



UNIVERSIDADE
LUSÓFONA

Ubiquitous Quiz Builder

Trabalho Final de curso

Relatório Final

Diogo Novo

Orientador: Miguel Tavares | Co-Orientador: Bruno Cipriano

Trabalho Final de Curso | LEI | 10/09/2021

www.ulusofona.pt

Direitos de cópia

Ubiquous Quiz Builder, Copyright de Diogo Novo, ULHT.

A Escola de Comunicação, Arquitectura, Artes e Tecnologias da Informação (ECATI) e a Universidade Lusófona de Humanidades e Tecnologias (ULHT) têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objectivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.

Resumo

Com este Trabalho Final de Curso (TFC) pretendeu-se dar continuidade ao projeto *Android Quiz Builder v2*. Este projeto consiste numa plataforma onde as pessoas podem responder a questionários que poderão ser configurados como um jogo. Estes questionários são respondidos pelos utilizadores na aplicação *Android*.

O projeto original também tem um *website* que serve simultaneamente como suporte, ou melhor dizendo, *backoffice*. Este permite aos administradores gerir toda a informação, incluindo, a gestão dos questionários, como também dos utilizadores e ainda disponibiliza a consulta de estatísticas, acerca dos questionários respondidos na aplicação móvel.

Porém, em edições anteriores deste projeto foi desenvolvida uma aplicação em *Android* Nativo, destinada unicamente para *Android*. Sendo assim, o *Ubiquous Quiz Builder*, teve como objetivo principal, desenvolver uma nova aplicação multiplataforma (*Android* e *iOS*) que disponibilize, todas e se possível, mais algumas funcionalidades do que as já existentes, entre elas, a possibilidade de haver perguntas com imagens. Isto tudo por meio da tecnologia Flutter[1].

Abstract

This Final Year Project intended to continue the Android Quiz Builder v2 project. This project consists of a platform where people can answer questionnaires that can be configured as a game. These quizzes are answered by users through the Android application.

The original project also has a website that serves simultaneously as support, or rather backoffice. This allows you to manage the application data, including all the management of the questionnaires, as well as the users and also provides the consultation of statistics about the questionnaires answered in the mobile application.

But in the previous project, as the name says, the mobile application was developed only for Android. Thus, Ubiquous Quiz Builder's main goal, to develop a cross- platform application that provides, all and if possible, some more features than the existing ones. However, it will be able to run on the two most widely used mobile device operating systems, Android and iOS, through Flutter technology.

However, in previous editions of this project, an application was developed in Android Native, intended only for Android. Thus, the Ubiquous Quiz Builder had as its main objective, to develop a new multiplatform application (Android and iOS) that makes available, all and if possible, some more features than those already existing, among them, the possibility of having questions with images. This was accomplished through Flutter technology.

Índice

Resumo	iii
Abstract.....	iv
Índice	v
Índice de Figuras	vii
Lista de Tabelas	ix
1. Identificação do Problema.....	1
2. Levantamento e análise dos Requisitos.....	2
3. Viabilidade e Pertinência	6
4. Solução Desenvolvida	8
4.1. Tecnologia Utilizada - Flutter	9
4.2. Arquitetura do Sistema	10
4.3. Storyboard.....	11
4.4. Resultados dos Testes	12
5. Benchmarking	15
5.1. Tecnologias.....	15
5.2. Aplicações no mercado	18
5.2.1. Socrative (Teacher & Student) Apps.....	19
5.2.2. QuizUp	21
6. Método e Planeamento	22
6.1. Fases de desenvolvimento da aplicação	22
6.2. Plano de Testes e Validação	24
6.2.1. Tarefas para os utilizadores	24
6.3. Calendário.....	25
7. Resultados	26
7.1. Backoffice/Website Público	26

7.2. Aplicação Móvel.....	32
8. Conclusão	38
Bibliografia.....	39
Glossário.....	42

Índice de Figuras

Figura 1 - Arquitetura do sistema.....	10
Figura 2 - Storyboard da aplicação móvel.....	11
Figura 3 - Ecrã de <i>login</i> na fase intermédia.....	12
Figura 4 - Ecrã de <i>login</i> final.....	12
Figura 5 - Ecrã de resposta depois dos testes	13
Figura 6 - Ecrã de resposta antes dos testes	13
Figura 7 - Alerta ao abandonar o questionário	13
Figura 8 - Prós e contras de aplicações nativas e híbridas.....	15
Figura 9 - Principais <i>frameworks</i> escolhidas por software <i>developers</i>	16
Figura 10 - Gráfico que compara o número de perguntas feitas do Stack Overflow[22]17	
Figura 11 – Ecrã login	19
Figura 12 - Homepage	19
Figura 13 - Feedback de uma resposta incorreta	19
Figura 14 - Feedback de uma resposta correta.....	19
Figura 15 - Ecrã pergunta com imagem	19
Figura 16 - Ecrã dos resultados geral	20
Figura 17 - Ecrã dos resultados detalhado.....	20
Figura 18 - Ecrã da categoria de carros	21
Figura 19 - Ecrã de escolha da resposta	21
Figura 20 - Feedback interativo da opção escolhida	21
Figura 21 - Esquema de calendarização do desenvolvimento do TFC.....	25
Figura 22 - Formulário para criar questionário (nova versão).....	26
Figura 23 - Formulário para criar questionário (versão antiga).....	26
Figura 24 - Formulário de criação para o modo Questionário	27
Figura 25 - Formulário para inserir pergunta (nova versão)	28
Figura 26 - Formulário para inserir pergunta (versão antiga)	28
Figura 27 - Erro ao finalizar questionário (versão antiga)	29
Figura 28 - Erro ao finalizar questionário (nova versão)	29
Figura 29 - Ecrã de <i>Login</i>	33
Figura 30 - Ecrã de Registo.	33
Figura 31 - Ecrã do Perfil.	33
Figura 32 – <i>Homepage</i>	33

Figura 33 - Lista de questionários do modo Clássico.....	34
Figura 34 - Lista de questionários de todos os modos de jogo.....	34
Figura 35 - Lista de questionários do modo Questionário.....	34
Figura 36 - Filtros da lista de questionários.	34
Figura 37 - Ecrã com informações do questionário.....	35
Figura 38 - Ecrã de resposta no modo Questionário.	35
Figura 39 - Ecrã de resposta no modo de jogo com imagem.	35
Figura 40 - Ecrã de resposta no modo de jogo sem imagem.....	35
Figura 41 - Ecrã de resultado no modo de jogo.....	36
Figura 42 - Ecrã de resultado no modo questionário.....	36
Figura 43 - Alerta de novo recorde pessoal.	36
Figura 44 - Ecrã de <i>ranking</i> no modo Contrarrelógio.	37
Figura 45 - Ecrã de <i>ranking</i> no modo Clássico.	37

Lista de Tabelas

Tabela 1 - Requisitos funcionais	4
Tabela 2 - Requisitos não funcionais.....	4

1. Identificação do Problema

Nos nossos dias, cada vez mais tem vindo a aumentar a procura por aplicações móveis, que possibilitem às pessoas testar o seu conhecimento ou também para o seu próprio entretenimento. Por isso é muito importante que estas possam estar disponíveis em diversas plataformas e dispositivos para atingirem um maior alcance de pessoas.

O trabalho realizado nos anos passados, tal como o nome indicava, “*Android Quiz Builder v2*”, apenas permitia retirar o uso da aplicação em dispositivos móveis com o sistema operativo *Android*. Isso acaba por limitar uma grande percentagem dos utilizadores de smartphones

Sendo assim, para este trabalho, teve como objetivo desenvolver uma aplicação multiplataforma, de maneira a abranger um maior público-alvo.

2. Levantamento e análise dos Requisitos

Para atingirmos os objetivos deste projeto, toda a atividade de desenvolvimento tem de ser elaborada e desenvolvida criteriosamente, de maneira a assegurar o bom funcionamento da aplicação. Sendo assim, foram definidos requisitos funcionais, estes que são requisitos de software, que englobam todas as funções que o sistema ou uma componente do sistema deve ser capaz de realizar.

A Tabela 1 indica-nos os requisitos funcionais da nossa aplicação, onde cada um deles tem um estado, que mostra em que fase da implementação se encontra, podendo ser:

- Por fazer;
- Parcialmente feito - Tem o funcionamento base implementado e alguma interação com a *user interface*;
- Feito - Implementado o funcionamento e a *user interface*;
- Abandonado.

ID	Nome	Descrição	Estado
RF1	Registo de conta	Permitir ao utilizador criar uma conta, com os seguintes dados: <ul style="list-style-type: none">• Nome de utilizador;• <i>Email</i>;• <i>Password</i>.	Feito
RF2	Efetuar login	Permitir que um utilizador previamente registado faça <i>login</i> introduzindo o nome de utilizador e <i>password</i> .	Feito
RF3	Listar questionários por modo de jogo	Mostrar ao utilizador autenticado, quais os questionários ativos nos diferentes modos de jogo	Feito

RF4	Listar questionários por categoria	Mostrar ao utilizador autenticado, quais os questionários que pertencem a uma certa categoria	Feito
RF5	Sistema de ordenação e filtragem	Conseguir filtrar e ordenar as listas de questionários por: <ul style="list-style-type: none"> • Categoria; • Modo de jogo. 	Feito
RF6	Modo de jogo Clássico	Modo em que cada pergunta tem um tempo específico para ser respondida. Em caso de não ser respondida, passa para a pergunta seguinte	Feito
RF7	Modo de jogo contrarrelógio	Modo em que o questionário tem um tempo total para ser respondido e não um tempo por pergunta	Feito
RF8	Modo de jogo morte súbita	Modo em que o questionário termina mal o utilizador erre uma pergunta	Feito
RF9	Modo questionário	Modo que apenas permite responder uma única vez aos questionários incluídos no mesmo	Feito
RF10	Atribuição de pontuações	Após um utilizador responder a um questionário, deve ser-lhe atribuída uma pontuação de acordo com a quantidade de respostas certas e o tempo de resposta.	Feito

RF11	Visualização de estatísticas	Deve permitir ao utilizador ver o resultado dos questionários respondidos incluindo: <ul style="list-style-type: none"> • Perguntas totais; • Respostas certas; • Pontuação obtida. 	Feito
RF12	Sistema de ranking nos diferentes modos de jogo	De acordo com a pontuação ganha ao responder aos questionários, os utilizadores são classificados em cada modo de jogo.	Feito
RF13	Imagens nos questionários	Permitir perguntas nos questionários que venham acompanhada por uma imagem fazendo upload da mesma na criação do questionário	Feito

Tabela 1 - Requisitos funcionais

Além destes, também temos os requisitos não funcionais que estão relacionados com o uso da aplicação em termos de desempenho, usabilidade, segurança e tecnologias envolvidas, etc.

A Tabela 2 indica os requisitos não funcionais:

ID	Nome
RNF1	Aplicação disponível para <i>Android</i> e <i>IOS</i>
RNF2	Aplicação desenvolvida em Flutter
RNF3	Aplicação comunica via REST com o servidor, lendo dados em formato JSON enviados pela <i>API</i>

Tabela 2 - Requisitos não funcionais

Durante a minha familiarização ao projeto, no *backoffice* foram identificados pontos a melhorar, relativamente ao pouco *feedback* que recebemos pelas mensagens de erro, como também, alguns aspetos relacionados com alguns dos campos de preenchimento do formulário ao criar questionários. Estes, por sua vez, acabavam por gerar alguma confusão ao utilizador. E por fim, também foram levados em consideração alguns aspetos, a ver com a maneira do site a alteração no tamanho da janela, do motor de busca a ser utilizado.

Por exemplo, quando estava a criar um questionário, recebia uma mensagem de erro pouco específica, o que me obrigava a perceber o que poderia estar mal. Ou ainda, durante a criação de um questionário no modo questionário, que não necessita dos parâmetros “tempo” nem “dificuldade”, ainda assim, estes campos pediam que fossem preenchidos. Todos estes pontos foram reajustados na solução final deste projeto.

3. Viabilidade e Pertinência

De acordo com um estudo publicado no site [statista.com](https://www.statista.com) intitulado “*Share of smartphone users that use an Apple iPhone in the United States from 2014 to 2021*”[2], cerca de 45% dos utilizadores de *smartphones* nos EUA e aproximadamente 20% no mundo, usam *iPhone* com o sistema operativo *iOS*. Estes dados sugerem ser bastante pertinente o desenvolvimento de uma aplicação que consiga ser compatível em ambos os sistemas operativos, *Android* e *iOS*. Neste trabalho não vamos dar atenção aos restantes sistemas operativos como o *Windows Phone* e o *Blackberry*, de acordo com o site [idc.com](https://www.idc.com) “*Worldwide Smartphone Shipment OS Market Share Forecast*”[3] estes correspondem a praticamente 0% das vendas de *smartphones* mundialmente.

Atualmente, especialmente devido à pandemia da COVID-19, cada vez mais as instituições de ensino, entre outras, optam pelo ensino à distância[4]. Uma destas formas de ensino é o *e-learning* que tira proveito de plataformas digitais como o Moodle[5], Zoom[6] ou o Microsoft Teams[7] para possibilitar um acompanhamento e correção das atividades realizadas pelos alunos. Um exemplo que temos da adoção destes questionários no que toca ao ensino, é a própria ULHT. Vários professores da universidade, incluindo professores que fazem parte do DEISI, têm vindo a utilizar vários tipos de questionários.

De acordo com o artigo “*Ten Benefits of Testing and Their Applications to Educational Practice*”[8] há evidências claras de testes/experiências psicológicas de que praticar a recuperação de conhecimentos depois de aprendê-los, por exemplo, respondendo a um questionário no início ou no final de uma aula, aumenta a probabilidade de mantermos esses conhecimentos por um tempo bem mais prolongado.

Responder a questionários pode provocar em nós diferentes estímulos e emoções. De acordo com John Kounios[9], Professor de Psicologia e diretor do programa doutoral em *Applied Cognitive and Brain Sciences na Drexel University (Pennsylvania)*, quando respondemos a uma pergunta corretamente, isso causa uma libertação de dopamina, mais conhecida como a “molécula da motivação”[10], no cérebro.

Além deste benefício de curto prazo, os questionários aplicam-se a áreas muito diversificadas, sendo assim, auxiliam a identificar lacunas no conhecimento. Por outro lado, auxiliam-nos na organização do nosso conhecimento, ajudando o cérebro a organizar o que aprendemos em grupos. Outro benefício dos questionários é que, também devido à sua limitação temporal para dar as respostas, são excelentes ferramentas para treinarmos o nosso cérebro diariamente, permitindo assim, termos um pensamento cada vez mais rápido.

Por fim, como falámos, no que diz respeito ao *e-learning*, como também em contexto de aula presencial, os questionários podem ter um efeito muito positivo tanto para os professores como para os alunos. Como por exemplo, após uma aula, por meio de um simples questionário, um professor pode obter um bom *feedback* e perceber se os alunos aprenderam ou não o que foi ensinado. E para os próprios alunos a realização frequente de questionários, ter acesso imediato aos seus próprios resultados e perceber quais os pontos fortes e/ou menos fortes vai incentivá-los a estudar que conseqüentemente ajuda a reduzir a procrastinação[11].

4. Solução Desenvolvida

Este trabalho não veio apenas replicar a aplicação desenvolvida anteriormente, mas também melhorá-la, tornando-a mais adaptada ao mercado atual. Atualmente, a aplicação também se encontra disponível para os sistemas operativos Android e iOS.

Desta maneira, a nova aplicação disponibiliza todas as funcionalidades já presentes no projeto Android Quiz Builder v2, sendo estas:

- Um sistema de registo/*login* de utilizadores seguro;
- Acesso aos diversos modos de jogo (Clássico, Morte-Súbita e Contrarrelógio);
- Acesso ao modo de questionário (só pode ser respondido uma vez por utilizador);
- Sistema de filtragem por modo de jogo;
- Ordenação dos questionários por Data de Criação, Dificuldade e Título;
- Dar feedback de acordo com a resposta escolhida. Por exemplo, destacar qual é a resposta certa com a cor verde e se a resposta está errada, a cor vermelha;
- Alerta de nova *High Score* (Pontuação Máxima);
- Sistema de *ranking*.

Cumprindo também com um dos objetivos pessoais que tinha para esta aplicação, foi possível dar-lhe uma aparência mais apelativa e moderna.

Por fim, como requisito de implementação bónus, foi implementado um sistema que permite criar questionários que contêm perguntas com imagens, dando desta maneira uma dinâmica diferente à plataforma.

4.1. Tecnologia Utilizada - Flutter

O Flutter é uma *framework open-source*[12] criada pela Google destinada para o desenvolvimento de aplicações móveis para *Android* e *iOS* compilados nativamente a partir do mesmo código, a seguir vamos ver um pouco mais sobre isso.

O Flutter[13] oferece-nos um bom desempenho através do Dart Native[14], que é direcionado para o desenvolvimento de programas para dispositivos (móveis, *desktop* ou *server*), permite a execução de código Dart compilado para ARM nativo ou código de máquina X64.

O Dart Native inclui uma Dart VM (máquina virtual) com compilação JIT (just-in-time) e um compilador AOT (*ahead-of-time*) para a produção do código máquina, que por sua vez faz com que os aplicativos sejam executados de uma maneira mais eficiente. A técnica de compilação JIT (*just-in-time*) é particularmente interessante porque dá origem a uma funcionalidade muito útil do Flutter, o “*Hot-reload*”[15]. Esta funcionalidade permite ao programador, de uma forma fácil e rápida, fazer experiências, ou seja, alterar ou construir interfaces de utilizador (UI), adicionar novos elementos, corrigir *bugs* e visualizar estas alterações em tempo real.

No que toca à construção de toda a UI, o Flutter trabalha à base de *widgets*. “Em Flutter, tudo é um *widget*”, ou seja, todas as componentes que constituem uma aplicação desenvolvida em Flutter são *widgets*. Estes podem ser divididos em duas categorias:

- **Widgets de interface:** São aqueles que o utilizador consegue visualizar nos diferentes ecrãs.
- **Widgets de layout:** Ajudam no posicionamento e organização dos *widgets* de interface (alinhamento, containers, colunas, linhas, grelhas, etc)

O Flutter não utiliza os *widgets* originais do sistema operativo de um determinado equipamento (OEM *widgets*). Por isso, fornece um grande conjunto de *widgets*, *layouts* que podem ser divididos em dois grupos de elementos que são amplamente utilizados, o Material Design[16] (estilo *Android*) e ainda, o Cupertino[17] com um *design* inspirado no *iOS*, que são amplamente utilizados.

4.2. Arquitetura do Sistema

A arquitetura que compõe esta plataforma de questionários é constituída por um *frontend* com três componentes principais, a aplicação móvel, o *website* público e o *backoffice*. Na Figura 1 podemos verificar como estes componentes estão interligados:

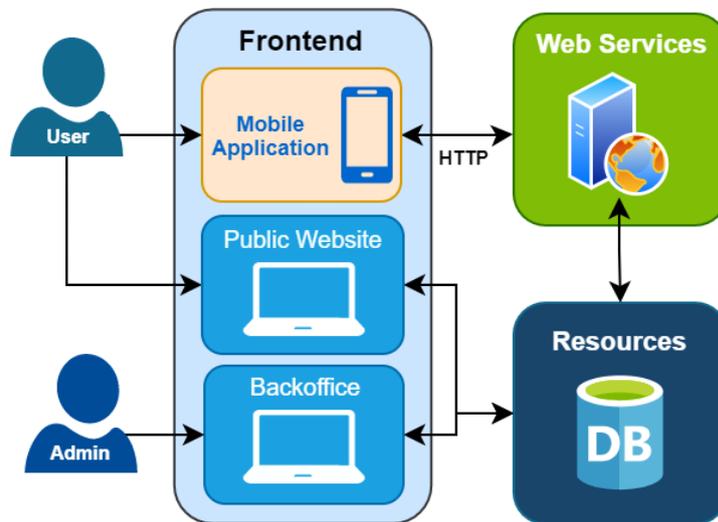


Figura 1 - Arquitetura do sistema

Aplicação móvel: Esta é a componente que queremos destacar neste TFC. A aplicação móvel, será desenvolvida do zero, em Flutter, tendo por base a aplicação antiga. Sendo assim, esta visa a recriar todas as funcionalidades presentes na atual aplicação *Android Quiz Builder*. Para isso a aplicação necessita de uma constante ligação à base de dados quer seja para ter acesso a todos os questionários como também para o envio, registo e tratamento dos resultados, sendo esta feita com o Sistema de Gestão de Bases de Dados (SGBD) MySQL. Esta comunicação com a base de dados será feita por meio de *Web Services* usando pedidos HTTP com o formato de mensagem JSON. Estas mensagens em formato JSON facilitam e aceleram esta comunicação com a base de dados.

Backoffice: O *backoffice* constitui todo um bloco funcional de administração do sistema, ou seja, só pode ser acessado por administradores. Este será o sítio onde os administradores poderão fazer toda a gestão dos questionários e também dos utilizadores da aplicação. Tanto o *backoffice* como o Website Público comunicam diretamente com a base de dados.

Website Público: Por fim temos o website público que pode ser acedido por qualquer utilizador, onde podem, por exemplo, consultar todas as suas estatísticas, criar questionários e gerir o seu perfil.

4.3. Storyboard

A elaboração de um *storyboard*, permite-nos identificar as principais funcionalidades que um potencial utilizador precisa de ter na aplicação, permitindo que estas sejam o foco quando chegar a parte do desenvolvimento da mesma.

O *storyboard* também tem um grande impacto, no que toca a ter uma boa representação visual da aplicação. Isto também vai auxiliar na fase do desenvolvimento da *app* em si, o *storyboard* vai servir como um guia para sabermos como desenvolver o formato dos diferentes ecrãs e ainda nos ajuda a perceber como se vai fazer a navegação entre eles.

A Figura 2 apresenta o *storyboard* atual da aplicação para o projeto *Ubiquous Quiz Builder*.

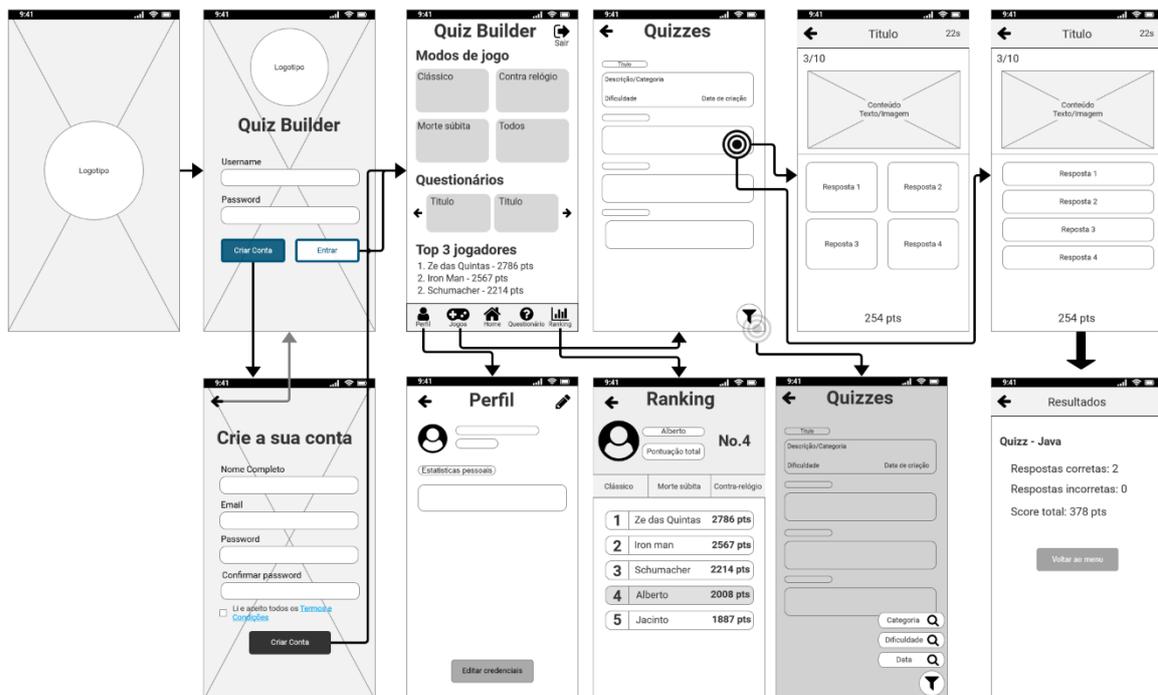


Figura 2 - Storyboard da aplicação móvel

4.4. Resultados dos Testes

Durante a realização dos testes com diversos utilizadores, foram identificados vários aspetos de melhoria, desde falhas técnicas como até alguns aspetos relativos à usabilidade da aplicação. Uma parte destas melhorias diz respeito ao feedback e sugestões obtidas durante as reuniões com os professores orientadores deste TFC.

O problema que mais se evidenciou foi o de registo de uma conta. Quando um utilizador tentava realizar esta ação, não estavam a ser verificados ambos os valores dos campos da palavra-chave (“palavra-chave” e “confirmar palavra-chave”). Assim sendo, um utilizador podia introduzir uma palavra-chave diferente nos dois campos que o mesmo era aceite e apenas registava o primeiro campo. Para além do mais quando não havia ligação à Internet este iniciava um *loading screen* infinito.

De momento todos estes problemas foram resolvidos, inclusive foi adicionado um número mínimo de caracteres (seis) para a palavra-chave tornando-a minimamente mais protegida.

Ainda relativamente aos formulários de registo e *login*, como dica por parte do Professor Orientador, estes sofreram algumas alterações visualmente, permitindo destacar-se melhor os *hint text*, o ícone para mostrar a palavra-chave, e a própria palavra-chave já que esta após preenchida, causa agora um maior contraste e menor confusão para o utilizador.



Figura 3 - Ecrã de *login* na fase intermédia



Figura 4 - Ecrã de *login* final

De seguida, durante um questionário, o utilizador podia voltar para trás e cancelar o questionário sem qualquer tipo de confirmação. O resultado disto era perder todo o progresso naquele questionário. Sendo assim, foi adicionado um *popup* quando o utilizador pressiona o botão para sair do questionário, que pede confirmação da sua saída, informando que perderá o progresso e assim o utilizador pode sair, ou continuar a responder ao questionário.

Outro aspeto melhorado, que foi destacado principalmente pelo professor orientador, foi a falta de contraste, dos elementos que constituem o ecrã para escolher a resposta. Sendo assim mudou-se a cor da caixa com as respostas como também a tonalidade do verde e do vermelho que sinalizam se a resposta está certa ou errada. Ficando com o seguinte resultado.

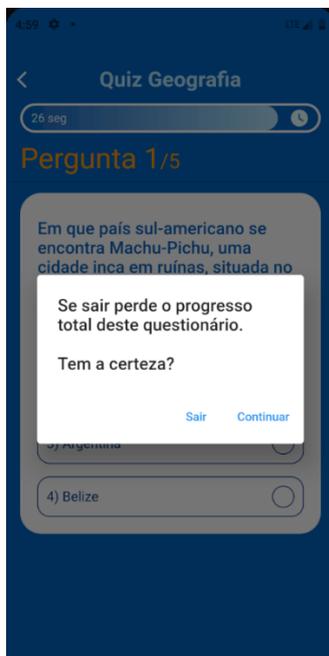


Figura 7 - Alerta ao abandonar o questionário



Figura 6 - Ecrã de resposta antes dos testes



Figura 5 - Ecrã de resposta depois dos testes

Já relativamente ao ecrã que mostra o resultado, após responder ao questionário no modo de jogo, este ecrã era exatamente igual ao do modo questionário, aparecendo neste último sempre uma pontuação com zero pontos e o número de respostas certas e erradas com valor zero pois estes dados não são considerados neste modo.

No que toca ao *feedback* obtido no final dos questionários no modo de jogo, era exibida apenas uma mensagem predefinida independentemente da pontuação obtida.

Atualmente dependendo do tempo perdido por cada pergunta e da pontuação final, são apresentadas mensagens diferentes, como por exemplo:

- Se o utilizador não consegue acertar nenhuma pergunta é apresentada a mensagem: “Este não correu muito bem, mas não desistas!”.
- No caso de ter errado mais perguntas do que as que acertou, então a mensagem exibida será: “Vamos lá! Tu consegues melhor.” E se conseguir acertar acima da metade então será: “Boa! Conseguiste acertar mais de metade do questionário.”. Caso acerte metade então seria: “Boa! Conseguiste acertar metade do questionário.”.
- Se conseguiu acertar todas as perguntas, mas demorou em média mais de 5 segundos por pergunta, então é apresentada a mensagem: “Muito bem! Conseguiste acertar todas as perguntas, agora tenta ser mais rápido.”
- Por fim, se acertou todas as perguntas e demorou em média menos de 3 segundos por pergunta, então a mensagem exibida é: “Incrível! Estás mesmo quase a obter a pontuação máxima.”.

5. Benchmarking

Nesta seção serão apresentadas as diferentes tecnologias que foram estudadas e consideradas para o desenvolvimento da aplicação, assim como, duas aplicações de questionários que estão atualmente disponíveis no mercado.

5.1. Tecnologias

Atualmente no mercado existem diversas *frameworks* para o desenvolvimento de aplicações híbridas[18]. Mas primeiramente vamos ver quais são as principais diferenças entre aplicações híbridas e nativas. De uma forma resumida, uma aplicação nativa é uma aplicação que é desenvolvida para um sistema operativo específico (*Android* ou *iOS*), ao passo que uma aplicação híbrida é desenvolvida para múltiplas plataformas com a mesma base de código. A Figura 8 apresenta algumas vantagens e desvantagens entre as duas:

	Híbrida	VS	Nativa
Prós	<ul style="list-style-type: none"> • Uma única base de código • Baixo custo • Maior facilidade e agilidade de desenvolvimento • Manutenção mais simples e comum a todas as plataformas • Entrega rápida para o mercado • Acesso a funcionalidades do sistema por meio de <i>plug-ins</i> 		<ul style="list-style-type: none"> • Melhor <i>performance</i> em geral • Proteção de dados • Melhor acesso a funcionalidades do sistema • Experiência de utilizador • Interface nativa permite aos utilizadores aprenderem mais rapidamente
Contras	<ul style="list-style-type: none"> • Algumas inconsistências entre plataformas • Aplicações complexas com muitos recursos podem se tornar mais lentos. 		<ul style="list-style-type: none"> • Desenvolvimento mais demorado e sai mais caro manter uma aplicação em diferentes sistemas operativos. • Base de código distribuída

Figura 8 - Prós e contras de aplicações nativas e híbridas

O React Native[19] é uma *framework open-source*, que foi lançada em março de 2015 pelo Facebook e é usada para criar aplicativos móveis. O React Native utiliza como linguagem de programação para o desenvolvimento de aplicações móveis, o JavaScript.

A alta popularidade e a posição no mercado do JavaScript é indiscutível e isso reflete-se no relatório do GitHub deste ano, “The 2020 - State of the Octoverse”[20] onde mostra que desde 2014, JavaScript se encontra na primeira posição do “*Top languages over the years*”. Por estas razões é muito provável que os programadores que estão mais acostumados com esta linguagem prefiram o uso do React Native para o desenvolvimento de aplicativos móveis.

Por outro lado, o site [statista.com](https://www.statista.com) apresenta o gráfico, “*Cross-platform mobile frameworks used by software developers worldwide in 2019 and 2020*”[21], que resulta de um questionário feito pela JetBrains acerca de quais são as principais *frameworks* escolhidas pelos *software developers*. Este gráfico mostra que a diferença entre as percentagens de *developers* que utilizam o React Native e o Flutter passou de 12% para 3%. Neste ano de 2020, 42% dos *software developers* afirmaram usar o React Native e 39% o Flutter. Podemos notar que apesar do Flutter ser uma *framework* mais recente, já começa a ser uma escolha para muitos programadores.

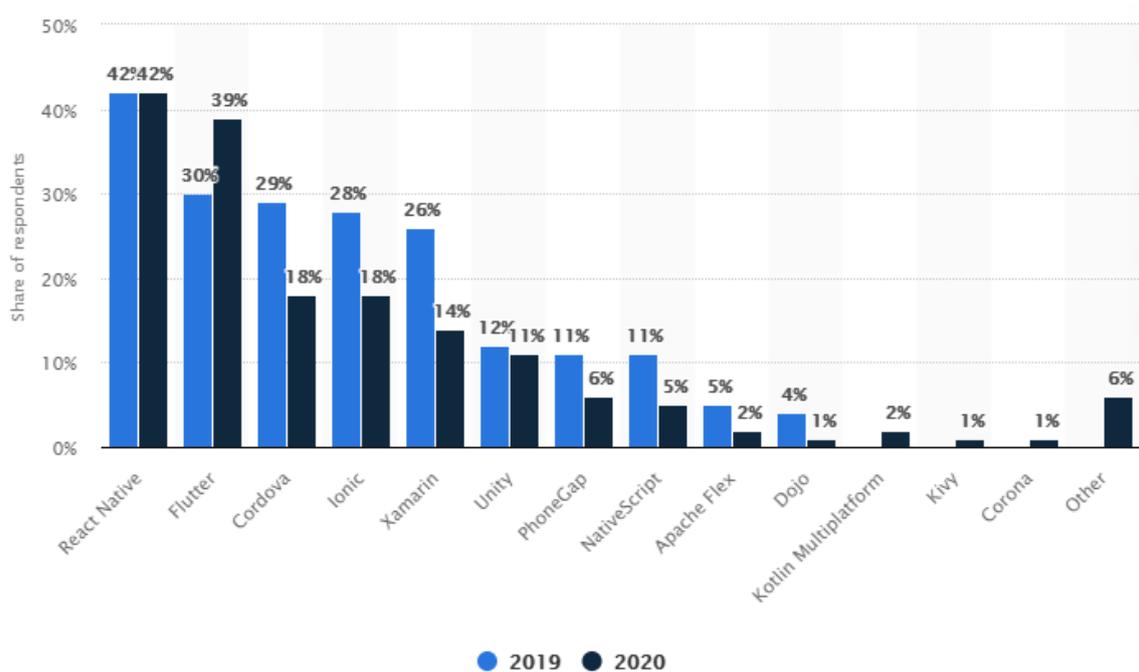


Figura 9 - Principais *frameworks* escolhidas por *software developers*

Outro aspeto notável relativamente ao Flutter tem vindo a ser o aumento da sua comunidade. Como podemos ver no gráfico da Figura 10, com o passar dos anos tem vindo a ser cada vez mais, procurada informação relativamente ao Flutter e devido ao aumento da sua popularidade, encontrar ajuda online fica cada vez mais fácil, o que é de grande benefício para toda a comunidade de *developers*.

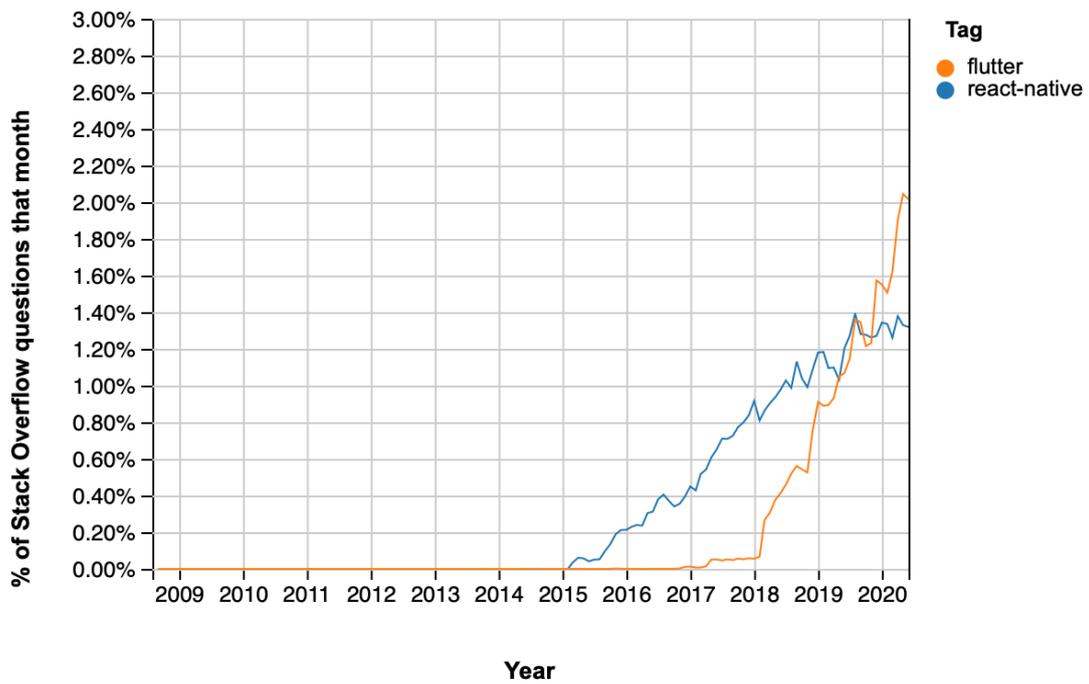


Figura 10 - Gráfico que compara o número de perguntas feitas do Stack Overflow[22]

5.2. Aplicações no mercado

Agora vamos ver algumas aplicações de questionários, que poderão ser comparadas com a nossa, quer no âmbito de entretenimento como para a área de ensino.

No que toca ao ensino, temos a plataforma *Socrative*[23] que disponibiliza duas aplicações e um acesso *web*. Esta plataforma tem dois tipos de utilizadores, professor e aluno. O professor à semelhança do nosso administrador é quem pode criar, gerenciar os questionários, salas e também os relatórios gerados pelos resultados dos questionários. Sendo assim existem duas aplicações: *Socrative Student* e *Socrative Teacher*.

Ambas as aplicações são muito simples de usar e dinâmicas. O *Socrative Student* apenas serve para que o aluno se possa juntar a uma sala, que é criada previamente pelo professor. No que toca ao *Socrative Teacher* esta permite ao utilizador a criação de questionários que podem ser respondidos pelos alunos nas salas virtuais, sendo estes muito úteis para consolidação de conhecimento e também por exemplo na preparação para exames. Ainda existem vários modos e customizações para cada questionário tornando o seu uso muito mais interativo.

De acordo com o seu guião[24], esta aplicação pode ser muito útil para ser usada em sala de aula. Pois por meio de um sistema de perguntas e respostas o professor pode receber feedback em tempo real da aprendizagem dos alunos.

Vamos agora ver algumas imagens dos ecrãs destas aplicações.

5.2.1. Socrative (Teacher & Student) Apps

Ecrãs de *Login*, homepage e navigation bar:

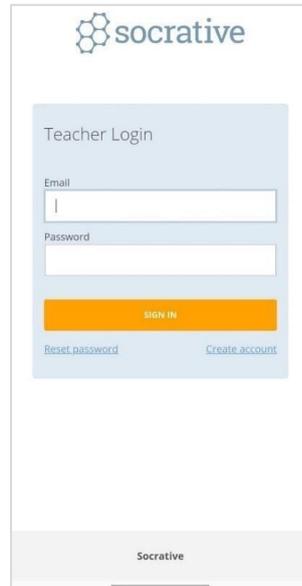


Figura 11 – Ecrã login

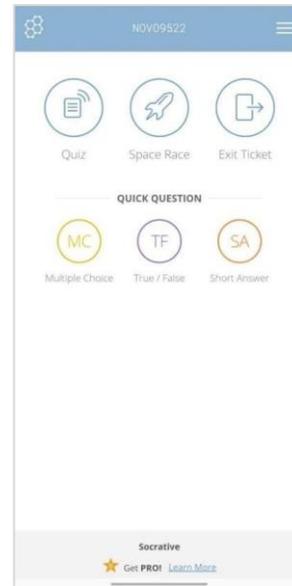


Figura 12 - Homepage

- Exemplo do funcionamento de um questionário base, que dependendo da pergunta, quando a resposta é escolhida, destaca uma mensagem que confirma o resultado e que pode também incluir uma breve explicação da resposta certa independentemente se foi selecionada a resposta certa ou não:



Figura 15 - Ecrã pergunta com imagem

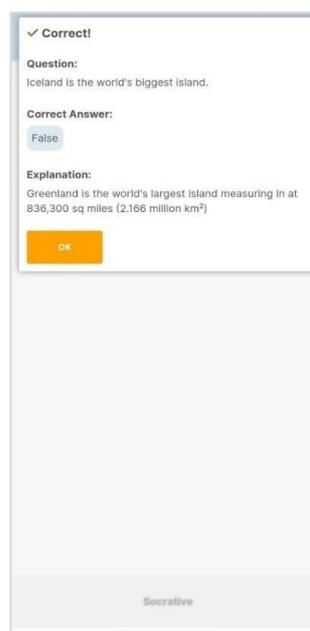


Figura 14 - Feedback de uma resposta correta

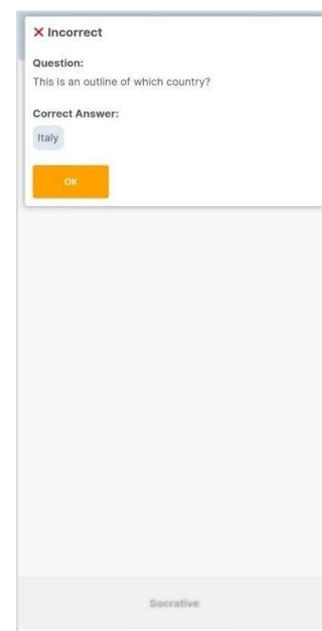


Figura 13 - Feedback de uma resposta incorreta

Outra funcionalidade muito interessante, que pode ser vista na Figura 15 acima é o suporte de imagens nos questionários que é um requisito para ser implementado no futuro.

- Visualização dos resultados dos questionários:

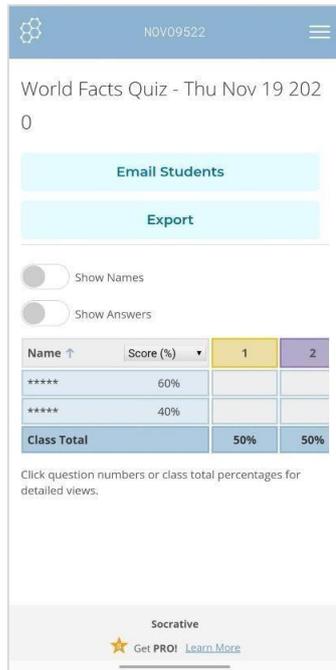


Figura 16 - Ecrã dos resultados geral



Figura 17 - Ecrã dos resultados detalhado

No que toca a entretenimento, temos a aplicação *QuizUp*, que também se encontra disponível tanto para *Android* como para *iOS*.

5.2.2. QuizUp

Esta é uma aplicação para responder a questionários mais num estilo de jogo, permitindo até a jogabilidade entre dois utilizadores como forma de competição, que apesar de fugir um pouco ao nosso objetivo, é uma funcionalidade muito interessante.

Porém algo que me chamou à atenção foi a sua interatividade na aplicação em geral, como também, quando estamos a responder aos questionários. O QuizUp permite a filtragem dos questionários por categorias e interesses dos utilizadores, que são indicados quando este inicia a aplicação pela primeira vez.

Em baixo, fica uns exemplos dos ecrãs de um questionário individual na categoria de carros:

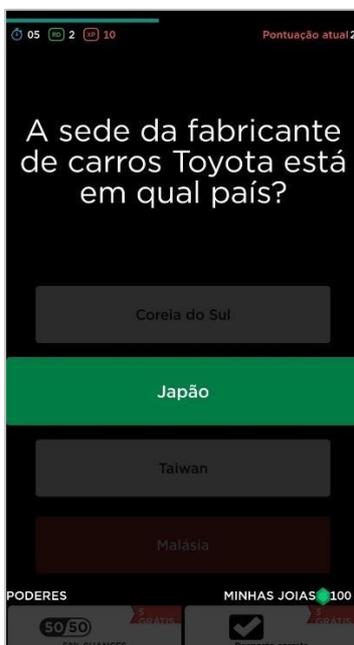


Figura 20 - Feedback interativo da opção escolhida



Figura 19 - Ecrã de escolha da resposta



Figura 18 - Ecrã da categoria de carros

Como podemos visualizar, a Figura 18 mostra algumas informações sobre a categoria em questão, mas gostaria de destacar a Figura 19 e a Figura 20, onde nos mostra a interface na resposta ao questionário. Em cima podemos verificar que temos um cronómetro normal, mas também temos um objeto gráfico (barra verde) no topo que nos mostra o tempo que nos resta.

Por fim, a Figura 20 mostra uma maneira muito apelativa e interativa de destacar a resposta correta, fazendo um *fade-out* de todas as outras respostas, não deixando dúvidas ao utilizador acerca de qual era a resposta correta.

6. Método e Planeamento

A fase mais inicial do desenvolvimento da aplicação, envolveu muita pesquisa acerca da tecnologia utilizada, o Flutter. Foram assim explorados diversos métodos para o desenvolvimento da solução e abordagem com a troca de dados com o servidor.

6.1. Fases de desenvolvimento da aplicação

Após a exploração inicial da *framework*, o desenvolvimento da aplicação foi dividida em diferentes objetivos principais:

1º Objetivo – Implementação de *web services* REST que comuniquem com a API da plataforma por meio de pedidos HTTP. Desta maneira, seria possível efetuar login, registo de um novo utilizador, obter todos os dados necessários, incluindo toda a informação dos questionários, respetivas perguntas e respostas, resultados e os dados dos utilizadores.

Para a implementação destes *web services*, foi sugerido pelo professor Miguel Tavares que utilizasse uma biblioteca adicional chamada de Chopper.

Este primeiro objetivo acabou por se estender um pouco mais do que seria esperado acabando por atrasar um pouco o resto do desenvolvimento da aplicação.

2º Objetivo – Desenvolvimento de ecrãs estáticos da aplicação, como ecrã de login e lista de questionários, registo e homepage. Para realizar o *state management* da aplicação foi utilizado o padrão BLoC, recomendado pela Google e que também foi abordado inicialmente nas aulas de Computação Móvel.

Este padrão permite abstrair a camada de apresentação, da lógica do negócio e tratamento de dados, tornando assim o código mais organizado, simples de testar e reutilizável.

3º Objetivo – Implementação dos ecrãs de resposta aos questionários. Inicialmente apenas foi desenvolvido no modo clássico, ou seja, com um tempo específico que havia sido previamente estabelecido no próprio código (30s). Mais tarde após garantir que existia uma transição correta entre as perguntas do questionário, mostrando também o *feedback* relativamente à resposta selecionada. Foram implementadas restrições e adaptações para os outros modos de jogo e o modo questionário.

4º Objetivo – Cálculo de estatísticas e envio de dados para o servidor. Esta fase envolveu o cálculo da pontuação dos questionários pertencentes aos diversos modos de jogo. Aqui a pontuação final é baseada no número de respostas certas, erradas e o tempo de cada

resposta, permitindo assim haver uma melhor competitividade entre os diversos utilizadores da aplicação.

5º Objetivo – Desenvolvimento do ecrã com os rankings de cada modo de jogo. Esta fase envolveu o tratamento de todos os dados dos resultados obtidos do servidor e o cálculo de posições no *ranking* de cada modo de jogo existente.

6º Objetivo – Sistemas de ordenação das listas de questionários, por data de criação, dificuldade e título.

8º Objetivo – Tratamento de exceções com a falta de ligação à *Internet*. Uma vez que para aceder à aplicação o utilizador precisa obrigatoriamente de fazer *login*, não existindo atualmente um utilizador “convidado” (que funcione *offline*), é necessária ligação à *Internet* ao entrar na mesma. Após estar dentro da aplicação esta obtém os dados iniciais do servidor permitindo assim que o utilizador a utilize sem acesso à *Internet*. Exceto no envio dos resultados ao servidor que ficam mantidos em cache, até que seja possível estabelecer novamente ligação com o servidor e aí a aplicação faz o envio dos resultados em cache.

9º Objetivo – Criação do ecrã de perfil onde o utilizador pode verificar algumas das suas estatísticas.

10º Objetivo – Início de uma fase de testes que será abordada na próxima secção.

11º Objetivo – Correção de bugs e aspetos relativamente à usabilidade da aplicação baseado nos resultados dos testes e nas revisões feitas pelos professores orientadores ao longo do desenvolvimento deste TFC.

6.2. Plano de Testes e Validação

Com o intuito de testar as várias funcionalidades da aplicação e a usabilidade da mesma, foram desenvolvidas diversas tarefas que serão fornecidas aos utilizadores.

6.2.1. Tarefas para os utilizadores

Para resolver estas tarefas, planeou-se a realização de testes com um grupo de utilizadores. Para facilitar a logística destes testes, ainda por mais devido à situação pandémica que estamos a passar atualmente, para além do ambiente local de desenvolvimento da aplicação, também foi disponibilizado o acesso remoto à plataforma web (que também serve de API para a aplicação) como também à base de dados permitindo assim criar um ambiente que será usado para o grupo de utilizadores realizarem os testes à aplicação.

Assim sendo, será fornecido um guião com as diferentes tarefas a desempenhar por cada testante, que estará incluído num formulário em Google Form[25], onde o utilizador se poderá expressar acerca da facilidade com que realizou as tarefas, se foi possível realizar as mesmas e quais foram alguns problemas ou bugs com que se possam ter deparado.

Link para o guião de tarefas (no Anexo 3 encontram-se descritas as tarefas):

<https://forms.gle/NDT47zQEe7umoToTA>

6.3. Calendário

Foi elaborado um Mapa de Gantt para guiar o desenvolvimento da aplicação e facilitar a deteção e o impacto de eventuais atrasos:

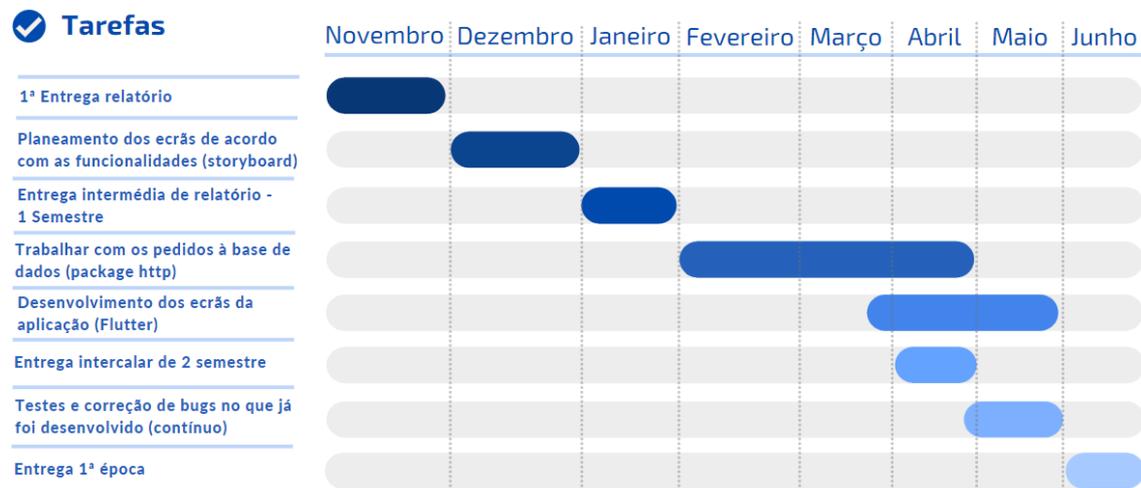


Figura 21 - Esquema de calendarização do desenvolvimento do TFC

Após a primeira entrega intercalar, iniciei com o planeamento dos ecrãs da aplicação, por meio da elaboração de um *storyboard*, assim consegui identificar os ecrãs que constituem a minha aplicação e como funcionaria a navegação entre eles.

Seguidamente, comecei a aprender e explorar melhor o funcionamento dos *web services* no Flutter e que devido à sua complexidade, por se tratar de uma linguagem nova, foi um processo mais demorado. Isto acabou por fazer com que se atrasasse um pouco o desenvolvimento da interface gráfica, que comecei a desenvolver entre o fim de março e início de abril.

Daí em diante, sendo possível obter todos os dados sem problema, restava dar vida aos mesmos por meio da interface gráfica. Os planos eram, desenvolver os ecrãs que faltam como o de *Rankings* e Perfil, e o resto da implementação dos modos dos questionários.

No início de Outubro, já se encontravam desenvolvidos praticamente todos os ecrãs da aplicação. Sendo assim, inicio-se uma fase de testes e dependendo do feedback do mesmo, alterações iam sendo feitas na aplicação de maneira garantir a sua melhor *performance*.

Ouve um grande atraso nesta fase de testes, mas no final tudo se desenvolveu como planeado

7. Resultados

Atualmente, a solução final desenvolvida cumpre todos os requisitos propostos inicialmente. Por meio do *backoffice/website*, é permitida a criação de questionários nos diversos modos disponíveis (Clássico, Contrarrelógio, Morte-Súbita e Questionário), que por sua vez são respondidos por meio da aplicação móvel.

7.1. Backoffice/Website Público

O *website*, como foi dito anteriormente, permite-nos criar diversos tipos de questionários. E de maneira a simplificar a criação dos mesmos, foram feitas algumas mudanças nos formulários que envolvem a criação de um questionário.

Começando no ecrã onde se preenchem as informações sobre o questionário (título, descrição, tempos e dificuldade), foram feitas algumas alterações dependendo do modo do questionário que está a ser criado.

A primeira alteração está no aumento dos campos Título e Descrição, permitindo ao utilizador escrever mais texto sem ficar omitido o que está para trás. O campo da Descrição até é mesmo redimensionável conforme der mais jeito ao utilizador.

Dependendo do modo do questionário, os campos de tempo e dificuldade são omitidos ou alterados. Por exemplo, como no modo clássico o tempo é definido para cada pergunta, então é pedido o “Tempo por pergunta”, já no modo Contrarrelógio pede o “Tempo total do questionário”, não causando quaisquer dúvidas ao utilizador.



Figura 23 - Formulário para criar questionário (versão antiga)

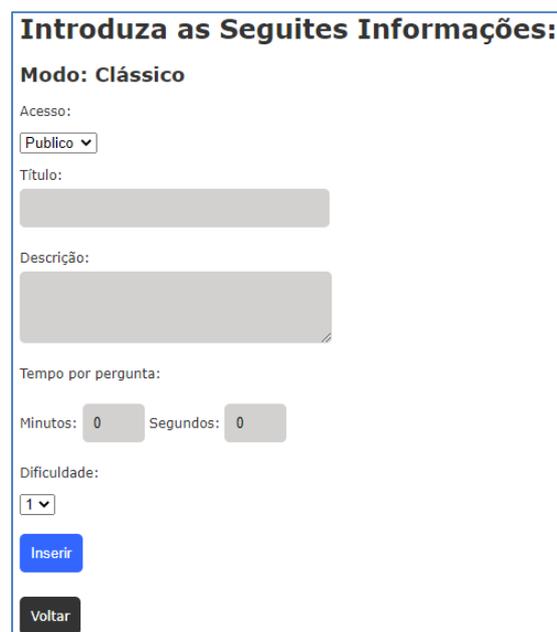
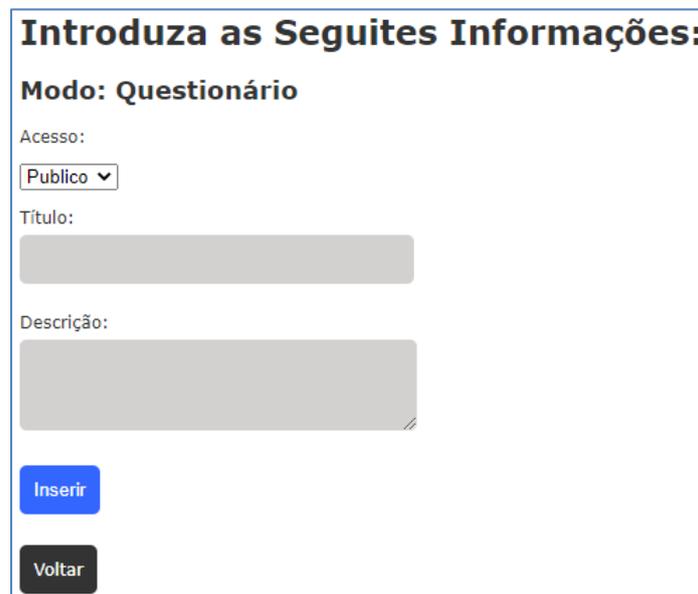


Figura 22 - Formulário para criar questionário (nova versão)

No que toca ao modo Questionário, não havendo qualquer tipo de valorização sobre os campos de tempo e dificuldade, estes são omitidos por completo evitando confusão ao utilizador.



O formulário, intitulado "Introduza as Seguintes Informações:", contém os seguintes elementos:

- Modo:** Questionário
- Acesso:** Um menu suspenso com a opção "Publico" selecionada.
- Título:** Um campo de texto cinzento.
- Descrição:** Um campo de texto cinzento.
- Botões:** "Inserir" (azul) e "Voltar" (preto).

Figura 24 - Formulário de criação para o modo Questionário

De seguida, como maior novidade no formulário das perguntas do questionário, foi implementado um campo que permite adicionar uma imagem a cada pergunta do questionário. Esta que acaba por ser codificada em Base64 e posteriormente guardada na base de dados no registo da respetiva pergunta. Para além desta adição, também foram feitas algumas modificações nos campos já existentes de maneira a tornar a página mais *responsive* e mais amigável ao utilizador.

Como podemos observar na Figura 25, o campo da pergunta foi alterado (passou a ser uma `TextArea`) de maneira a ser visível o seu texto na totalidade. Também conseguimos reparar que inicialmente existiam três botões neste ecrã, o de "Inserir", de "Finalizar" e "Voltar". Para não causar confusão ao utilizador, estes botões foram renomeados e agrupados. Desta maneira passamos a ter mais perto das perguntas, o botão "Inserir Pergunta" e um pouco abaixo o botão de "Cancelar" em conjunto com o de "Submeter Questionário", estes últimos, que acabam por tomar decisões finais, para o futuro do questionário que está a ser criado.

Insira as suas perguntas e as respetivas respostas :

Pergunta 1:

[Redacted question text]

Insira as respetivas respostas(mínimo 3 respostas):

Resposta Texto	Resposta Certa
[Redacted]	<input type="radio"/>

Figura 26 - Formulário para inserir pergunta (versão antiga)

Insira as suas perguntas e as respetivas respostas :

Pergunta 1:

[Redacted question text]

Insira as respetivas respostas(mínimo 3 respostas):

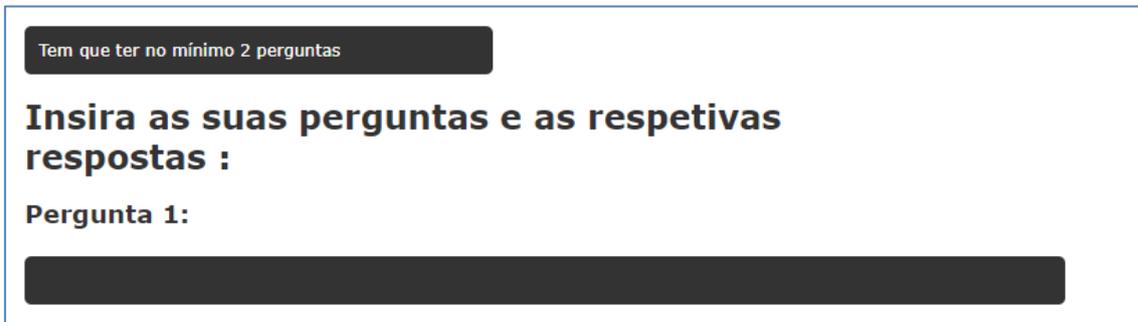
Resposta Texto	Resposta Certa
[Redacted]	<input type="radio"/>

Se quiser que esta pergunta mostre uma imagem, insira em baixo:

Nenhum ficheiro selecionado

Figura 25 - Formulário para inserir pergunta (nova versão)

Por fim, as últimas alterações feitas na parte visual da plataforma, foi a modificação de algumas mensagens de erro, especificando melhor ao utilizador qual foi o problema que ocorreu ao preencher o formulário da pergunta. A cor destas mesmas mensagens também foi alterada de maneira a destacar melhor que é um erro.



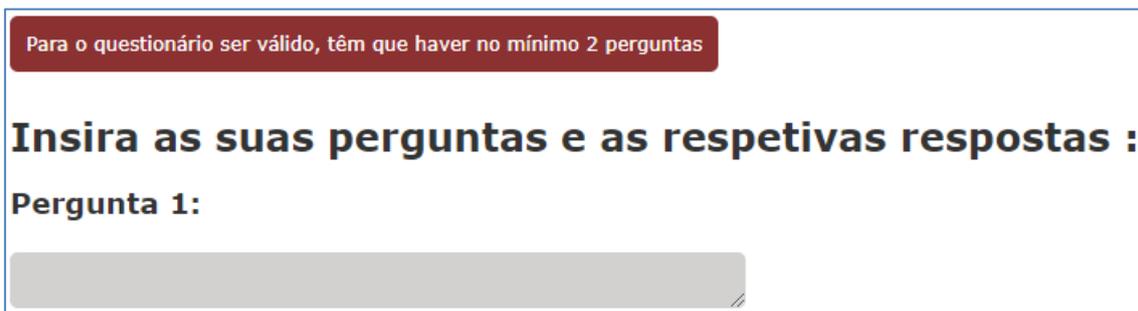
Tem que ter no mínimo 2 perguntas

Insira as suas perguntas e as respetivas respostas :

Pergunta 1:

[Redacted input field]

Figura 27 - Erro ao finalizar questionário (versão antiga)



Para o questionário ser válido, têm que haver no mínimo 2 perguntas

Insira as suas perguntas e as respetivas respostas :

Pergunta 1:

[Redacted input field]

Figura 28 - Erro ao finalizar questionário (nova versão)

Por fim, foram feitas algumas alterações no backend. Em certas ocasiões, como por exemplo, ao fazer login de um utilizador, o programa ia buscar todas as entradas da tabela de utilizadores da base da dados (recorrendo à query “*SELECT * FROM users*” chamada na função “*get_all_users()*”). E em seguida, como podemos observar na figura X, por meio de um “*for loop*”, percorria todos os dados obtidos e verificava se os dados de algum destes utilizadores coincidia com os dados do utilizador que está a tentar fazer *login*.

```
$users=get_all_users($pdo);

foreach($users as $user){
    if($user["nome"]==$username &&
        $user["password"]==sha1($password)){

        if($user["tipo"]=="user"){
            echo "<h1>user</h1>";
        }else if($user["tipo"]=="admin"){
            echo "<h1>admin</h1>";
        }

        $message="";
        $user_id=$user["id"];
        $_SESSION['user_id']=$user_id;
        header('location: questionariosAtivos.php');

    }else{
        $message="Username ou Password incorreta";
    }
}
```

Para evitar o gasto de memória desnecessário ao obter todos os dados de cada utilizador, passou-se a ir buscar apenas uma entrada da base de dados (“*SELECT * from users WHERE (nome=:nome AND password=:pwd)*”), assim só são verificadas as credenciais do utilizador com as. E como podemos visualizar na FIGURA x apenas se estas coincidirem, então o utilizador faz *login* com sucesso.

```
$pwdHash = sha1($password);
try {
    $sql = "SELECT * from users WHERE
           (nome=:nome AND password=:pwd)";

    $statement = $pdo->prepare($sql);
    $statement->bindParam(':nome', $username);
    $statement->bindParam(':pwd', $pwdHash);
    $statement->execute();
    $users = $statement->fetchAll();

} catch (Exception $ex) { echo $ex->getMessage(); }

if(sizeof($users) == 1){
    // Se encontrou um utilizador com as
    // credenciais inseridas entao faz login

    if($users[0]['tipo'] == 1){
        echo "<h1>admin</h1>";
    }else{
        echo "<h1>user</h1>";
    }

    $message="";
    $user_id=$users[0]["id"];
    $_SESSION['user_id']=$user_id;
    header('location: questionariosAtivos.php');

}else{
    $message="Username ou Password incorreta";
}
```

7.2. Aplicação Móvel

A solução implementada visou responder a todos os requisitos, resultando assim numa aplicação disponível para Android e iOS, que permite responder a diversos questionários, seja em modo de jogo, ou no modo questionário, que apenas pode ser respondido uma vez.

Após a Avaliação Intercalar, o foco esteve maioritariamente no desenvolvimento da aplicação, correção de bugs e garantir o seu bom funcionamento.

Por volta de meados de março, ao contrário do que havia sido definido inicialmente no calendário (que era começar pelo desenho dos ecrãs), decidi começar a aprender a trabalhar com os web services do tipo REST, pois estes seriam a base para ir buscar toda a informação que compunha o resto da aplicação. Pode-se dizer que foi um processo relativamente demorado, envolvendo também toda a conversão dos dados recebidos em formato JSON para objetos do Dart.

Posteriormente, foi iniciado o desenvolvimento da User Interface, tendo desenvolvido até à Avaliação Intercalar:

- Ecrã de *login*
- *Homepage*
- Ecrã com a lista de questionários
- Ecrã inicial dos questionários com algumas informações do mesmo
- Ecrã do questionário (modo Clássico)

No que diz respeito aos modos de jogo, o que estava implementado era apenas o modo Clássico, por isso ainda faltava implementar os outros dois modos (Contrarrelógio e Morte-Súbita) que são variações do Clássico.

Para facilitar a visualização do progresso até à data da entrega intermédia, foi feito um vídeo que mostra os diversos ecrãs e funcionalidades implementadas.

Link do vídeo demonstrativo do 1º protótipo Intermédio:

<https://youtu.be/MMza2qZrjus>

Atualmente a aplicação já tem implementados todos os requisitos iniciais.

Assim sendo, foram implementados os seguintes ecrãs:

- Ecrã de registo e *login*:

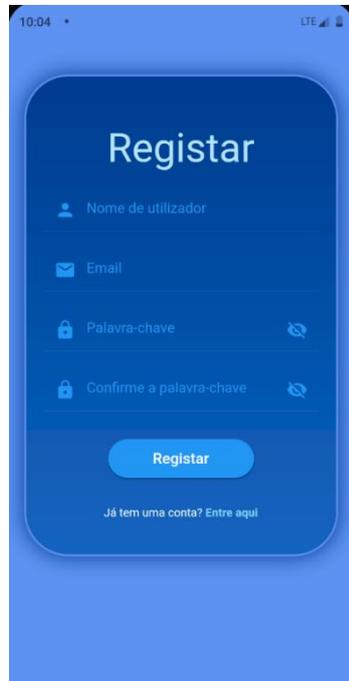


Figura 30 - Ecrã de Registo.



Figura 29 - Ecrã de Login.

- *Homepage* e ecrã do Perfil:



Figura 32 – *Homepage*.



Figura 31 - Ecrã do Perfil.

- Ecrã listas dos questionários num modo de jogo, em todos os modos de jogo e no modo questionário:



Figura 33 - Lista de questionários do modo Clássico.



Figura 34 - Lista de questionários de todos os modos de jogo.



Figura 35 - Lista de questionários do modo Questionário.

- Sistema de filtragem e ordenação na lista de questionários;



Figura 36 - Filtros da lista de questionários.

- Ecrã inicial do questionário e ecrã de resposta a um questionário do modo questionário:

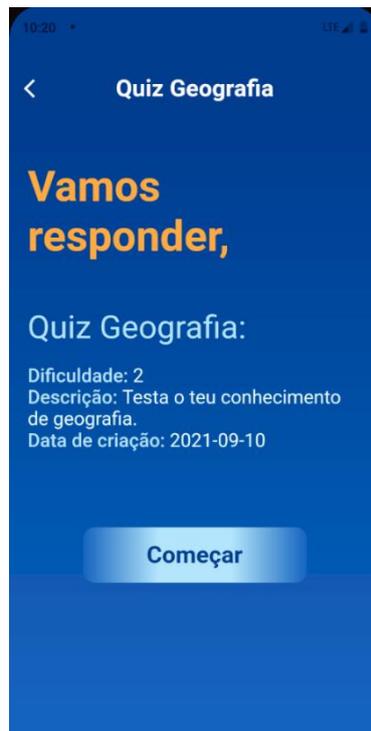


Figura 37 - Ecrã com informações do questionário.

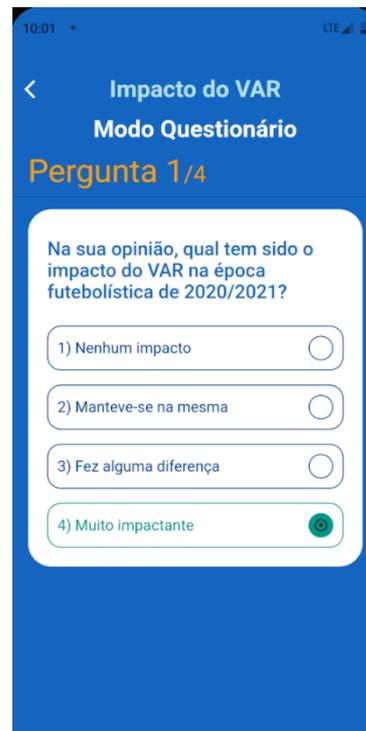


Figura 38 - Ecrã de resposta no modo Questionário.

- Ecrã de resposta ao questionário no modo de jogo (com e sem imagem):



Figura 39 - Ecrã de resposta no modo de jogo com imagem.

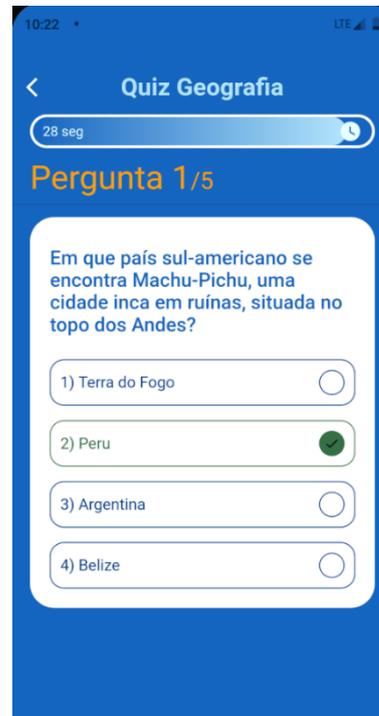


Figura 40 - Ecrã de resposta no modo de jogo sem imagem.

- Ecrã do resultado dos questionários no modo questionário e no modo de jogo:



Figura 42 - Ecrã de resultado no modo questionário.



Figura 41 - Ecrã de resultado no modo de jogo.

- Alerta que mostra se a pontuação obtida foi o seu novo recorde pessoal:



Figura 43 - Alerta de novo recorde pessoal.

- Ecrã dos *rankings* (topo de pontuação dos utilizadores por cada modo de jogo)



Figura 45 - Ecrã de *ranking* no modo Clássico.



Figura 44 - Ecrã de *ranking* no modo Contrarrelógio.

Dois pontos a destacar relativamente à versão anterior desta aplicação (*Android Quiz Builder*), é primeiramente, a implementação de perguntas com a possibilidade de exibir uma imagem. Sendo assim, a aplicação permite responder a questionários criados por meio da plataforma *web*, que também contenham perguntas com imagens.

Um segundo ponto que foi levado em conta ao longo de todo desenvolvimento da aplicação, além do bom funcionamento, é dar um aspeto mais moderno à mesma, permitindo que esta se destaque melhor no mercado atual.

Link do vídeo da solução final da aplicação no repositório:

<https://github.com/diogo19o/Ubiquous-Quizz-Builder>

8. Conclusão

Este TFC permitiu-me explorar novas tecnologias e linguagens de programação como o Flutter, PHP e ainda foi de grande ajuda para aumentar os meus conhecimentos na área de MySQL. Maioritariamente, foi muito proveitoso para entender melhor como funcionam os *web services* REST e as diversas maneiras que os podemos utilizar para troca de dados entre um servidor e um cliente.

Outro ponto que foi muito útil para a minha aprendizagem, foi a implementação do último requisito, de incluir imagens nas perguntas dos questionários. Pois o que parece à primeira vista ser muito complexo, acaba por ser um processo relativamente acessível.

No geral, gostei muito da experiência de desenvolver uma aplicação híbrida e vivamente aconselho a investir um pouco de tempo a experimentar desenvolver aplicações sejam elas móveis, web ou para desktop com Flutter.

Bibliografia

[1] Flutter

<https://flutter.dev/>

[2] Website Statista (consultado no dia 25 de outubro de 2020).

<https://www.statista.com/statistics/236550/percentage-of-us-population-that-own-a-iphone-smartphone/>

[3] Website idc (consultado no dia 25 de outubro de 2020.)

<https://www.idc.com/promo/smartphone-market-share/os>

[4] Aumento do ensino à distância (consultado no dia 22 de abril de 2021)

<https://www.weforum.org/agenda/2020/04/coronavirus-education-global-covid19-online-digital-learning/>

[5] Moodle

<https://moodle.org/?lang=pt>

[6] Zoom

<https://zoom.us/pt-pt/meetings.html>

[7] Microsoft Teams

<https://www.microsoft.com/pt-pt/microsoft-365/microsoft-teams/free>

[8] Artigo

January 2011 - Psychology of Learning and Motivation 55:1-36; Chapter:

Psychology of learning and motivation: Cognition in education Publisher: Oxford:

Elsevier

[9] Artigo - *Why Trivia? Three Major Benefits of Playing Trivia* (consultado 27 no dia 27 de novembro de 2020)

<https://www.sporcle.com/blog/2018/10/three-major-benefits-of-playing-trivia/>

[10] Dopamina – Molécula da motivação (consultado no dia 25 de outubro de 2020.)

<https://www.farmaciasportuguesas.pt/menu-principal/bem-estar/a-molecula-da-motivacao-conheca-a-dopamina.html>

[11] *Ten benefits of quizzes* (consultado no dia 26 de outubro de 2020)

<https://www.questionmark.com/ten-benefits-of-quizzes-and-tests-in-educational-practice/>

[12] Repositório Flutter

<https://github.com/flutter/flutter>

- [13] *Website Concise Software* (consultado no dia 12 de novembro de 2020)
<https://medium.com/@concisesoftware/what-is-flutter-here-is-everything-you-should-know-faed3836253f>
- [14] Dart Native
<https://dart.dev/platforms>
- [15] *Hot-reload* Flutter
<https://flutter.dev/docs/development/tools/hot-reload>
- [16] Material Design
<https://flutter.dev/docs/development/ui/widgets/material>
- [17] Cupertino
<https://flutter.dev/docs/development/ui/widgets/cupertino>
- [18] Aplicações híbridas (consultado no dia 21 de janeiro de 2021.)
<https://ymedialabs.com/hybrid-vs-native-mobile-apps-the-answer-is-clear#part2>
<https://bloomidea.com/blog/aplicacoes-nativas-vs-hibridas-qual-escolher-para-o-seu-projeto>
<https://www.imaginarycloud.com/blog/native-vs-hybrid-vs-pwa-the-pros-and-cons/>
- [19] Repositório React-Native
<https://github.com/facebook/react-native>
- [20] Relatório do GitHub (consultado no dia 21 de janeiro de 2021.)
<https://octoverse.github.com/>
- [21] Statista gráfico uso das *frameworks* pelos desenvolvedores (consultado no dia 21 de janeiro de 2021.)
<https://www.statista.com/statistics/869224/worldwide-software-developer-working-hours/>
- [22] Website Stack Overflow
<https://pt.stackoverflow.com/>
- [23] Plataforma online Socratica
www.socratica.com

[24] Guião do Socrative (consultado no dia 17 de novembro de 2020)

http://edx.dge.mec.pt/asset-v1:ERTE+LA-FCL+LA-2016-2ed+type@asset+block/Tutorial_SOCRATIVE_MOOCedicao2.pdf

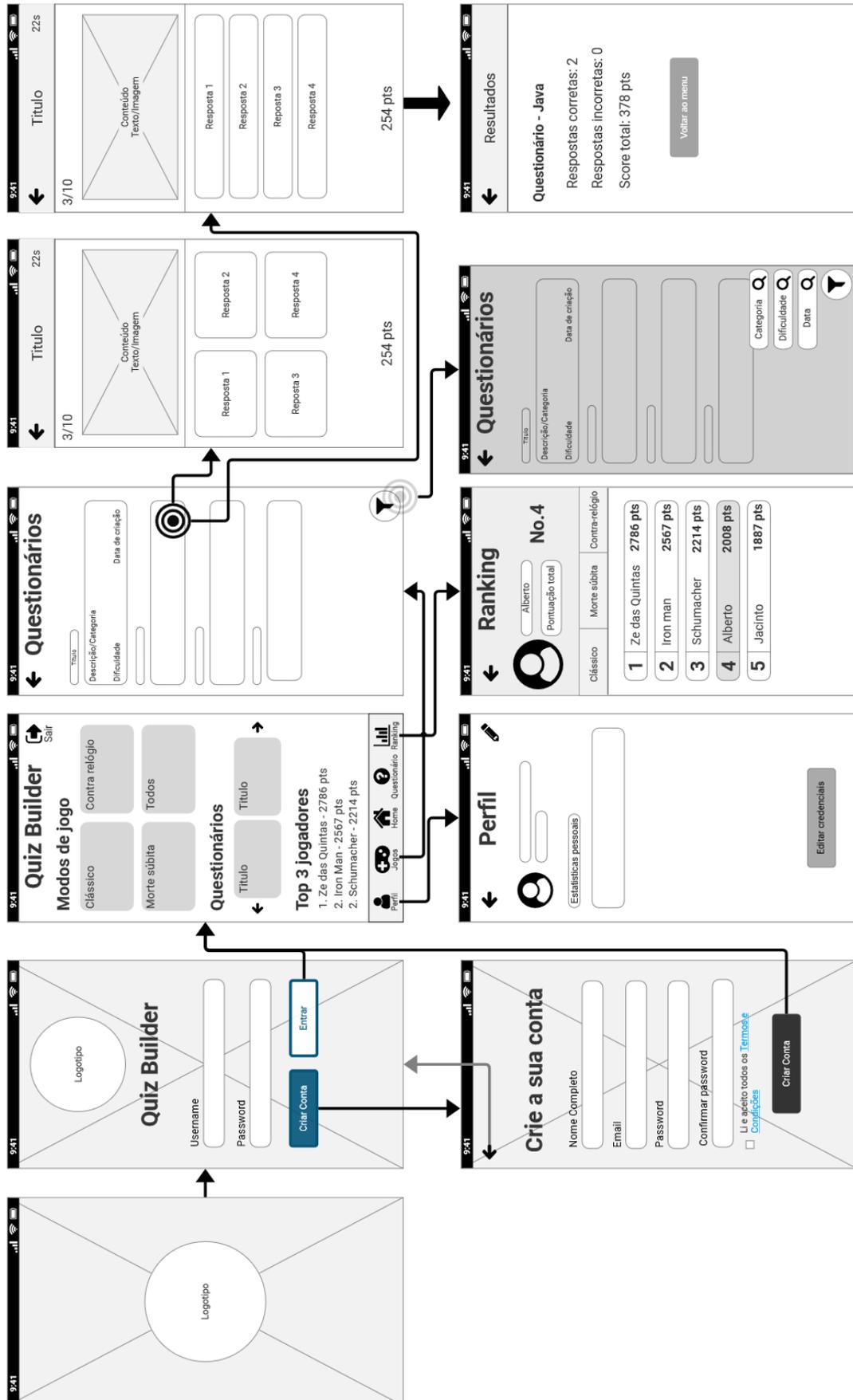
[25] Google Forms

<https://www.google.com/forms/about/>

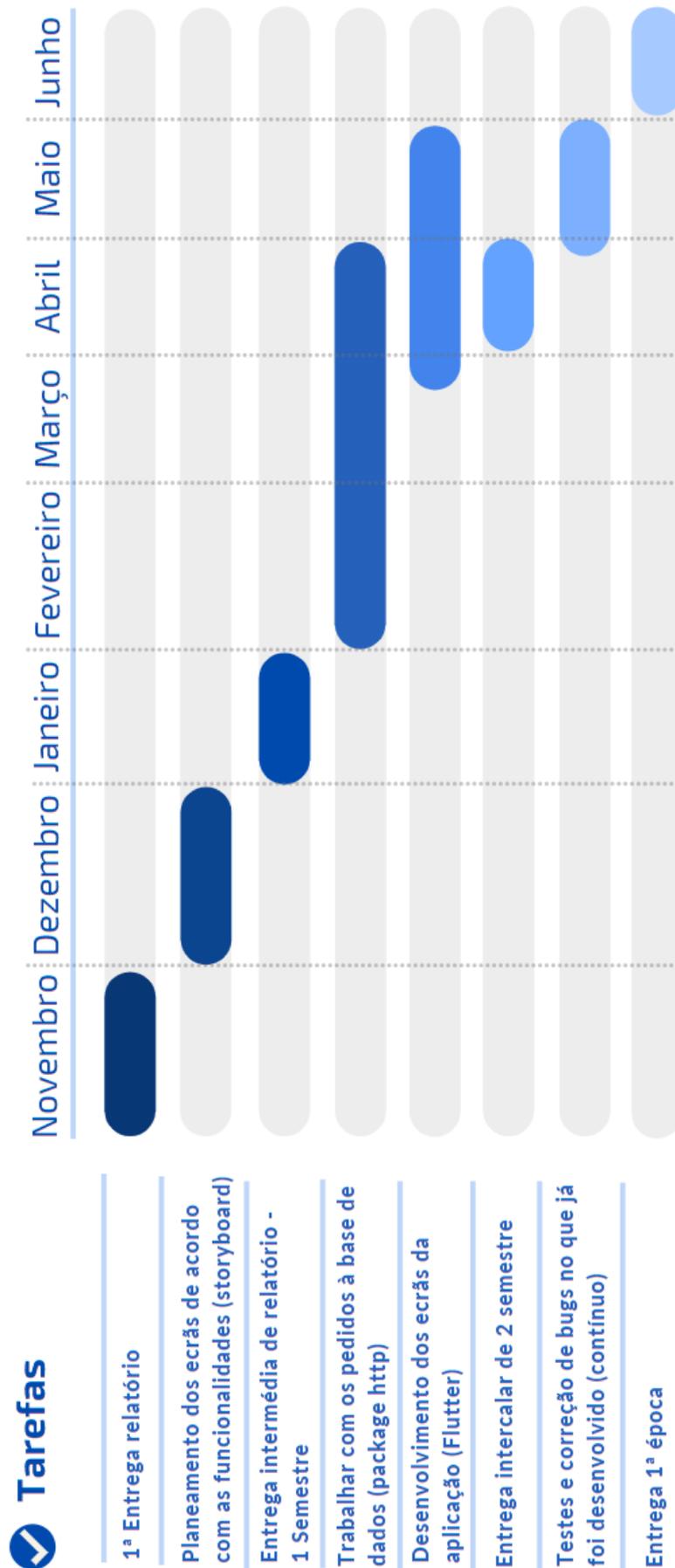
Glossário

LEI	Licenciatura em Engenharia Informática
TFC	Trabalho Final de Curso
ULHT	Universidade Lusófona de Humanidades e Tecnologias
DEISI	Departamento de Engenharia Informática e Sistemas de Informação
BLoC	Business Logic Component

Anexo - 1 – Storyboard



Anexo - 2 – Calendário



Tarefas

- 1ª Entrega relatório
- Planeamento dos ecrãs de acordo com as funcionalidades (storyboard)
- Entrega intermédia de relatório - 1 Semestre
- Trabalhar com os pedidos à base de dados (package http)
- Desenvolvimento dos ecrãs da aplicação (Flutter)
- Entrega intercalar de 2 semestre
- Testes e correção de bugs no que já foi desenvolvido (contínuo)
- Entrega 1ª época

Anexo 3 – Tarefas para o utilizador testante

Link para o guião de tarefas:

<https://forms.gle/NDT47zQEe7umoToTA>

Tarefa 1: Crie uma conta na aplicação.

Objetivo: Esta tarefa tem como objetivo, verificar se o utilizador consegue criar uma conta com sucesso. O utilizador conclui a tarefa com sucesso após receber um aviso indicando que a sua conta foi criada.

Tarefa 2: Efetue login com os dados relativos à conta previamente criada.

Objetivo: Verificar se o utilizador consegue iniciar sessão com a conta que foi criada previamente

Tarefa 3: Ao entrar na aplicação, depara-se com o ecrã inicial onde pode ver os diversos modos dos questionários, quantos modos de jogo existem?

Objetivo: Verificar se o utilizador consegue distinguir os diferentes modos de jogo.

Tarefa 4: Selecione o modo Clássico e responda a um dos questionários desse modo de jogo.

Objetivo: Confirmar que os questionários do modo clássico funcionam corretamente.

Tarefa 5: Volte para a página inicial e selecione o modo Contrarrelógio e responda a um dos questionários desse modo de jogo.

Objetivo: Confirmar que os questionários do modo clássico funcionam corretamente.

Tarefa 6: Repita o mesmo processo para o modo Morte-súbita e responda a um dos questionários desse modo de jogo errando a partir da segunda pergunta. Foi-lhe mostrada uma mensagem de derrota?

Objetivo: Confirmar que os questionários do modo morte-súbita acaba quando um utilizador falha uma das perguntas.

Tarefa 7: Volte ao ecrã inicial e selecione a opção “Ver Todos” ou a opção “Jogo” na barra de navegação na parte inferior da página e ordene os diferentes questionários por “Data de criação”, “Nome” e “Dificuldade”.

Objetivo: Testar a ordenação da lista dos diferentes questionários.

Tarefa 8: No ecrã inicial selecione a opção "Questionário" na barra de navegação na parte inferior da página e responda ao questionário "Controlo Qualidade". A seguir repita o processo. Foi-lhe mostrada uma mensagem a afirmar que já respondeu ao mesmo.

Objetivo: Garantir que os questionários do Modo Questionário apenas podem ser respondidos uma vez por utilizador.

Tarefa 9: No ecrã inicial selecione a opção "Ranking" na barra de navegação na parte inferior da página e verifique se consegue visualizar o seu nome de utilizador na tabela de pontuações dos diferentes modos de jogo.

Objetivo: Verificar o bom funcionamento do sistema de ranking.

Tarefa 10: No ecrã inicial selecione a opção "Perfil" na barra de navegação na parte inferior da página ou a imagem no canto superior direito verifique se consegue visualizar os seus dados no ecrã de perfil.

Objetivo: Testar o ecrã de perfil do utilizador.

Tarefa 11: Termine a sessão na aplicação.

Objetivo: Verificar se o utilizador sabe sair da sua conta.