



UNIVERSIDADE
LUSÓFONA

Ubiquous Quiz Builder

Trabalho Final de curso

Relatório Intercalar – 2 Semestre

Diogo Novo

Orientador: Miguel Tavares | Co-Orientador: Bruno Cipriano

Trabalho Final de Curso | LEI | 23/04/2020

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) pretende-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*, tem 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, através da tecnologia Flutter[1].

Abstract

With this Final Year Project, it is 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, has as main objective, 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 has as 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, through Flutter technology.

Índice

Resumo	iii
Abstract.....	iv
Índice	v
Lista de Figuras	vi
Lista de Tabelas	vii
1. Identificação do Problema.....	1
2. Viabilidade e Pertinência	2
3. Levantamento e análise dos Requisitos.....	4
4. Solução Proposta.....	8
4.1. Flutter.....	9
4.2. Arquitetura do Sistema	10
4.3. Storyboard.....	11
4.4. Protótipo.....	12
5. Benchmarking	13
5.1. Tecnologias.....	13
5.2. Aplicações no mercado	16
5.2.1. Socrative (Teacher & Student) Apps.....	17
5.2.2. QuizUp	18
6. Plano de testes e validação	20
6.1. Tarefas para os utilizadores	20
7. Calendário	23
Bibliografia.....	24
Glossário.....	27

Lista de Figuras

Figura 1 - Arquitetura do sistema.....	10
Figura 2 - Storyboard da aplicação móvel	11
Figura 3 - Prós e contras de aplicações nativas e híbridas	13
Figura 4 - Principais <i>frameworks</i> escolhidas por software <i>developers</i>	14
Figura 5 - Gráfico que compara o número de perguntas feitas do Stack Overflow.....	15
Figura 6 – Ecrã login.....	17
Figura 7 - Homepage	17
Figura 8 - Feedback de uma resposta incorreta	17
Figura 9 - Feedback de uma resposta correta.....	17
Figura 10 - Ecrã pergunta com imagem.....	17
Figura 11 - Ecrã dos resultados geral.....	18
Figura 12 - Ecrã dos resultados detalhado	18
Figura 13 - Ecrã da categoria de carros	19
Figura 14 - Ecrã de escolha da resposta	19
Figura 15 - Feedback interativo da opção escolhida	19
Figura 16 - Esquema de calendarização do desenvolvimento do TFC	23

Lista de Tabelas

Tabela 1 - Requisitos funcionais	6
Tabela 2 - Requisitos não funcionais.....	6

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 (este tema será abordado no capítulo Viabilidade e Pertinência).

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

2. 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, em específico 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].

3. 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:

- Não implementado
- Inicial - Já tem algumas componentes programáticas desenvolvidas
- Intermédio - Tem o funcionamento base implementado e alguma interação com a *user interface*
- Implementado - Implementado, porém sujeito a alterações na interface gráfica e funcional

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>	Inicial
RF2	Efetuar login	Permitir que um utilizador previamente registado faça <i>login</i> introduzindo o nome de utilizador e <i>password</i> .	Implementado
RF3	Listar questionários por modo de jogo	Mostrar ao utilizador autenticado, quais os questionários ativos nos diferentes modos de jogo	Implementado

RF4	Listar questionários por categoria	Mostrar ao utilizador autenticado, quais os questionários que pertencem a uma certa categoria	Implementado
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 	Inicial
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	Intermédio
RF7	Modo de jogo contrarrelógio	Modo em que o questionário tem um tempo total para ser respondido e não um tempo por pergunta	Inicial
RF8	Modo de jogo morte súbita	Modo em que o questionário termina mal o utilizador erre uma pergunta	Inicial
RF9	Modo questionário	Modo que apenas permite responder uma única vez aos questionários incluídos no mesmo	Não implementado
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.	Inicial

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 	Não implementado
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.	Não implementado
RF13	Imagens nos questionários	Permitir perguntas nos questionários que venham acompanhada por uma imagem fazendo upload da mesma na criação do questionário	Não implementado

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

Por fim no *backoffice*, onde pode ser feita toda a gestão dos questionários e dos utilizadores, gostaria de melhorar a segurança nos campos de preenchimento prevenindo o tão conhecido “SQL *injection*” que consiste na inserção de código SQL (*Structured Query Language*) por meio dos campos de preenchimento o que pode causar danos graves na base de dados ou a perda da privacidade dos mesmos. Durante a minha familiarização ao projeto, identifiquei problemas relativos ao pouco *feedback* que recebemos pelas mensagens de erro. 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. Portanto seria bom melhorar isso de maneira a tornar a plataforma mais *user friendly* (amiga do utilizador).

4. Solução Proposta

Neste trabalho pretende-se não só replicar a aplicação existente, mas também melhorá-la, tal como já foi citado, disponibilizando-a também para o sistema operativo *iOS*. Isto implica que a nova aplicação disponibilize 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
- Acesso ao modo de questionário (só pode ser respondido uma vez por utilizador)
- Sistema de filtragem e ordenação dos questionários
- *Feedback* das respostas
- Alerta de nova *High Score* (Pontuação Máxima)
- Sistema de *ranking*

No entanto, se o tempo nos permitir gostaríamos de implementar novas funcionalidades para além das que falamos, para tornarmos os questionários mais interativos (temos alguns exemplos na parte do 13). Tais como:

- Dar uma aparência mais apelativa e moderna à aplicação
- Melhorar o *feedback* das respostas
- Permitir que os questionários tenham imagens nas perguntas

4.1. 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:

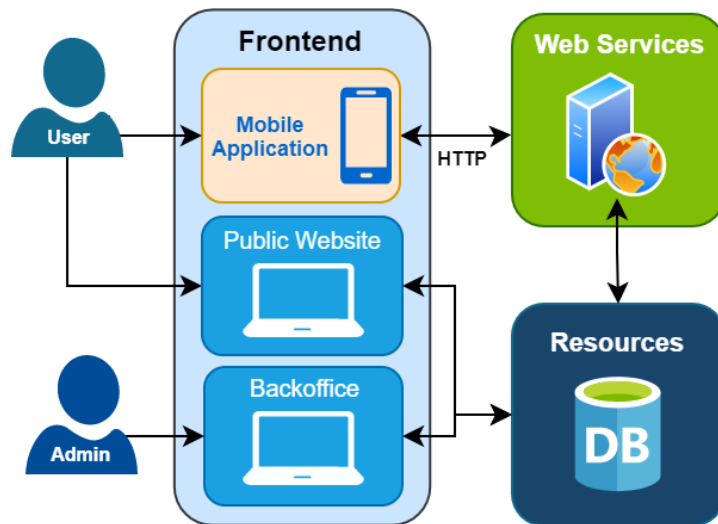
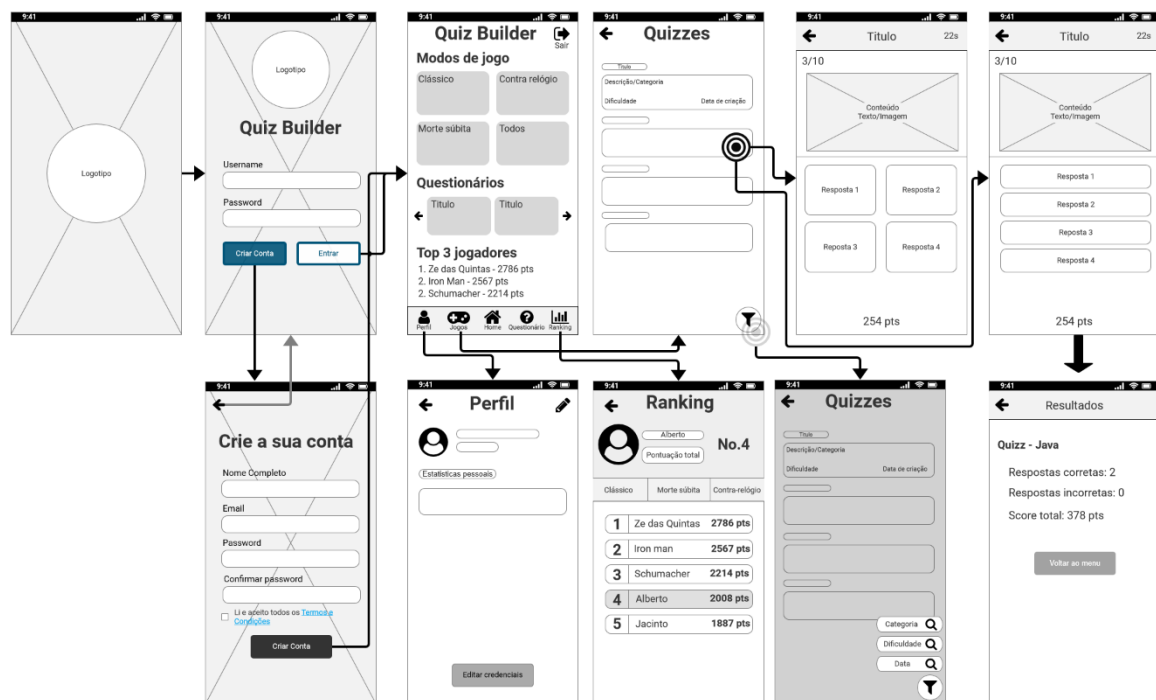


Figura 1 - Arquitetura do sistema

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.



4.4. Protótipo

Desde a última entrega, o foco tem vindo a ser no desenvolvimento da aplicação e 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 *webservices* do tipo REST, pois estes seriam a base para ir buscar toda a informação que compunham 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, iniciei o desenvolvimento da User Interface, tendo desenvolvido até ao momento:

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

Nota: Todos estes estão sujeitos a alterações tanto gráficas como funcionais.

No que toca aos modos de jogo, o que está implementado é o modo clássico, por isso ainda faltam implementar os outros dois que são praticamente uma variação do clássico.

Um dos pontos que tenho levado em conta durante o desenvolvimento da aplicação além do bom funcionamento, é dar um aspeto mais moderno à mesma. Para facilitar a visualização do progresso desde a última entrega, foi feito um vídeo que mostra os diversos ecrãs e funcionalidades atualmente implementadas que fica disponibilizado abaixo em conjunto com o respetivo repositório do *github*.

Link do vídeo demonstrativo:

<https://youtu.be/MMza2qZrjus>

Link do repositório da aplicação:

[GitHub - diogo19o/Ubiquous-Quizz-Builder: Ubiquous Quizz Builder](https://github.com/diogo19o/Ubiquous-Quizz-Builder)

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 3 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 3 - 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.

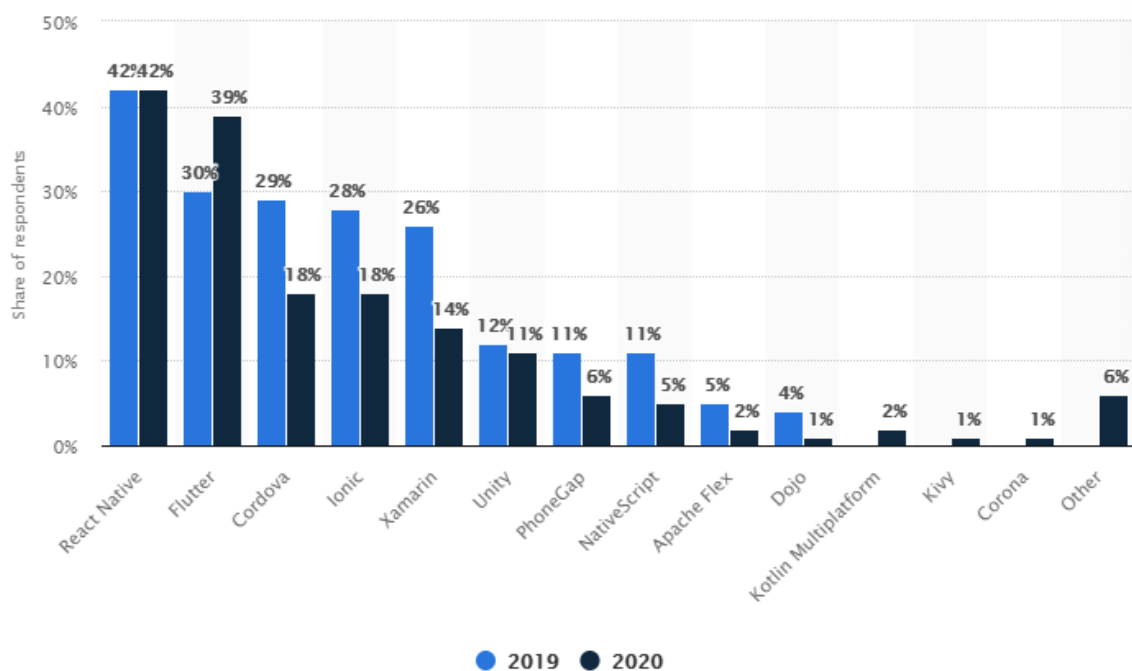


Figura 4 - Principais *frameworks* escolhidas por *software developers*

Outro aspeto notável relativamente ao Flutter tem vindo a ser o aumento da sua comunidade. Como podemos ver no gráfico da Figura 5, 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*.

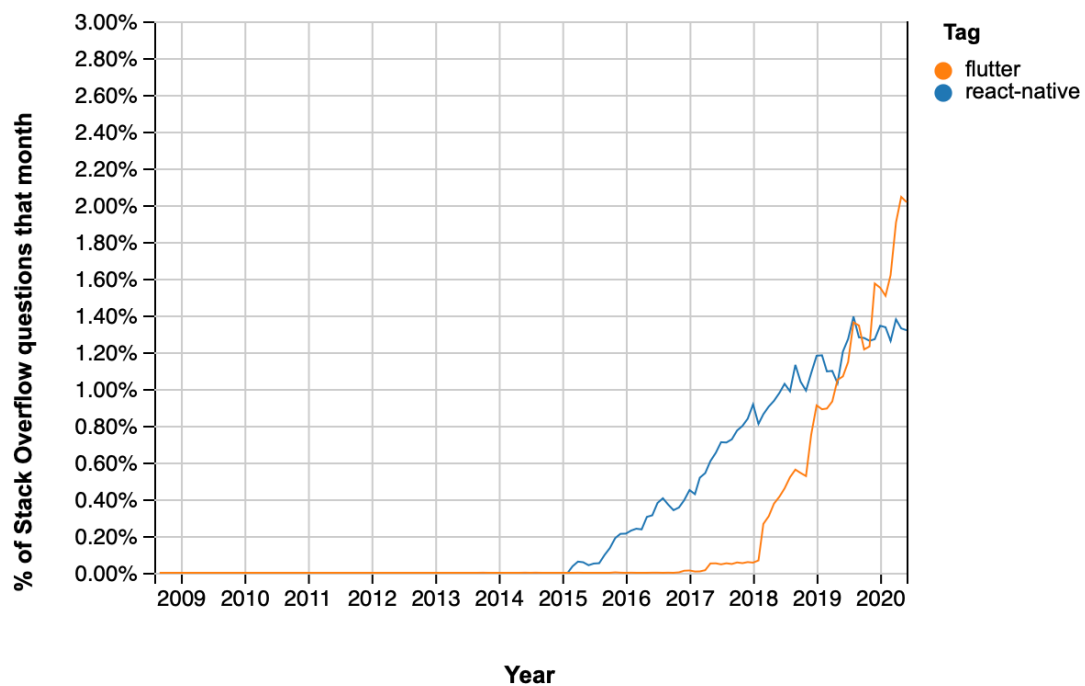


Figura 5 - Gráfico que compara o número de perguntas feitas do Stack Overflow[22]

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:

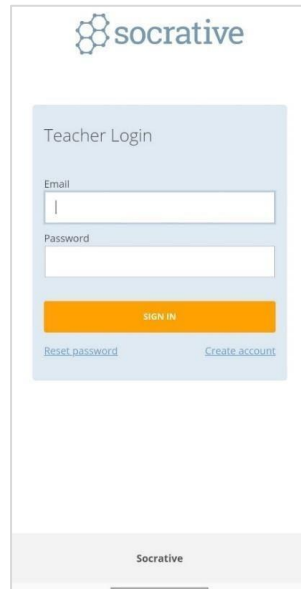


Figura 6 – Ecrã login

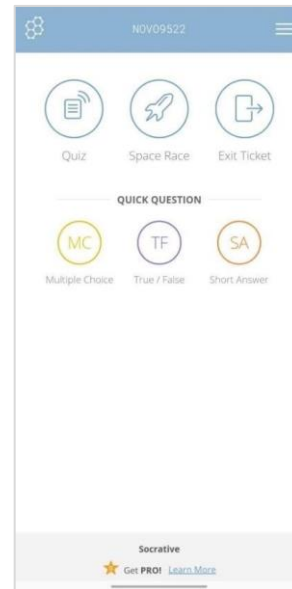


Figura 7 - Homepage

- 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 10 - Ecrã pergunta com imagem

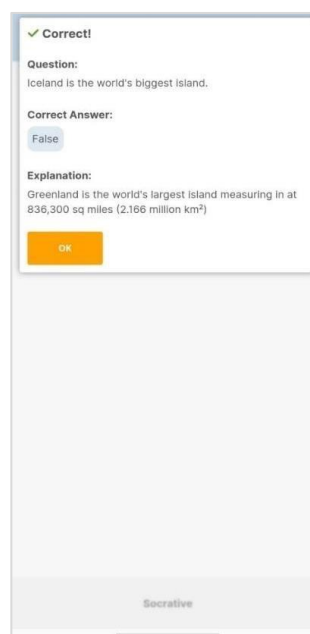


Figura 9 - Feedback de uma resposta correta

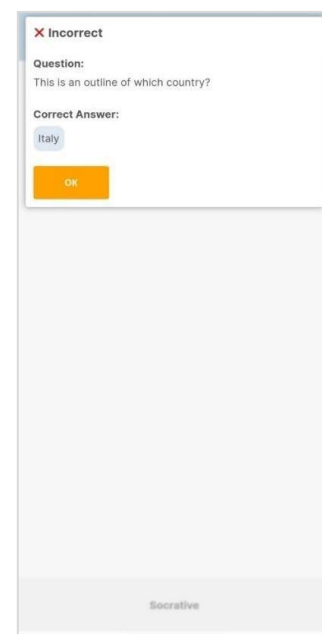


Figura 8 - Feedback de uma resposta incorreta

Outra funcionalidade muito interessante, que pode ser vista na Figura 10 acima é o suporte de imagens nos questionários que é um requisito para ser implementado no futuro.

- Visualização dos resultados dos questionários:

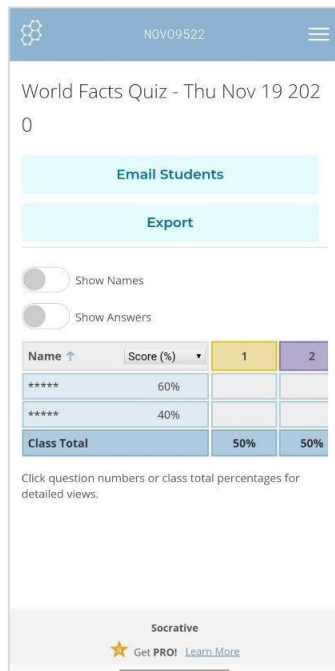


Figura 11 - Ecrã dos resultados geral

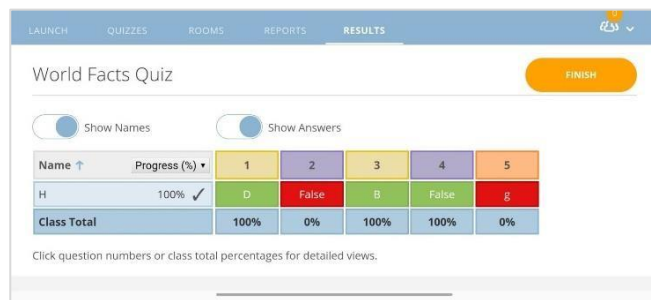


Figura 12 - Ecrã dos resultados detalhado

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:

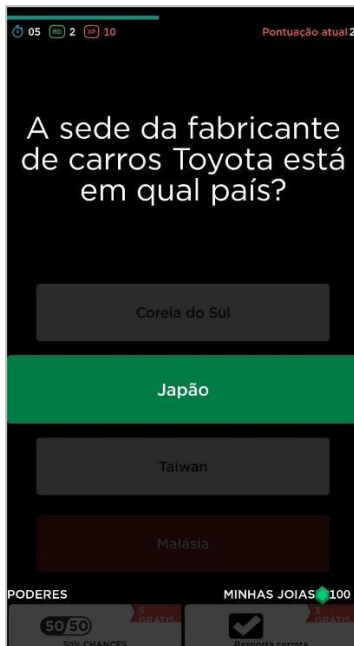


Figura 15 - Feedback interativo da opção escolhida



Figura 14 - Ecrã de escolha da resposta



Figura 13 - Ecrã da categoria de carros

Como podemos visualizar, a Figura 13 mostra algumas informações sobre a categoria em questão, mas gostaria de destacar a Figura 14 e a Figura 15, onde nos mostra a interface na resposta ao questionário. Em cima podemos verificar que temos um cronometro normal, mas também temos um objeto gráfico (barra verde) no topo que nos mostra o tempo que nos resta.

Por fim, a Figura 15 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. Plano de testes e validação

Com o intuito de testar as várias funcionalidades da aplicação, encontrar bugs e verificar se a mesma funciona de forma concisa e clara, cumprindo assim, os seus requisitos, foram desenvolvidas diversas tarefas que serão fornecidas aos utilizadores.

6.1. Tarefas para os utilizadores

Para resolver estas tarefas, planeia-se a realização de testes com um grupo de utilizadores reais. 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:

<https://forms.gle/NDT47zQEe7umoToTA>

Tarefas:

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.

7. Calendário

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

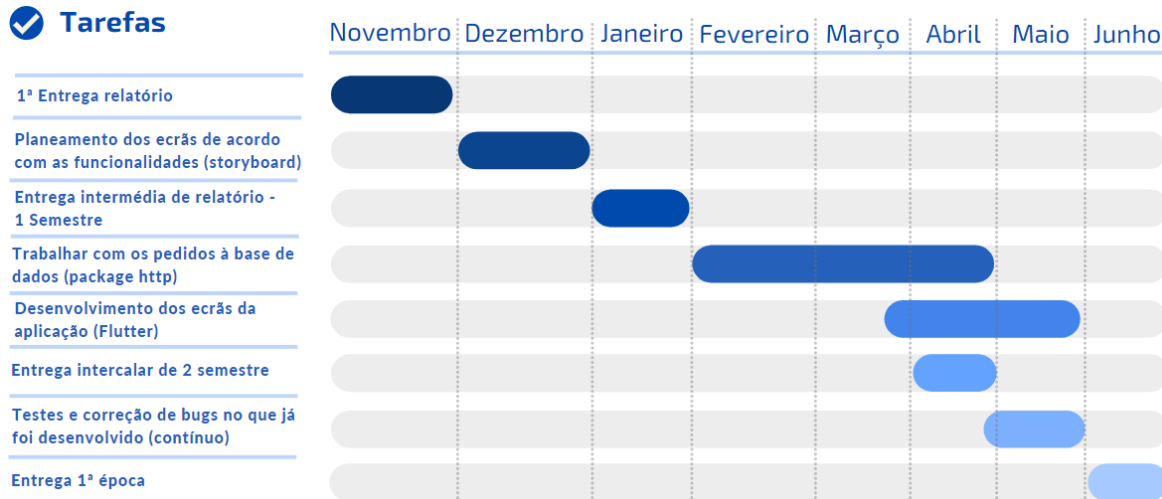


Figura 16 - Esquema de calendarização do desenvolvimento do TFC

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 funcionará a navegação entre eles.

Depois, um pouco mais tarde do que planeado, por volta comecei por aprender e explorar mais sobre como funcionavam os *webservices* no Flutter e foi um processo relativamente demorado. Isto acabou por fazer com que se atrasasse um pouco o desenvolvimento da *user interface*, que comecei a desenvolver entre o fim de março e início de abril.

A partir de agora em diante tendo de todos os dados praticamente feita, resta dar vida aos mesmos por meio da interface gráfica. Esses são os planos para agora, desenvolver os ecrãs que faltam como o de *Rankings* e Perfil, e o resto da implementação dos modos dos questionários.

No início maio gostaria de começar uma fase de testes à aplicação com a ajuda de um grupo de utilizadores e corrigir problemas e bugs funcionais. A partir daí aproxima-se a entrega final entrega e esse será o foco.

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 Socrative
www.socrative.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

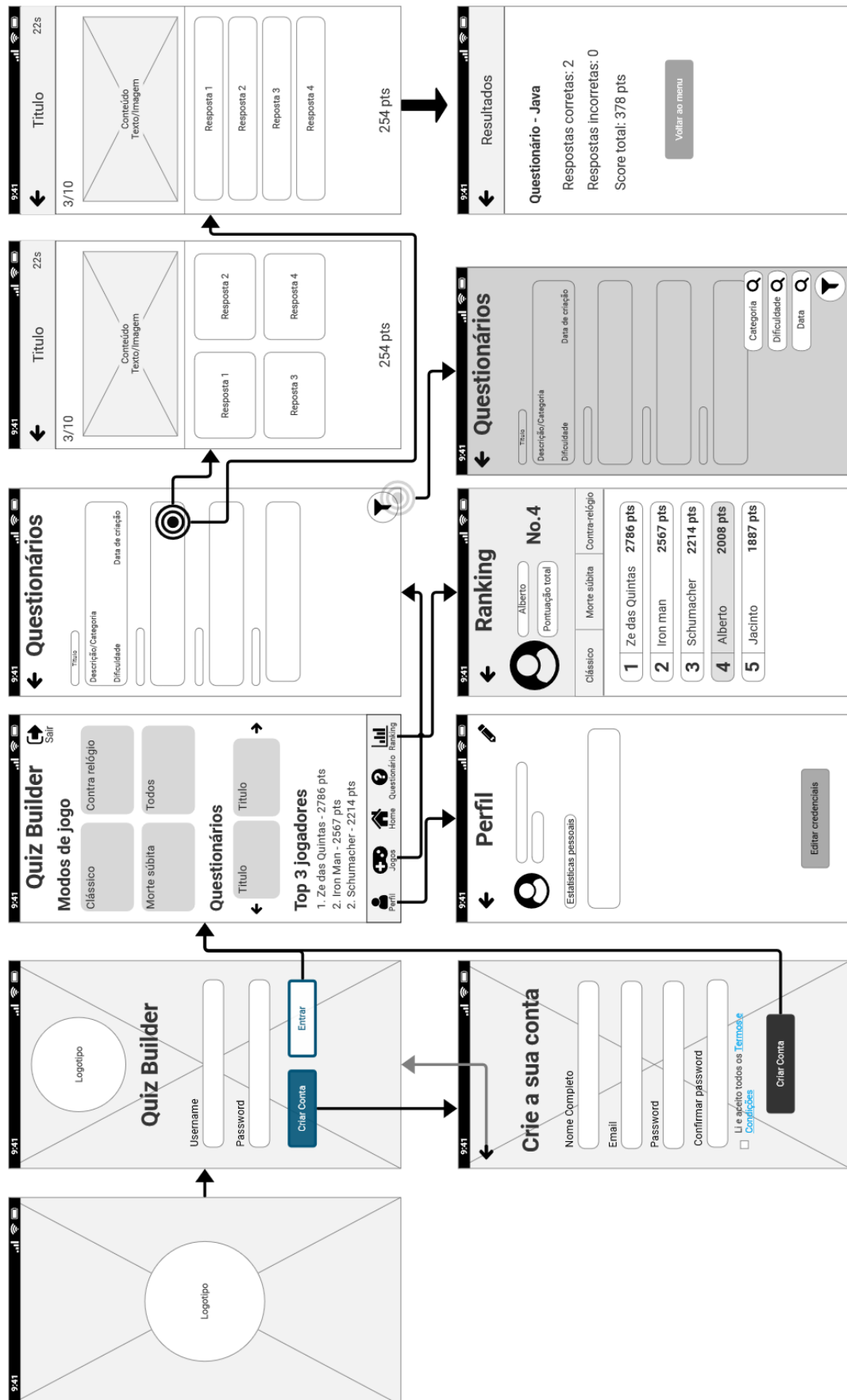
[25] Google Forms

<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

Anexo - 1 – Storyboard



Anexo - 2 – Calendário

