecia a plataforma". "Agora que lá cheguei percebo que é simples". Obtém-se assim o segundo valor de desvio padrão mais elevado relativo ao nível de dificuldade das questões, como é possível observar em **??**

Com todos estes resultados diferenciados e influenciados, pelo facto de ser a primeira questão, pode parecer difícil tirar uma conclusão. No entanto, olhando para a frequência da dificuldade atribuída (5.4) é possível constatar que os níveis 1 e 2 de dificuldade (que representam dificuldade baixa) representam um total de 68 por cento. Logo, apesar do impacto inicial e da fase de "exploração"influenciarem o tempo de execução, a maioria dos utilizadores considera fácil encontrar as opções de personalização da grelha e colocar linhas na mesma.

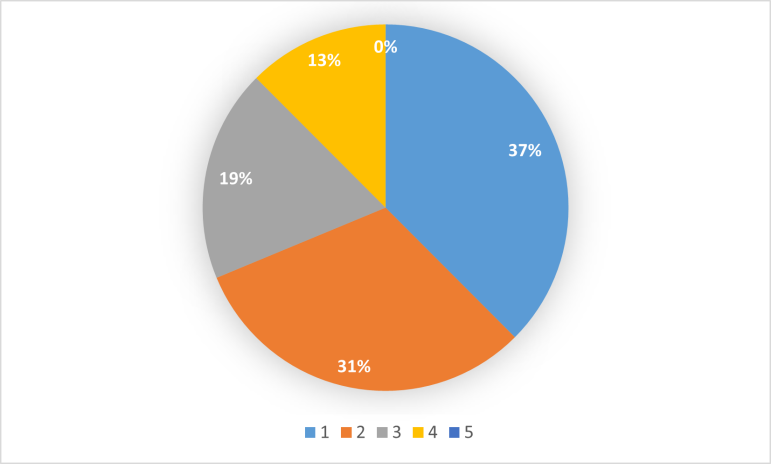


Figura 5.4: Frequência da dificuldade na questão 1

2. **Apagar o "Gráfico 1"**

   Verificar se o ícone para apagar um widget está numa posição praticável. Uma questão simples, que visa, explorar a funcionalidade de apagar widgets do dashboard.

   Os resultados obtidos são conclusivos. Através dos gráficos 5.2 e 5.3, é possível concluir que apagar um widget é uma ação fácil de se fazer e que requer pouco tempo. Facilmente os utilizadores chegaram á resposta e todos eles atribuem uma classificação de dificuldade que não ultrapassou o nível 2 (tabela 5.2).

3. **Fazer *re-scale* do "gráfico 2"até que a legenda fique na horizontal**

   A intenção era perceber se a opção para fazer os ajustes do tamanho dos widgets era alcançada espontaneamente e que a sua funcionalidade cumpre com o requisito. Verificar se o utilizador consegue manusear a funcionalidade sem dificuldade. A legenda ficar totalmente na horizontal serve como ponto de referência. Desta forma o utilizador verifica também que os gráficos se ajustam quando o tamanho do widget é atualizado.

   O *feedback* dos utilizadores é positivo. Apesar do tempo médio ainda ser elevado (32,13s) para uma operação que se quer praticamente instantânea, o seu desvio padrão também é elevado (35,82). Mais uma vez o comportamento dos utilizadores em relação ao tempo de execução é bastante diferenciado. Grande parte dos utilizadores após a conclusão da tarefa referiu que estava a complicar quando a tarefa é simples. Este facto, é devido a outras

Utilizador	Nivel de dificuldade
1	1
2	1
3	1
4	1
5	1
6	2
7	1
8	1
9	1
10	2
11	1
12	1
13	1
14	1
15	2
16	1

Tabela 5.2: Dificuldade atribuída por cada utilizador á questão 2

funcionalidades da plataforma ABC que oferecem ajuste de tamanho, mas, para o fazer a configuração é mais complexa. Daí que o tempo de execução seja mais elevado do que o esperado. Os utilizadores tinham á partida uma ideia formatada para opções de *re-scale*.

Olhando para a dificuldade atribuída (5.3) verificamos que a média é baixa (1,75). A funcionalidade para fazer *re-scale* é bem recebida pelos utilizadores. Olhando para o gráfico 5.5 constata-se que 75 por cento dos utilizadores atribuíram uma classificação de 1 ou 2 á questão, concluindo que é uma funcionalidade simples, bem implementada e que irá ajudar o utilizador.

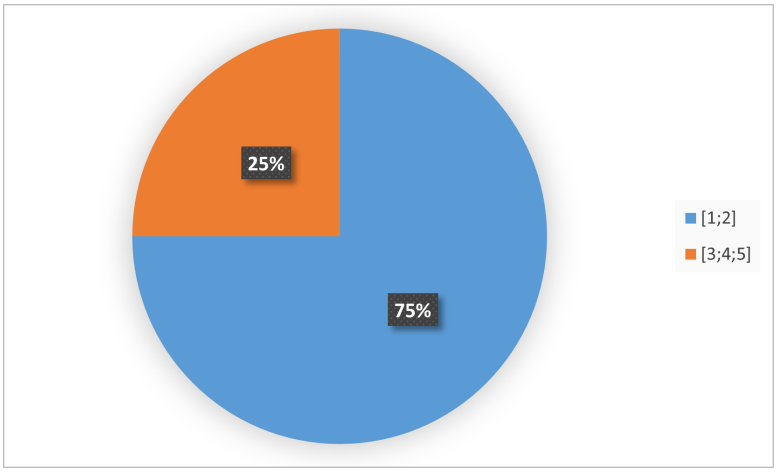


Figura 5.5: Dificuldade na questão 3

4. **Trocar as posições do "gráfico 2"com o "gráfico 3"**

O primeiro objetivo desta questão é verificar se o ícone de movimentação dos widgets é acessível e se não gera conflito com os ícones ao seu redor. O segundo objetivo é perceber se a movimentação dos widgets é uma funcionalidade que agrada aos utilizadores nos moldes que está programada.

Apesar dos resultados serem muito semelhantes á questão anterior, como podemos verificar através dos gráficos do tempo médio de resposta 5.2 e da dificuldade média 5.3 a abordagem para a análise não é a mesma. Neste caso a pergunta aborda duas variáveis. O ícone para movimentar os widgtes e a funcionalidade de movimentação.

Portanto, o facto de os resultados serem muito semelhantes é muito positivo pelo facto de estarem em avaliação duas componentes ao invés de uma. Apenas um utilizador classificou esta ação num nível superior a 2, como pode ser observado ver na tabela 5.3.

Nível	**Número de respostas**
>2	15
<2	1

Tabela 5.3: Dificuldade na questão 4

5. **Colocar o valor das margens da grelha em 12**

   Entender se o utilizador acede e altera os valores das margens da grelha naturalmente. Entrar novamente nas definições da grelha e alterar a funcionalidade que controla o valor das margens entre gráficos. Alterando o valor de 10 (pré-definido) para 12.

   Esta questão visa expor uma debilidade da biblioteca Gridster2. A caixa que formata o valor das margens oferece duas opções. Uma em que o utilizador pode colocar diretamente o valor escrevendo-o, e outra em que o utilizador através das setas de auxílio incrementa ou diminui ao valor.

   A fragilidade, consiste em que, sempre que esta caixa é aberta fora da área da grelha apenas é permitido ao utilizador fazer um clique. Após efetuar um clique a opção desaparece. Ou seja, se esta opção estiver dentro da área da grelha não existe nenhum problema. Se esta opção for colocada fora da área da grelha como numa side bar, menu de definições ou em cima da grelha (como é o caso) apenas vai ser possível fazer um clique.

   Como a opção está colocada por cima da grelha e não dentro, o utilizador apenas consegue efetuar um clique. Portanto, se ele tentar carregar na área para escrever o número da margem, já efetua um clique, e de seguida a opção desaparece.

   Se carregar nas setas de auxílio, apenas consegue fazer alterações no valor de uma unidade. Carrega para aumentar ou diminuir ao valor e a opção desaparece. Se pretender fazer alterações superiores a uma unidade tem que estar sucessivamente a abrir a opção, fazer a alteração, abrir novamente, voltar a alterar o valor até que atinga o pretendido.

Está prevista que a nova versão do gridster2 já resolva este problema, mas á data, ele persiste.

Devido a este facto, os valores de dificuldade para responder são os mais elevados (5.3). Os utilizadores tiveram dificuldade para fazer esta formatação. Esta questão é a que obtém o maior número de respostas com valores superiores a 3 na dificuldade, como é possível constatar na tabela 5.4. O que é perfeitamente normal e compreensível.

Questão	**Respostas com valor superior a 3**
1	2
2	0
3	1
4	1
5	4
6	0
7	0
8	0
9	0
10	0
11	1
12	1
13	0
14	0
15	0

Tabela 5.4: Respostas com nível de dificuldade superior a 3

No entanto, os utilizadores referem que não é uma funcionalidade crucial para o bom funcionamento. Portanto, após os testes esta funcionalidade foi retirada e ficará em suspenso até que uma nova *release* resolva o problema.

6. **Adicionar o gráfico :**

   - **Selecionar HuntGroup como grupo de estatísticas**
   - **Selecionar a métrica "Chamadas não atendidas"**
   - **Clicar em adicionar**

   Esta questão tem como objetivo introduzir a funcionalidade de adicionar um novo widget. Está estruturada para propositadamente falhar. A intenção é que faça aparecer uma mensagem de erro.

   Por ser uma introdução a uma funcionalidade que o utilizador ainda não explorou o tempo médio de resposta é cerca de 50 segundos (figura 5.2).

   No entanto, os utilizadores facilmente compreendem o menu de adicionar e seguem aquilo que lhes é pedido. Tendo uma classificação média de dificuldade de 1,13 (muito baixa)

(figura 5.3) e com um desvio padrão de 0,33. Concluindo que todos os utilizadores aceitam o menu de adicionar.

7. **Fazer as correções necessárias para conseguir adicionar o gráfico anterior**

   Perceber se as mensagens de erro conseguem transmitir as correções que são necessárias fazer para adicionar um widget e se a forma como estão estruturadas e passadas aos utilizadores é positiva. O objetivo é fazer com que o utilizador não fique bloqueado. Sempre que um erro ocorra o utilizador é informado do erro, e como o pode ultrapassar.

   Todos os utilizadores facilmente seguiram a mensagem e todos concluíram com sucesso a tarefa. Apresentando um resultado de dificuldade de 1 e com um desvio padrão de 0. É possível concluir que é uma das funcionalidades mais bem recebidas pelos utilizadores.

8. **Quantas chamadas não atendidas (em valor exato) foram rejeitadas pela origem**

   A partir do momento que o gráfico é adicionado é necessário saber se efetivamente ele passa as informações ao utilizador. A pergunta faz com que o utilizador olhe para o gráfico em maior detalhe e dê um dado específico.

   De salientar que esta etapa não foi possível de ser concluída por todos os utilizadores. Um dos utilizadores não conseguiu finalizar a tarefa. Para que o gráfico carregue os dados necessária uma conexão VPN. No caso desse utilizador a ligação VPN estava em baixo e o tempo de espera para carregar os dados foi ultrapassado.

   Quanto aos restantes, conclui-se que os gráficos passam bem as informações visto que, o tempo de execução não supera os 10 segundos (figura 5.2) e a dificuldade média é de 1,20 (figura 5.3). Validando que a biblioteca ApexCharts produz gráficos com qualidade para os utilizadores fazerem as devidas leituras.

9. **Adicionar o gráfico :**

   - **Escreva o título como "Chamadas por causa de não atendimento"**
   - **Selecionar WaitingQueue como grupo de estatísticas**
   - **Selecionar a métrcia "Chamadas não atendidas"**
   - **Escolha o donut como tipo de gráfico**

   O objetivo é explorar o tipo de gráfico donut e as suas propriedades.

   Como os utilizadores já exploraram a funcionalidade de adicionar widgets assiste-se a uma diminuição no tempo de execução da tarefa. Enquanto que na pergunta 6 (primeiro contacto com o menu adicionar) foi obtido um tempo médio de cerca de 50 segundos, nesta pergunta o tempo médio obtido é de sensivelmente 35 segundos (figura 5.6).

   A funcionalidade é muito bem aceite pela generalidade dos utilizadores visto que, apresenta um nível de dificuldade média de 1,08 (figura 5.3) com um desvio padrão de 0,27.

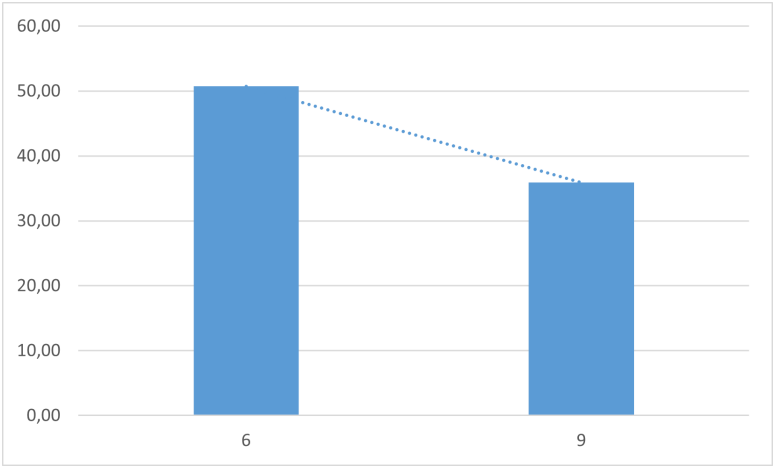


Figura 5.6: Tempo de execução da pergunta 6 e 9

10. **Quantas chamadas foram rejeitadas por fila cheia (em valor exato) ?**

    Fazer com que o utilizador explore o gráfico de donut em maior detalhe e dê um dado específico. O objetivo é provar que o gráfico de donut passa bem a informação.

    Como anteriormente o utilizador já explorou como se obtém um dado específico, visto que a biblioteca é a mesma (ApexChart), o princípio para obter dados em específico é o mesmo. Colocar o rato sobre a informação.

    Assim sendo, o tempo de execução é dos mais baixos - cerca de 10 segundos (figura 5.2) - bem como o nível de dificuldade - 1,13 (figura 5.3)

    De referir que novamente o utilizador que anteriormente não conseguiu carregar dados para o gráfico, voltou a não conseguir fazê-lo nesta questão.

11. **Desabilite a opção "push items"da grelha**

    Explorar uma funcionalidade de personalização da grelha. Entender se a opção está corretamente colocada nas definições e se a sua conceção é a mais adequada. Nesta fase o utilizador já conhece o menu de *settings* da grelha. O propósito é apurar se o posicionamento é correto visto que é a única opção que apresenta uma *check-box*.

    Os resultados são positivos visto que, o tempo da efetuação da questão e os níveis de dificuldade apresentados são baixos. Quinze segundos e 1,50 respetivamente (figura 5.2 e figura 5.3).

12. **Coloque o gráfico "Chamadas por causa de não atendimento"na canto superior esquerdo**

    Depois de desabilitar a opção, esta pergunta visa colocar a nova funcionalidade em prática. Tem como objetivo explorar se a opção de personalização de "push items"é necessária.

    Aqui é colocada uma dificuldade que o utilizador não está á espera. Depois de na questão 4 trocar posições entre gráficos, o utilizador está á espera que nesta pergunta o princípio de

movimentação de widgets seja o mesmo. No entanto a pergunta anterior desabilita a opção de *"push items"* pondo a movimentação de widgets a funcionar de forma diferente.

Dois utilizadores contornaram a realização da tarefa. Voltaram a habilitar a opção de *"push items"* e resolveram a questão. No entanto, como o pretendido é que os utilizadores contactem com a nova funcionalidade não foram registadas as respostas destes dois utilizadores.

Quanto aos restantes as respostas são positivas. Com uma dificuldade média de 1,64 (figura 5.3) e com um desvio padrão de apenas 0,97. Conclui-se que a generalidade dos utilizadores afirma ser fácil perceber a funcionalidade.

Quanto ao tempo de execução, se for comparada a questão 4 com a questão 12 conclui-se que são muito idênticos. Com cerca de 37 e 39 segundos respetivamente.

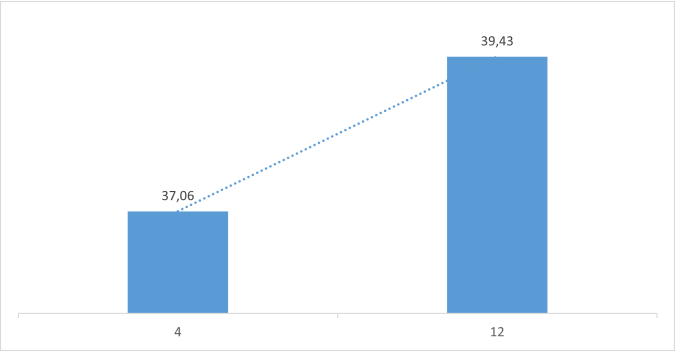


Figura 5.7: Tempo de execução da pergunta 4 e 12

Portanto, se os utilizadores estão á espera que o componente se comporte de uma forma, obtém uma resposta diferente e mesmo assim o tempo de execução é praticamente igual conclui-se que facilmente os utilizadores entendem a nova funcionalidade e adaptam-se a ela.

13. **Adicionar o gráfico :**

    - **Escreva o título como "Chamadas por tipo"**
    - **Selecionar SMS And Call como grupo de estatísticas**
    - **Selecionar a métrcia "Chamadas por tipo"**
    - **Escolha o gráfico de colunas**

    O objetivo é explorar as propriedades de gráfico de colunas.

    Nesta fase os utilizadores já adicionaram outros gráficos anteriormente por isso, já obtiverem conhecimento sobre esta funcionalidade. Daí que a dificuldade tenha um valor de 1,45 (figura 5.3) com um desvio padrão de 0,78.

    Esta foi a pergunta que mais utilizadores falharam na concretização. Este facto foi devido á conexão VPN instável. Dos 16 participantes, 5 deles não conseguiram finalizar a questão. Na tabela 5.5 é possível consultar a taxa de concretização por pergunta.

Questão	Taxa de concretização
1	100%
2	100%
3	100%
4	100%
5	100%
6	100%
7	88%
8	94%
9	88%
10	88%
11	100%
12	94%
13	69%
14	94%
15	100%

Tabela 5.5: Taxa de concretização por pergunta

14. **Faça download em SVG do gráfico adicionado anteriormente**

    Pela primeira vez pede-se ao utilizador que efetue download de informação. O objetivo é verificar se a opção de download está acessível e que o utilizador facilmente chega às várias opções de download. Neste caso em específico era pedido que efetua-se em SVG.

    De notar que os participantes que na questão anterior não conseguiram concluir a tarefa, têm a opção de download acessível na mesma. No entanto, quando descarregam a informação, o ficheiro estará vazio. O objetivo é entender se o posicionamento para efetuar download está bem colocado e se o utilizador o percebe, portanto, as respostas dos utilizadores que não concluíram a tarefa anterior com sucesso foram na mesma consideradas.

    No entanto, um dos utilizadores não conseguiu concluir a tarefa porque durante esta questão o rato deixou de funcionar. Não permitindo que conseguisse clicar na opção.

    Os resultados são confirmativos que os utilizadores acatam bem a funcionalidade. Foram rápidos a descobrir onde se encontra a funcionalidade e não tiverem nenhuma dificuldade na execução da questão. Apresentando uma média de resposta de 5,56 (tabela 5.1) segundos, apresentando um dos valores mais baixos de desvio padrão de apenas 0,96 (tabela 5.1). A dificuldade que os utilizadores atribuíram é de 1,33. Um valor baixo, que permite concluir que os utilizadores apreciam a funcionalidade.

15. **Efetue um *log off* da plataforma**

    Verificar se quando o utilizador quer sair da plataforma, o pode fazer rápido e de forma segura.

Os resultados desta questão são claros. Apresenta o valor de execução mais baixo de todas as questões com apenas 3,38 segundos (figura 5.2) de execução. O tempo mais elevado de execução nesta pergunta foi de 8 segundos. Apresentando assim um valor de desvio padrão também baixo de 1,62 (tabela 5.1). Quanto á dificuldade nenhum utilizador teve a mínima dificuldade em concluir a tarefa. Obtendo-se assim o valor mais baixo de dificuldade com o valor de 1 (figura 5.3) e desvio padrão de 0.

5.2.2 Avaliação da despistagem

A avaliação deste questionário tem de ser feita questão a questão. A escala é feita de 1 a 5 sendo o 1 "Discordo Totalmente"e 5 "Concordo Totalmente". As questões foram feitas de forma que os utilizadores não respondessem ou tudo 1 ou 5. Obrigando-os a estar concentrados e responder atentamente.

A tabela 5.6 mostra a resposta esperada para cada questão, a resposta obtida e o desvio padrão por questão. Nenhuma das questões ultrapassou um valor superior a 2 no desvio padrão. Sendo possível concluir que houve um comportamento generalizado ao responder.

Questão	Resposta esperada	Obtida	Desvio padrão
1	[4;5]	4,50	0,87
2	[4;5]	4,38	0,60
3	[4;5]	4,13	1,54
4	[4;5]	4,44	1,32
5	[1;2]	1,50	1,06
6	[1;2]	1,81	1,42
7	[4;5]	4,31	1,31
8	[1;2]	2,13	1,69
9	[1;2]	1,94	1,30
10	[1]	1,00	0,00
11	[1]	1,00	0,00
12	[1]	1,31	0,77

Tabela 5.6: Resposta esperada, resposta obtida e desvio padrão por questão de despistagem

1. **O trabalho que está a realizar está “on schedule”**

   Perceber se o teste é feito sem o *stress* de ter trabalho atrasado. Para esta questão o que se pretende é que responda "Concordo Totalmente"(5). Confirmando que está tudo dentro dos prazos e que se encontrava tranquilo para dispensar tempo para efetuar o teste de usabilidade.

   Os resultados obtidos são dentro do expectável. A media de resposta foi de 4,5 e 75% dos utilizadores responderem com 5 (concordo totalmente) á questão (figura 5.8). Todas as respostas tiverem valores de 5 ou 3.

Conclui-se que os utilizadores efetuaram o teste concentrados e sem a preocupação de ter o trabalho fora de controlo.

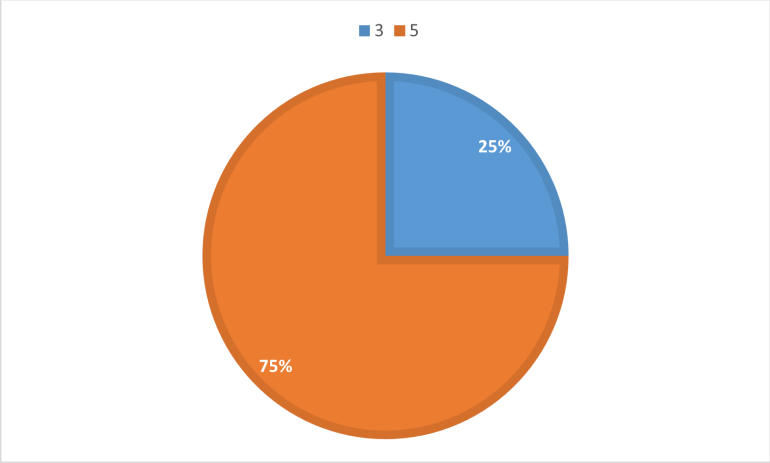


Figura 5.8: Frequência de respostas á questão 1 de despistagem

2. **A carga de trabalho que tem é apropriada**

   Apurar se o utilizador não está massacrado com o trabalho que tem para ou por realizar. O pretendido é perceber se os utilizadores não estão demasiado focados no trabalho.

   As respostas são esclarecedoras. Obtendo apenas uma resposta fora do expectável que é de 4 ou 5. A resposta obtida fora do expectável tem uma particularidade. O utilizador atribuiu 3 á questão, mas referiu que a classificação que atribuiu não é por ter trabalho á mais mas sim, a menos. Obtendo assim, um desvio padrão de baixo com o valor de 0,60 (tabela 5.6)

   Mais uma vez, as respostas obtidas estão dentro do expectável como se pode ver na tabela 5.7.

Classificação	Número de respostas
1	0
2	0
3	1
4	8
5	7

Tabela 5.7: Número de respostas á questão 2 de despistagem

3. **A internet funcionou corretamente durante a realização do teste de usabilidade**

   Verificar se as condições de conexão eram apropriadas e se não ouve dificuldades de comunicação. Ex: cortes na videochamada, *frozen* do ecrã etc.

   Em três dos testes realizados, verificaram-se algumas dificuldades técnicas relacionadas com a conexão internet. Nestes três casos a qualidade da videochamada foi afetada. No entanto,

não impeditivas que os utilizadores não conseguissem ouvir as questões, concluí-las e ser feito o registo.

Obteve-se uma média de 4,1 o que é satisfatório. As respostas alternaram entre o 4,5 e 1. Daí que os valores de desvio padrão desta questão sejam elevados (tabela 5.6).

Concluindo, apesar de em 3 casos a qualidade da videochamada não ter sido a melhor as dificuldades foram ultrapassadas.

4. **A VPN funcionou corretamente durante a realização do teste**

   Averiguar se a VPN funcionou corretamente durante a realização do teste. Só com a VPN ativa é que é possível carregar dados para os gráficos. Visa perceber se após a adicionar gráficos, ou até no dashboard inicial, os dados não excedem o tempo expectável de espera no carregamento dos dados. A resposta esperada é o "Concordo Totalmente"

   Em dois testes a conexão VPN dos utilizadores não funcionou. O que impossibilitou que as questões em que era necessário carregar dados para os gráficos não fossem efetuadas. Nestes casos, as questões não foram concluídas e não foram efetuados os seus registos.

   Obteve-se uma média de 4,4 de resposta, um valor muito satisfatório. Verificando apenas, dois valores fora do expectável 4 e 5 como se pode ver na tabela 5.8. De destacar que em 13 casos a VPN funcionou impecavelmente.

Classificação	Número de respostas
1	2
2	0
3	0
4	1
5	13

Tabela 5.8: Respostas á questão 4 de despistagem

5. **O programa teve alguma dificuldade de funcionamento (ex: demorou a carregar os dados, demorou a abrir o menu para adicionar gráficos)**

   Aferir se o programa funcionou sem qualquer problema. Ex: Encravar, ser necessário fazer refresh etc. Neste caso o utilizador deve discordar da afirmação. Atribuindo a classificação de 1.

   Obteve-se um resultado bastante esclarecedor com uma média de 1,5. Concluindo que durante a realização do teste o programa comportou-se como esperado e não apresentou deficiências. Apresentando uma eficácia de 81% como se pode verificar na figura 5.9.

6. **Durante o teste de usabilidade teve algum fator externo a ocorrer (ex: telefone tocar, mensagem no skype)**

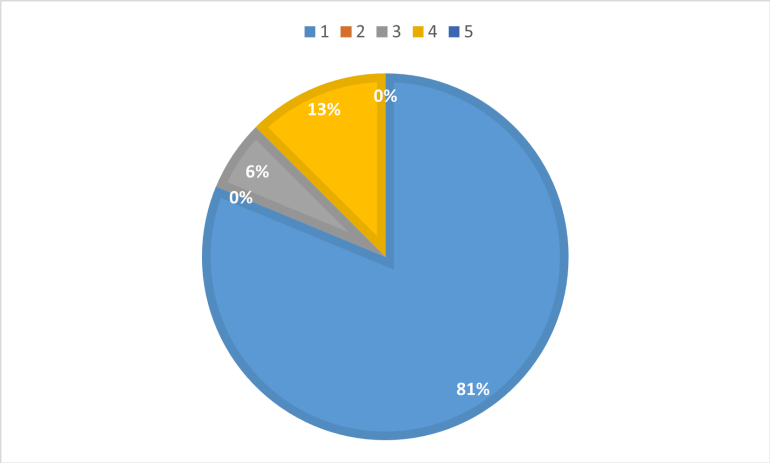


Figura 5.9: Percentagem de respostas á questão 5 de despistagem

Analisar se as respostas do utilizador estão influenciadas. Ex: se no decorrer de uma pergunta o utilizador responder a uma mensagem no skype o tempo de resposta está influenciado. A resposta esperada será entre o 1 e 2 (Discordo) visto que o utilizador deveria estar focado no teste sem perturbações.

Dos 16 utilizadores que realizaram o teste, 11 deles não tiverem qualquer perturbação e 2 deles tiveram perturbações mínimas que não afetaram o tempo de resposta. Quanto aos 3 restantes, sofreram com distrações que afetou o tempo de execução (tabela 5.9).

Classificação	**Número de respostas**
1	11
2	2
3	0
4	1
5	5

Tabela 5.9: Respostas á questão 6 de despistagem

Visto que o teste foi realizado á distância não foi possível perceber quando é que as perturbações ocorriam e descontar esse tempo no tempo de execução. Apenas era partilhado o ecrã do utilizador, portanto quando este parava para por exemplo, responder a uma mensagem no telefone, era impossível verificar tal situação.

Num regime presencial seria fácil perceber estes acontecimentos e não contabilizar o tempo de influência.

No entanto, como apenas 3 dos utilizadores afirmaram ter tido este tipo de perturbações, e estas poderem ter acontecido em qualquer momento do teste, pode-se acreditar na firmeza dos tempos de resposta dos restantes utilizadores.

Os utilizadores que afirmaram ter sofridos estas perturbações são: User06, User09 e User15. A tabela 5.10 apresenta os tempos deste utilizadores em comparação com a média.

Questão	**Média**	**User 6**	**User 9**	**User 15**
1	67,00	45	7	120
2	13,13	6	27	96
3	32,13	15	22	27
4	37,06	30	70	55
5	31,50	60	80	40
6	50,75	45	130	50
7	27,67	29	26	26
8	8,13	4	8	5
9	35,92	147	37	4
10	10,75	10	4	8
11	14,81	37	10	5
12	39,43	15	12	80
13	25,18	122	12	10
14	5,56	5	5	5
15	3,38	3	3	5
Total	402,39	573	453	536

Tabela 5.10: Tempos de resposta dos utilizadores com influências externas

As influências externas aumentam o tempo de resposta. Portanto, todas as respostas em que estes utilizadores tenham respondido abaixo da média levam a crer que nessa questão não foram perturbados.

Posto isto, isolou-se cada utilizador e filtraram-se as questões em que estes ultrapassaram o tempo médio da resposta significativamente. Obtendo-se assim as seguintes tabelas 5.11,5.12 e 5.13.

Questão	**Média**	**User 6**
5	31,50	60
9	35,92	147
11	14,81	37
13	25,18	122

Tabela 5.11: Tempos de resposta do utilizador 6 que ultrapassam o tempo médio

Sendo impossível de calcular o tempo que as influências tiveram no tempo de resposta, o mais viável de ser fazer é anular estas respostas e verificar o efeito.

As questões que são suspeitas de influência, que sejam comuns entre os utilizadores mencionados serão agregadas. Por exemplo, a questão número 2 é comum ao utilizador 9 e 15. Por isso a análise dos efeitos vai ser efetuado anulando os dois registos e não separadamente.

Fazendo esta filtragem obtemos os resultados presentes na tabela 5.14.

Questão	Média	User 9
2	13,13	27
4	37,06	70
5	31,50	80
6	50,75	130

Tabela 5.12: Tempos de resposta do utilizador 9 que ultrapassam o tempo médio

Naturalmente após a filtragem verifica-se que a média desce. Para fazer uma análise mais eficaz seria descontar o tempo das influências no tempo total da resposta. Como tal não é possível devido á situação pandémica, fica assim registada a análise possível filtrando as influências externas

7. **O ambiente em que realizou o teste era apropriado (como: condições de temperatura, cadeira confortável, claridade etc.)**

   Visto ser um componente em que a parte visual é importante, é necessário averiguar se as condições de realização do teste são apropriadas. Ex: Se um utilizador realizar o teste ao ar livre, a claridade vai interferir na visualização dos componentes. A resposta pretendida é que o utilizador concorde com a afirmação (atribuindo 4 ou 5)

   Apenas dois utilizadores, afirmam que as condições em que realizou o teste não são as melhores. Em ambos os casos, a resposta teve que ver com o facto de estarem a utilizar um PC temporário com uma dimensão de ecrã mais reduzida do que o habitual.

   Como se pode ver na figura 5.10 conclui-se que em 87% dos casos os utilizadores realizaram os testes em condições adequadas.

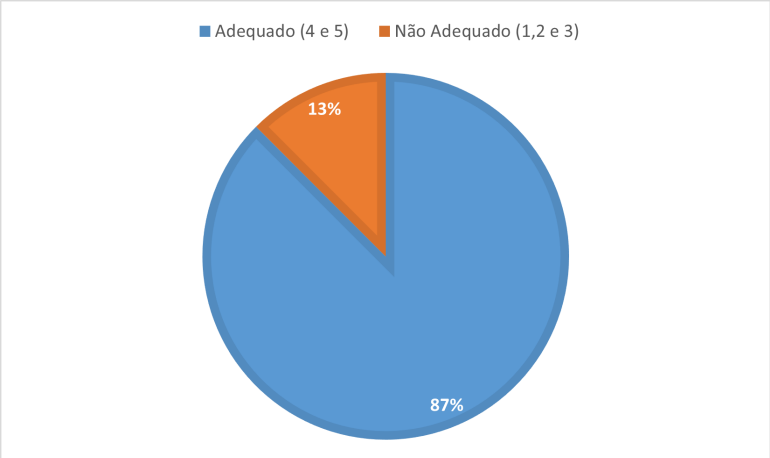


Figura 5.10: Condições de realização do teste

8. **Nos minutos antecedentes á realização do teste, saiu de alguma reunião**

   Apurar se o utilizador chega ao teste concentrado para o efeito. Era crucial que todos os utilizadores chegassem a horas para que todos os testes fossem efetuados nas horas agendadas.

Questão	Média	User 15
1	67,00	120
2	13,13	96
4	37,06	55
12	39,43	80

Tabela 5.13: Tempos de resposta do utilizador 15 que ultrapassam o tempo médio

Quando um teste se prolonga para lá da hora, o efeito bola de neve sentir-se-ia na agenda. O que poderia provocar cancelamentos. Felizmente isso não aconteceu e tudo foi cumprido dentro dos prazos.

Os 5 utilizadores que responderam com valores superiores a 3, cumpriram o horário respeitando o agendamento. Não afetando os resultados do teste.

9. **Nos próximos minutos tem alguma reunião agendada**

   Perceber se teve de responder apressadamente para entrar noutra reunião atempadamente.

   Três utilizadores responderam com um valor superior a 3. O que significa que tinham reuniões agendadas após a realização do teste. Os utilizadores nº8,14 e 15 (que responderam com valores superiores a 3) pelo facto de terem uma agenda apertada nos próximos minutos influenciam o tempo das repostas ao teste de usabilidade negativamente em algumas das questões. A tabela 5.15 apresenta a média dos tempos por resposta e os tempos dos utilizadores 8,14 e 15.

   Nos casos dos utilizadores nº8 e 14 nota-se efetivamente que foram mais rápidos que a média. No caso do utilizador nº15 ultrapassou o tempo médio.

   Nos casos em que as suas respostas são efetuadas rapidamente, é positivo porque significa que os componentes estão facilmente acessíveis e é fácil trabalhar com eles. No entanto, os tempos destes utilizadores estão influenciados por terem de ser rápidos para cumprir a agenda.

   Os resultados destes utilizadores não podem ser descartados. O facto de em grande parte estarem abaixo da média e responderem rapidamente é positivo. No entanto, é necessário ter presente que os resultados têm um fator de influência por detrás. Contudo pode retirar-se deste facto que, quando utilizadores tiverem a necessidade de ser mais rápidos o componente consegue dar essa resposta.

10. **O seu contrato está perto de terminar**

    Aferir o grau de comprometimento ao teste.

    Esta questão tem resultados claros. Todas as respostas foram 1, o que significa que todos os utilizadores estão comprometidos com a empresa e com o seu sucesso. Apresentando um desvio padrão de 0, como se pode ver na tabela 5.6.

Questão	Média	Desvio Padrão	Média após filtragem	Desvio Padrão após filtragem
1	67,00	52,94	63,47	52,82
2	13,13	22,65	5,44	5,84
4	37,06	32,81	29,25	33,16
5	31,50	18,30	27,75	18,22
6	50,75	39,80	42,63	35,87
9	35,92	37,08	24,62	19,94
11	14,81	15,44	12,50	14,69
12	39,43	35,82	33,71	35,21
13	25,18	31,74	14,09	9,49

Tabela 5.14: Dados removendo as influências do User6, User9 e User15

Pode-se concluir que os utilizadores estão comprometidos com o bom desenvolvimento desta solução.

11. **Durante a realização do Guião de utilização sentiu-se intimidado**

    Averiguar se as respostas foram executadas de forma natural ou tiveram uma influência negativa de intimidação pelo facto de o teste estar a ser visualizado.

    Á semelhança da pergunta anterior, os resultados são claros. Nenhum utilizador se sentiu intimidado e todas as respostas foram "Discordo totalmente". (tabela 5.6). Havendo assim, um clima positivo para a realização do teste o que vai de encontro ao ambiente normal em que a ferramenta será utilizada.

12. **Sentiu alguma pressão para terminar as tarefas o mais rápido possível**

    Pelo facto de o teste estar a ser visualizado, pretende-se perceber se o utilizador respondeu de uma forma espontânea. O facto de por vezes tentar ser rápido demais pode atrapalhar e ter o efeito contrário.

    Apenas um utilizador sentiu alguma pressão. Referiu que esta pressão é posta por ele mesmo para ser bem-sucedido nas tarefas. Quantos aos restantes, discordaram da afirmação. Apresentando um desvio padrão baixo com apenas 0,77 de valor (figura 5.6). O que indicam que resolveram as questões de forma natural.

5.3 Sumário

Na globalidade a realização dos testes correram bem. Os ajustes que foram necessários fazer face áquilo que estava inicialmente planeado foram eficazes e permitiram fazer uma boa avaliação. As novas ferramentas que têm aparecido e ganho relevância com a pandemia, como *Microsoft Teams*® e outros, fizerem com que as realizações dos testes se tornassem mais fáceis. A acrescentar a isto, aquando da realização do teste, já todos os intervenientes tinham contactado e explorado a ferramentas *Microsoft Teams*®. O que facilitou as típicas questões de: "Não sei onde se liga o

Questão	Média	User 8	User 14	User 15
1	67,00	56	95	120
2	13,13	2	5	96
3	32,13	7	12	27
4	37,06	20	85	55
5	31,50	19	45	40
6	50,75	45	25	50
7	27,67	20	x	26
8	8,13	3	x	5
9	35,92	58	x	4
10	10,75	3	x	8
11	14,81	8	4	5
12	39,43	60	20	80
13	25,18	x	x	10
14	5,56	5	6	5
15	3,38	2	2	5
Total	402,39	308	299	536

Tabela 5.15: Tempos de resposta dos utilizadores com agenda apertada

microfone"ou "Nunca fiz partilha de ecrã". Estas questões aparentemente irrelevantes, são aquelas que permitiram que todos os testes fossem realizados dentro dos prazos previamente estipulados. O ambiente em todas as sessões foi bastante positivo e criou-se uma boa atmosfera para que os utilizadores não vissem o teste como um aborrecimento.

Durante a realização dos testes de usabilidade, foi necessário haver algum rigor para que os participantes não se distraíssem do foco. Realizar as tarefas pedidas no teste. A tendência para quem realizava os testes era que, assim que terminavam uma tarefa tecer comentários. Os comentários e o feedback é bom e necessário mas o crucial era terminar o teste dentro do prazo. Depois de finalizar, aí sim poderiam comentar e sugerir aquilo que achassem necessário. Todo o feedback foi registado e organizado.

Na parte da despistagem os utilizadores estiveram muito confortáveis visto que, as questões eram de fácil compreensão. Enquanto, nos testes de usabilidade recebiam uma questão e tinham de explorar o componente para realizarem a pergunta, na despistagem respondiam de imediato e não houve necessidade de uma segunda explicação da pergunta para nenhum dos utilizadores. A despistagem foi importante para perceber que resultados estão ou não influenciados. O regime em que os testes foram efetuados faz com que as conclusões das influências seja mais difícil de obter.Num regime presencial uma ida á casa de banho durante a realização do teste faria o cronómetro parar. Num regime remoto tona-se mais complicado controlar estes aspetos. Apesar de os utilizadores terem sido chamados a atenção no início do teste, foi notória que ocorreram algumas interferências.

A adaptação ao contexto da pandemia não influenciou a avaliação. Tudo aquilo que seria expectável de ser avaliado assim o permaneceu, apenas mudou o método para chegar a esse fim.

Capítulo 6

Conclusões

Depois do problema apresentado estudou-se a melhor forma para o resolver. A solução mais adequada ao contexto, foi o desenvolvimento de um dashboard que permitisse ao utilizador manipulá-lo para o colocar a informação numa disposição adequada. Toda a informação que entra no dashboard é controlada pelo utilizador.

Efetuaram-se os estudos da tecnologia que melhor se adequa ao contexto das telecomunicações em grande escala. Dentro das tecnologias percebeu-se os seus mecanismos de funcionamento e como dão resposta a alguns dos problemas identificados. Selecionou-se a tecnologia Angular e foram procuradas bibliotecas que já tivessem resolvido alguns problemas identificados de forma a aproveitar trabalho já desenvolvido. Após a construção de um *product backlog* elaborou-se um plano de desenvolvimento seguindo uma metodologia SCRUM.

O componente desenvolvido obteve uma classificação média de 1,33 numa escala de 1 a 5. É possível concluir que é muito positivo e que não oferece qualquer dificuldade de utilização. Revelando-se uma ferramenta de trabalho útil e que ajudará no bom desenrolar do mesmo, contribuindo com a fluidez que apresentou.

Os 402 segundos (cerca de 7 minutos) que os utilizadores demoraram para percorrer cada uma das funcionalidade e explorar em detalhe os gráficos é bom indicador. Se um utilizador precisa de 7 minutos para realizar tarefas que lhe eram ditadas de um componente que desconhecia á partida, em situações de trabalho e já com algum conhecimento adquirido do componente conseguirá fluidamente trabalhar com o componente e aceder a todas as funcionalidades.

Na globalidade todos os indicadores são positivos para este novo componente que nasce e que se espera que permaneça funcional e útil durante toda a sua vida.

6.1 Contribuições

A funcionalidade oferecida anteriormente para consulta de dados era inadequada. Este novo componente revela ser *user-friendly*, funcional e resolve o anterior problema.

A plataforma ABC fica assim melhor preparada para responder ás necessidades dos clientes. O componente acrescenta valor á plataforma. Equipando os clientes com uma ferramenta mais poderosa para as ações de gestão e monitorização de um negócio.

Com o desenvolvimento deste componente com tecnologias mais atuais face às anteriormente utilizadas pela plataforma ABC, abre-se a porta para que a parte *front-end* de toda a plataforma passe por uma fase de reestruturação tecnológica. Aproveitanto o trabalho desenvolvido neste componente é possível expandir a utilização das tecnologias aqui utilizadas ao resto da plataforma.

6.2 Trabalho futuro

Está planeado continuar o desenvolvimento do Altycs. Uma ferramenta que mereceu elogios dos utilizadores e que querem ver ainda mais desenvolvida. Como tal, durante a realização dos inquéritos, foram recolhidas as várias sugestões e organizadas de forma que a plataforma satisfaça as suas necessidades.

As sugestões foram organizadas tendo em conta três componentes da plataforma e contabilizadas o número de vezes que foram sugeridas por utilizador (x/16):

1. **Relativas ao *widget***

    - Confirmação da ação de apagar **(3/16 – 19%)**
    - Ao adicionar, ter um tamanho padrão que permita visualizar o widget **(4/16 – 25%)**
    - Modo de ajuda a daltónicos **(1/16 – 6%)**
    - As checkBox do statisctic group a vermelho com fundo preto não é intuitivo (menu adicionar) **(1/16 – 6%)**
    - Colocar uma cor mais carregada ou outra cor quando um widget type é selecionado (menu adicionar) **(2/16 – 13%)**
    - Ser possível recuperar o widget apagado (por breves segundos) **(1/16 – 6%)**
    - Pop-up com duplo clique nos gráficos para abrir configuração/edição **(1/16 – 6%)**
    - Alterar o logotipo para fazer download do gráfico **(2/16 – 13%)**
    - Não saber onde o widget que adicionamos vai ficar colocado **(1/16 – 6%)**
    - Fechar o menu de adicionar gráfico carregando noutra zona da plataforma **(3/16 – 19%)**
    - Icon de lixo diferente **(1/16 – 6%)**
    - Refresh de dados no próprio widget **(1/16 – 6%)**
    - Editar o widget no próprio widget (ex: mudar o título) **(1/16 – 6%)**

2. **Relativas á grelha**

    - O nome push items não dá a entender a ação **(4/16 – 25%)**

- A posição do push items não é melhor **(2/16 – 13%)**
- Acrescentar as unidades nas settings (ex: pixéis, percentagem etc.) **(1/16 – 6%)**
- As settings da grelha deseparecem **(5/16 – 31%)**
- As opções da grid maiores **(1/16 – 6%)**

3. **Relativa às definições da grelha**

   - Modo de visualizar e editar **(2/16 – 13%)**
   - Ser possível jutar/organizar os gráficos através de um clique num botão **(1/16 – 6%)**
   - Arrastar os gráficos a partir de qualquer area de clique deixando o gráfico pressionado **(2/16 – 13%)**
   - Barra horizontal mais facilmente acessível sem ter de fazer scroll down **(2/16 – 13%)**
   - Ser possível adicionar texto livre **(1/16 – 6%)**

A sugestão que obteve maior número de frequência foi:

- As settings da grelha deseparecem **(5/16 – 31%)** [Relativa às definições da grelha].

Como foi analisado no capítulo 5 esta limitação será resolvida pela biblioteca *Gridster2* e assim que surgirem novos desenvolvimentos esta limitação será prontamente resolvida.

Quanto ás restantes sugestões, as que obtiverem mais do que duas sugestões foram abordadas, estudadas e concluídas ainda antes da conclusão desta tese. São elas:

- **4/16 – 25%**
  - Ao adicionar, ter um tamanho padrão que permita visualizar o widget [Relativo ao *widget*]
  - O nome push items não dá a entender a ação [Relativa às definições da grelha]
- **3/16 – 19%**
  - Confirmação da ação de apagar [Relativo ao *widget*]
  - Fechar o menu de adicionar gráfico carregando noutra zona da plataforma [Relativo ao *widget*]

As restantes sugestões foram anotadas e vão são ser abordadas num período á posteriori. Para trabalho futuro ficaram delineados os seguintes pontos, que visam uniformizar processos de chamada de dados e acrescentar opções de personalização :

- Desenvolvimento de todas as APIs de forma a generalizar o processo de chamada dos dados.
- Acrescentar mais opções de gráficos para adicionar como, gráficos de linhas, area, *timeline*, *heatmap*, *treemap* e *polar area* (Todas estas opções são suportadas pela biblioteca ApexCharts).

- Ao adicionar os gráficos permitir fazer *drag and drop* para posicionar os widgets na posição que desejar logo á partida.
- Acrescentar a filtragem temporal aos dados. Por exemplo, visualizar os dados dos últimos 7 dias, 15 dias, 1 mês etc.
- Oferecer mais opções de personalização do gráfico. Como: permitir escolher as cores que quer colocar nos gráficos e escrever nas legendas.
- Formatar o widget. Ter um icon de definições junto aos de movimentação e de apagar, que permita mudar o título, mudar o tipo de gráfico e mudar a filtragem temporal.

Referências

[1] Booth P. A. *An Introduction to Human-Computer Interaction*. Lawrence Erlbaum Associates, 1st edition edição, 1989.

[2] Kuhail M. A. A. e Lauesen S. Uvis: A formula-based end-user tool for data visualization. *IEEE Access*, páginas 110264–110278, 2020.

[3] Mitchell A., Ruml W., Spaniol F., Hoffmann J., e M Petrik. Real-time planning as decision-making under uncertainty. Em *AAAI*, páginas 2338–2345, 2019.

[4] Smyth B., Phelan O., e McCarthy K. U.s. patent no. 8,412,796. Washington, DC: U.S. Patent and Trademark Office, 2013.

[5] A. Dix, J. Finlay, Abowd D. G., e Beale R. *Human Computer Interaction*. Pearson Education Limited, 3rd ed. edição, 2004.

[6] Barbosa S. D. J. e Silva B. S. Interação humano- computador, 2010.

[7] Lazar J. e Feng J. H.and Hochheiser H. *Research Methods in Human-Computer Interaction*. Morgan Kaufmann Publishers, 2nd ed edição, 2017.

[8] Hikmat Ullah Khan, Shumaila Nasir, Kishwar Nasim, Danial Shabbir, e Ahsan Mahmood. Twitter trends: A ranking algorithm analysis on real time data. *Expert Systems with Applications*, página 113990, 2021.

[9] Losev V. V. Kovalev I. V., Saramud M. V. Simulation environment for the choice of the decision making algorithm in multi-version real-time system. Disponível em https://doi.org/10.3390/s20164557, 2020.

[10] John M. *Of keyboards and beyond-optimization in human-computer interaction*. Tese de doutoramento, Universität des Saarlandes, 2019.

[11] S Malik. *Enterprise Dashboards: design and best practices for it*. John Wiley & Sons, 2005.

[12] Elisabetta Mori e Shreeharsh Kelkar. Introduction to the special issue on interface architects: The evolution of human–computer interaction. *IEEE Annals of the History of Computing*, páginas 6–7, 2020.

[13] Castelli N. *Designing human-centered systems for the Internet of Things*. Tese de doutoramento, Universität Siegen, 2020.

[14] Kumar N. Call for special issue papers: Internet of things data visualization for business intelligence. *Big data*, páginas 452–453, 2020.

[15] Júnior G. A. O., Albuquerque R. O., Andrade C. A. B., Sousa R. T., Orozco A. L. S., e Villalba L. J. G. Anonymous real-time analytics monitoring solution for decision making supported by sentiment analysis. *Sensors*, páginas 20(16), 4557, 2020.

[16] Elizabeth R. Petrick. A historiography of human–computer interaction. *IEEE Ann. Hist. Comput.*, 42(4):8–23, outubro 2020.

[17] Hartono Pitoyo. Mixing autoencoder with classifier: Conceptual data visualization. *IEEE Access*, páginas 105301–105310, 2020.

[18] Jadhav A. R., S. K. MPR, e R. Pachamuthu. Development of a novel iot-enabled power-monitoring architecture with real-time data visualization for use in domestic and industrial scenarios. *Transactions on Instrumentation and Measurement*, páginas 1–14, 2020.

[19] Matheus R., Janssen M., e Maheshwari D. *Data science empowering the public: Data-driven dashboards for transparent and accountable decision-making in smart cities*. Elsevier, 2020.

[20] Lopes M. A. S., Neto A. D. D., e A. M. Martins. Parallel t-sne applied to data visualization in smart cities. *IEEE Access*, páginas 11482–11490, 2020.

[21] Gautam Shroff, Puneet Agarwal, e Lipika Dey. Enterprise information fusion for real-time business intelligence. páginas 1 – 8, 08 2011.

[22] Erotokritos Skordilis e Ramin Moghaddass. A deep reinforcement learning approach for real-time sensor-driven decision making and predictive analytics. *Computers Industrial Engineering*, 147:106600, 2020.

[23] Eckerson W. W. *Performance dashboards: measuring, monitoring and managing your business*. John Wiley Sons, 2end ed. edição, 2011.

[24] K. Wang, H. Bi, Y. Zhang, C. Zhang, Z. Liu, e S. Zheng. D2C-Net: A Dual-branch, Dual-guidance and Cross-refine Network for Camouflaged Object Detection. *IEEE Transactions on Industrial Electronics*, 2021.

[25] Mina Yoo e Sung-Hee Jin. Development and evaluation of learning analytics dashboards to support online discussion activities. *Educational Technology Society*, 23(2):1–18, 2020.

[26] Gareth W. Young e Rob Kitchin. Creating design guidelines for building city dashboards from a user's perspectives. *International Journal of Human-Computer Studies*, 140:102429, 2020.

[27] Yuliang Yun, Dexin Ma, e Meihong Yang. Human–computer interaction-based decision support system with applications in data mining. *Future Generation Computer Systems*, 114:285–289, 2021.