



ISEL – INSTITUTO SUPERIOR DE ENGENHARIA DE LISBOA
ADEETC – ÁREA DEPARTAMENTAL DE ENGENHARIA DE
ELECTRÓNICA E TELECOMUNICAÇÕES E DE COMPUTADORES

LEIM
LICENCIATURA EM ENGENHARIA INFORMÁTICA E MULTIMÉDIA
UNIDADE CURRICULAR DE PROJETO

Speedy Manager #24



Rui Inácio (45109)

João Folgado (45128)

Orientador

Professor Doutor António Teófilo

Julho, 2021

Resumo

Uma nota a falar sobre a necessidade deste sistema.

O Speedy Manager é uma aplicação movel que permite a organização de tarefas e listas de tarefas. Esta lista de tarefas pode ser partilhada e manipulada por vários utilizador em simultâneo.

O principal objetivo é dispor de uma interface amigável para o utilizador, mas que consiga incorporar diferentes elementos de aplicações do mesmo género tais como sistema de chat entre utilizadores, partilha de conteúdos multimédia nas tarefas e uma interface atualizada em tempo real com alterações feitas por outros utilizadores.

A aplicação foi contruída em Kotlin, uma linguagem de programação multiplataforma com interoperabilidade total com Java, e que é referencia no desenvolvimento de aplicações moveis. Alem do Koltin a aplicação integra o sistema de Firebase da google que permite que os utilizadores utilizem a sua conta google na aplicação e ainda, este sistema, dispõe ao programador de ferramentas tais como armazenamento em cloud, base de dados em tempo real, autenticação de utilizadores, entre outros.

Motivação

A motivação para a realização desta aplicação provem do facto de, embora existirem várias aplicações com o mesmo intuito, muitas delas possuem elementos/funcionalidades distintas que podem ser reunidas numa só aplicação. Ainda que a partilha entre utilizadores da organização de listas/tarefas possa ser encontrada em várias aplicações deste género de renome, notamos que, estas muitas vezes não possuem funcionalidades básicas como partilha de multimédia e conversação em tempo real ou suporte multi-plataforma. O Speedy Manager visa então entregar ao utilizador uma aplicação simples de se manusear, mas que disponha de várias ferramentas e funcionalidades que enriquecem a experiência na organização de tarefas.

Agradecimentos

Especial agradecimento ao Professor Doutor António Teófilo orientação
que nos ofereceu durante a realização deste projeto

Índice

Resumo	i
Motivação	iii
Agradecimentos	v
Índice	vii
Lista de Tabelas	ix
Lista de Figuras	xi
1 Introdução	1
2 Trabalho Relacionado	3
3 Modelo Proposto	5
3.1 Requisitos	5
3.2 Fundamentos	6
3.3 Abordagem	7
4 Implementação do Modelo	11
4.1 Termos Técnicos	11
4.1.1 Google FireBase	11
4.2 Fase inicial	12
4.2.1 Realização de wireframes	12
4.2.2 Realização de Mockups	13
4.2.3 Obtenção de feedback para Design proposto	15
4.2.4 Análise dos resultados obtidos no questionário	16

4.2.5	Análise dos resultados dos questionários	16
5	Validação e Testes	17
6	Conclusões e Trabalho Futuro	19
A	Um Detalhe Adicional	21
B	Outro Detalhe Adicional	23
	Bibliografia	25

Lista de Tabelas

3.1	Tabela de funcionalidades	6
5.1	Uma tabela	17

Lista de Figuras

3.1	Firebase	7
3.2	Kotlin	7
4.1	Wireframes Ecrãs de login e registo	12
4.2	Wireframes Ecrãs principais bottom bar navigation	13
4.3	Wireframes Ecrãs dentro de uma lista	13
4.4	MockUps Ecrãs de login e registo	14
4.5	MockUps Ecrãs principais bottom bar navigation	15
4.6	MockUps Ecrãs dentro de uma lista	15
5.1	Uma figura	17

Capítulo 1

Introdução

Capítulo 2

Trabalho Relacionado

Como esta aplicação tem como principal objetivo permitir aos seus utilizadores, através de uma interface gráfica simpática, uma fácil gestão de tarefas, fizemos um estudo do mercado com o objetivo de ver e comparar as diferentes aplicações que oferecem o mesmo serviço.

Esse estudo de mercado serviu não só para ter uma vaga ideia de como são as interfaces gráficas dos concorrentes bem como comparar as diferentes funcionalidades.

O nosso estudo focou principalmente nas três seguintes aplicações, [GoogleTasks, 2018], [MicrosoftToDo, 2017] e por fim [Trello, 2011].

Após comparação destas chegámos à conclusão de que, apesar de cada uma ter as suas vantagens, o Trello era aquela que mais se aproximava do que pretendemos realizar.

Capítulo 3

Modelo Proposto

3.1 Requisitos

Esta aplicação tem como requisitos funcionais como permitir a criação e gestão de listas de tarefas, o uso de uma cloud database que permita sincronização em tempo real entre os vários utilizadores, partilha de listas entre os mesmos, uma interface amigável e intuitiva.

De modo a perceber melhor o que o mercado oferece no que toca a concorrentes a esta aplicação decidimos comparar as funcionalidades entre várias aplicações semelhantes à nossa, nomeadamente entre “Microsoft To Do”, “Google Tasks”, e o “Trello”.

Após essa investigação conseguimos ter uma melhor perceção dos concorrentes diretos bem como ir buscar alguma inspiração às diversas aplicações acima referidas, as quais podem ser comparáveis pela tabela abaixo.

Features	Trello	Microsoft To Do	Google Tasks	Speedy Manager
Friendly, accessible, intuitive interface	✓	✓	✗	✓
Task creation	✓	✓	✓	✓
Creating Task Lists	✓	✓	✓	✓
Swipe commands to complete tasks	✗	✗	✓	✓
Swipe commands to delete tasks	✗	✓	✗	✓
Tasks with colors	✓	✗	✗	✓
Tasks with images	✓	✗	✗	✓
Tasks with files (txt, pdf, images)	✓	✓	✗	✓
Colored task lists	✓	✓	✗	✓
Task lists with default application images	✓	✓	✗	✓
Task lists with images of the user's choice	✓	✗	✗	✓
Task lists with task lists inside	✓	✓	✓	✓
Progress bar within task lists	✓	✗	✗	✓
Chat function	✓	✗	✗	✓
Shared task lists	✓	✓	✗	✓
Real-time synchronization	✓	✓	✗	✓
Drag tasks up and down	✓	✓	✓	✓
Tasks organized by date	✓	✓	✓	✓
Tasks organized by priority	✓	✓	✗	✓
Calendar with tasks to be completed (independent of the list)	✓	✓	✗	✓

Tabela 3.1: Tabela de funcionalidades

3.2 Fundamentos

Sendo o Speedy Manager uma aplicação com o objectivo, acima de tudo, de ser algo prático, eficiente e de rápido uso decidimos utilizar a plataforma Firebase como base de dados, devido a esta dispor de funcionalidades benéficas para real-time updates, nomeadamente com “Real-time database”.

Iremos também utilizar as funcionalidades de “Authentication” para gestão de utilizadores, “Storage” para guardar todos os ficheiros carregados pelos utilizadores, e “Hosting” para gerir da vertente multiplataforma.



Figura 3.1: Firebase

Como linguagem de programação decidimos utilizar o Kotlin com a ambição de tornar o Speedy Manager uma aplicação multiplataforma, pois esta linguagem dispõe funcionalidades que irão facilitar essa implementação.



Figura 3.2: Kotlin

3.3 Abordagem

Tendo em conta que o Speedy Manager tem como principal objetivo ser uma aplicação que permite uma rápida e intuitiva gestão de tarefas e de listas de tarefas e a partilha destas rapidamente começamos a pensar nas tecnologias que mais facilmente iriam permitir alcançar esse objetivo.

Para tal tivemos em consideração as linguagens de programação Java e Kotlin para desenvolver a nossa aplicação android bem como algumas opções de tecnologias para serem utilizadas como base de dados tais como o Firebase e o SQL. Após pesquisa sobre ambos os temas chegámos à conclusão de que Kotlin seria a linguagem mais vantajosa devido às suas capacidades para multiplataforma, bem como o Firebase devido à sua capacidade de funcionamento em real-time utilizando a sua funcionalidade “Real-time database”, o qual seria benéfico para nós para permitir sincronização em real time entre os vários utilizadores.

Após termos decidido que tecnologias utilizar começamos a realizar esboços em papel de qual seria o aspeto da nossa aplicação, ou seja, as wireframes, sendo estas uma representação técnica de cada ecrã da nossa aplicação android. No fim de concluir estes wireframes, utilizando a fer-

ramenta figma implementamos esses layouts e criamos uma demonstração (uma demo) da aplicação a qual foi enviada para alguns utilizadores para ser testada, bem como um questionário sobre melhorias a fazer, pontos fortes ou outras observações de utilizadores que achassem pertinentes.

De seguida começamos a implementar as principais mecânicas da aplicação, de um modo simplificado, começamos primeiramente por implementar um sistema de login, tanto com email/pass como com a conta da google de modo a conseguir já distinguir os diversos utilizadores da nossa aplicação, achamos este um ponto de partida inteligente pois a aplicação funciona em torno de listas de tarefas que cada utilizador criou ou é parte de.

Seguidamente implementamos o resto das funcionalidades apresentadas nos wireframes, de um modo básico ainda, recorreremos à utilização de RecyclerViews com uso de ItemTouchHelpers para permitir as funcionalidades de drag and drop bem como as funcionalidades de swipe right e left. Estas funcionalidades foram implementadas utilizando interfaces muito básicas sem grande detalhe estético utilizando ficheiros XML para definir estas.

Na seguinte etapa fomos avaliar outras opções para implementar as interfaces gráficas referidas no paragrafo acima, as quais encontramos o Jetpack que oferece a ferramenta “Compose” para a fácil e estética criação de interfaces gráficas e começamos a desenvolver os wireframes utilizando esta nova ferramenta, o que foi trabalho em vão pois após uma investigação mais profunda reparamos que o Jetpack não oferece ferramentas equivalentes às RecyclerViews e ao ItemTouchHelper, os quais são duas das principais ferramentas da nossa aplicação pois vão ser estas as responsáveis pelo mecanismo de comportamento das tarefas dentro de uma lista, e no fim decidimos voltar à abordagem utilizando o XML.

Posteriormente à implementação das wireframes decidimos avançar para a realização das mockups da nossa aplicação, novamente utilizando a ferramenta figma, com a qual realizamos uma “demo” que simulava a utilização da aplicação, enviamos essa demo bem como mais um questionário sobre a utilização desta de modo a obter algum feedback para melhorar/implementar funcionalidades que os utilizadores tivessem achado pertinentes.

Tendo em conta os resultados dos questionários, passámos à implementação destes mockups, utilizando o XML para definir os layouts, na nossa

aplicação de modo a já ter uma versão esteticamente apelativa desta.

Após esta implementação dos mockups chegou a fase de implementar as funcionalidades principais de um modo mais específico, nomeadamente a criação de listas de tarefas, associar imagens a listas ou a tarefas, mudar a cor da lista, editar textos, apagar tarefas, completar tarefas, alterar a data-limite de uma tarefa entre outras, de modo a cumprir todas as funcionalidades anunciadas anteriormente.

Quando a mecânica de funcionamento das listas de tarefas estava completa passamos à fase de criar um sistema de contactos onde o utilizador pode adicionar um contacto aos seus amigos, e posteriormente se quiser, adicioná-lo a uma lista, isto de modo a permitir a partilha de listas de tarefas entre diversos utilizadores.

Com este sistema implementado conseguimos então passar à fase onde testámos a funcionalidade em real-time da nossa aplicação, criando uma lista e convidando um utilizador conseguimos testar o funcionamento de uma lista para 2 utilizadores em simultâneo e corrigir possíveis erros nesta fase. Por fim implementamos as funcionalidades de carácter menos principal da nossa aplicação, nomeadamente um chat por lista, onde os utilizadores dentro de uma lista podem trocar mensagens uns com os outros, bem como uma dashboard na qual são apresentados os eventos que aconteceram de maneira mais recente com uns gráficos referentes a quem participa mais ou a quem completa mais tarefas. Implementamos também na página inicial um display que mostra os 3 lobbies recentemente visitados pelo utilizador, bem como outro display onde mostra as tarefas, de qualquer lista, que estão perto de expirar ou que a data limite está a chegar, de modo a dar jus ao nome “Speedy Manager”.

Capítulo 4

Implementação do Modelo

Implementação do modelo aqui . . . ; pode precisar de referir o capítulo 3 . . .

Aqui identifica as opções teóricas e justifica as dependências tecnológicas assumidas neste projecto. Descreva com rigor formal e detalhe adequado e faça evidência de tudo o que foi proposto e desenvolvido especificamente no contexto deste projecto. Aqui a ênfase está naquilo que foi de facto concretizado neste projecto.

O leitor quer detalhes de concretização. Ele já está enquadrado no tema (cf., capítulo 2), já conhece os aspectos mais abstractos do sua proposta (cf., capítulo 3) e agora precisa de entender os detalhes para conseguir também interpretar as validações e testes que posteriormente (cf., capítulo 5) lhe irá apresentar.

4.1 Termos Técnicos

4.1.1 Google FireBase

A [Moroney, 2017] Firebase é uma plataforma baseada em Cloud que foi adquirida e por sua vez distribuída pela Google destinada a auxiliar o desenvolvimento de aplicações moveis e para a web.

Os serviços disponibilizados por esta plataforma permitem, a quem a implemente na sua aplicação, funcionalidades tais como um sistema de base de dados em tempo real, sistema de armazenamento de ficheiros (á parte da base de dados), autenticação de utilizadores , web hosting utilizando

uma rede de fornecimento global, entre outras ferramentas de auxílio ao desenvolvimento e controlo da aplicação tais como análise e monitorização dos utilizadores/dados da aplicação.

A Firebase é, portanto, um serviço muito completo prestado pela Google no auxílio ao desenvolvimento de aplicações em Android e também para a web, e por tal será a plataforma que constitui o “backend” da nossa aplicação e, portanto, assume o papel de servidor.

4.2 Fase inicial

4.2.1 Realização de wireframes

Nas fases iniciais da resolução deste projeto começamos por perceber o contexto da aplicação que iríamos desenvolver. Para tal exploramos aplicações que abordam o mesmo contexto/temática para não só adquirirmos a experiência de utilização, mas também, para fazer um “estudo de mercado” ou simplesmente ver o que existe feito.

Exploramos aplicações tais como o Google Tasks, Trello, Samsung notes, Microsoft todo e também pesquisamos sobre outras relacionadas. Posto isto iniciamos o processo de desenvolvimento da aplicação segundo metodologias que achamos importantes adotar.

Este processo consistiu na discussão de ideias entre vários intervenientes de modo a ser produzido um esquema visual básico contendo os requisitos e funcionalidades da aplicação ou seja, a realização das wireframes.

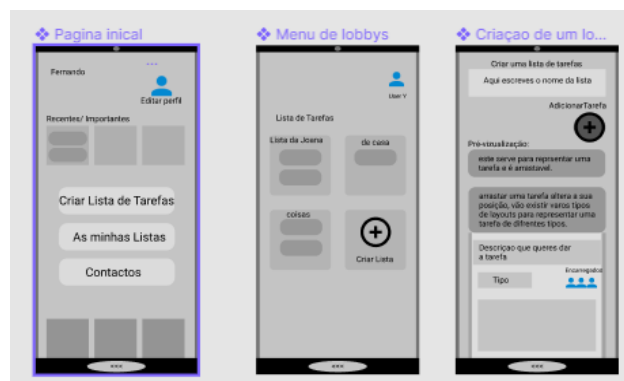


Figura 4.1: Wireframes Ecrãs de login e registo

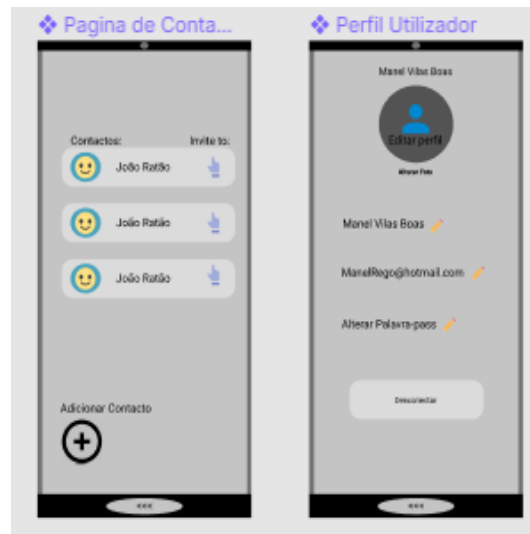


Figura 4.2: Wireframes Ecrãs principais bottom bar navigation



Figura 4.3: Wireframes Ecrãs dentro de uma lista

4.2.2 Realização de Mockups

Após a construção destes wireframes recorremos à criação de uma breve "demo", utilizando a ferramenta Figma de modo poder ser enviada para utilizadores para estes a testarem, foi ainda enviado também um questionário de usabilidade com o objetivo de obter algum feedback da demonstração e

alguns pontos que os utilizadores tivessem achado de interesse melhorar.

Com base nas wireframes e nas respostas dos questionários passámos à fase de desenvolvimento das mockups, sendo que estas já são uma versão mais detalhada e muito próximo de como a aplicação irá parecer, de um ponto de vista estético.

Realizamos também de novo uma demonstração para ser enviada com um questionário de usabilidade de modo a obter feedback e poder melhorar eventuais pontos que os utilizadores tenham achado mais fracos.

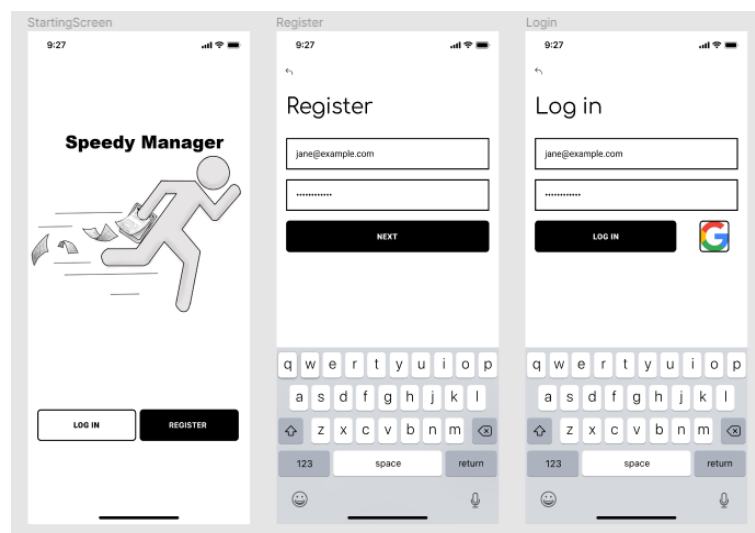


Figura 4.4: MockUps Ecrãs de login e registo

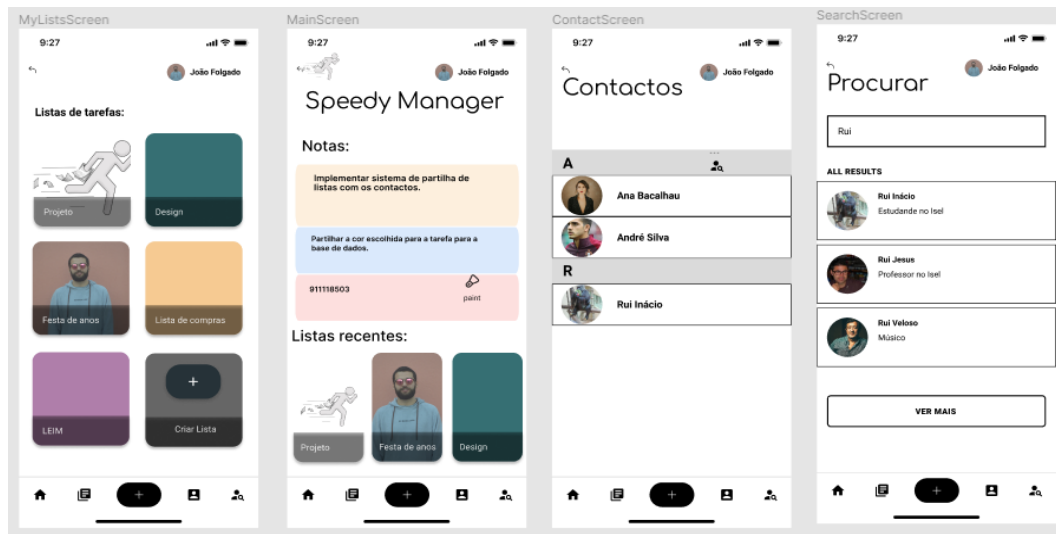


Figura 4.5: MockUps Ecrãs principais bottom bar navigation

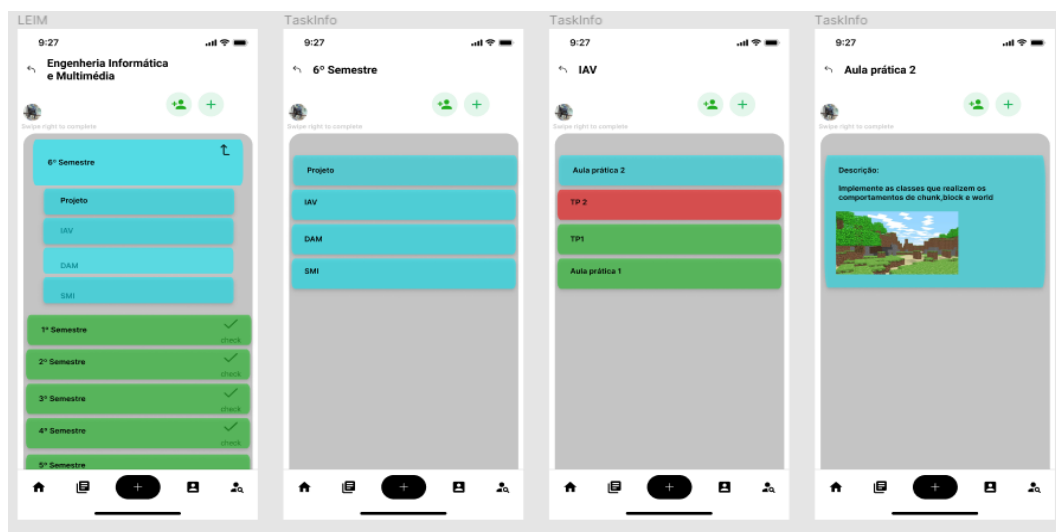


Figura 4.6: MockUps Ecrãs dentro de uma lista

4.2.3 Obtenção de feedback para Design proposto

Para avaliarmos as ideias/conceitos relacionados com projeto bem como este progresso no desenvolvimento de modelos visuais para a representação das mesmas, criamos um questionário com 10 perguntas que achamos pertinentes colocar ao publico que á partida sabíamos que iria responder.

Link para o questionário: <https://docs.google.com/forms/d/1v09wkguetqlYwzhXG3oQkmllL2cGsBD73wPfU/edit>

//// IMAGENS DO QUESTIONARIO EM MODO RESPOSTA

4.2.4 Analize dos resultados obtidos no questionario

METER IMAGENS DOS RESULTADOS DOS QUESTIONARIOS

A analize deste resultados sera feita no capitulo seguinte onde será feita uma avaliação das decisões tomadas nesta faze.

METER IMAGENS DOS DADOS BEM COMO DO QUESTIONARIO E A GRAFICOS DAS RESPOSTAS

4.2.5 Análise dos resultados dos questionários

Após a análise de resultados e algumas trocas de ideias e opiniões foram feitos algumas alterações de alguns elementos dos Mockups desenvolvidos nesta faze. Alem disto foram também feitas outras alterações ao longo do desenvolvimento da aplicação, alterações estas fundamentadas com melhores análises sobre o propósito de determinados pontos da aplicação.

Capítulo 5

Validação e Testes

Validação e testes aqui ...; pode precisar de referir o capítulo 3 ou alguma das suas secções, e.g., a secção 3.2 ...

Pode precisar de apresentar tabelas. Por exemplo, a tabela 5.1 apresenta os dados obtidos na experiência ...

c_1	c_2	c_3	$\sum_{i=1} c_i$
1	2	3	6
1.1	2.2	3.3	6.6

Tabela 5.1: Uma tabela

Para além de tabelas pode também precisar de apresentar figuras. Por exemplo, a figura 5.1 descreve ...



Figura 5.1: Uma figura

Atenção. Todas as tabelas e figuras, e.g., diagramas, imagens ilustrativas da aplicação em funcionamento, têm que ser devidamente enquadradas no texto antes de serem apresentadas e esse enquadramento inclui uma explicação da imagem apresentada e eventuais conclusões (interpretações) a tirar dessa imagem.

Capítulo 6

Conclusões e Trabalho Futuro

Conclusões e trabalho futuro aqui ...

Quais as principais mensagens a transmitir ao leitor deste trabalho?

O leitor está certamente interessado nos temas aqui abordados. Em geral procurará, neste projecto, pistas para algum outro objectivo. Assim, é muito importante que o leitor perceba rapidamente a relação entre este trabalho e o seu próprio (do leitor) objectivo.

Aqui é o local próprio para condensar a experiência adquirida neste projecto e apresentá-la a outros (futuros leitores).

O pressuposto é o de que de que este projecto é um “elemento vivo” que recorreu a outros elementos (cf., capítulo 2) para ser construído e que poderá servir de suporte à construção de futuros projectos.

Apêndice A

Um Detalhe Adicional

O “apêndice” utiliza-se para descrever aspectos que tendo sido desenvolvidos pelo autor constituem um complemento ao que já foi apresentado no corpo principal do documento.

Neste documento utilize o apêndice para explicar o processo usado na **gestão das versões** que foram sendo construídas ao longo do desenvolvimento do trabalho.

É especialmente importante explicar o objetivo de cada ramo (“branch”) definido no projeto (ou apenas dos ramos mais importantes) e indicar quais os ramos que participaram numa junção (“merge”).

É também importante explicar qual a arquitetura usada para interligar os vários repositórios (e.g., Git, GitHub, DropBox, GoogleDrive) que contêm as várias versões (e respetivos ramos) do projeto.

Notar a diferença essencial entre “apêndice” e “anexo”. O “apêndice” é um texto (ou documento) que descreve trabalho desenvolvido pelo autor (e.g., do relatório, monografia, tese). O “anexo” é um texto (ou documento) sobre trabalho que não foi desenvolvido pelo autor.

Para simplificar vamos apenas considerar a noção de “apêndice”. No entanto, pode sempre adicionar os anexos que entender como adequados.

Apêndice B

Outro Detalhe Adicional

Escrever aqui o detalhe adicional que melhor explique outro aspecto (diferente do que está no apêndice A) descrito no corpo principal do documento ...

Bibliografia

[GoogleTasks, 2018] GoogleTasks (2018). Google tasks. https://play.google.com/store/apps/details?id=com.google.android.apps.tasks&hl=pt_PT&gl=US.

[MicrosoftToDo, 2017] MicrosoftToDo (2017). Microsoft to do. <https://todo.microsoft.com/tasks/>.

[Moroney, 2017] Moroney, L. (2017). The firebase realtime database. Technical report, Springer, TR/DCC-2006-8.

[Trello, 2011] Trello (2011). Aplicação trello. <http://www.trello.com/>.