

Desenvolvimento de uma aplicação móvel de monitorização de calorias para Linux Ubuntu Touch

Development of a mobile calories monitoring application for Linux Ubuntu Touch

Ivo Fernandes, Mestrado em Sistemas de Informação, Universidade do Minho, Portugal,
a76170@alunos.uminho.pt

José Luís Pereira, Centro ALGORITMI, Universidade do Minho, Portugal,
jimp@dsi.uminho.pt

Resumo

A necessidade de complementar a oferta de aplicações na loja da Ubuntu Touch representa uma oportunidade de desenvolvimento de aplicações inovadoras e com interesse para a comunidade de potenciais utilizadores. Este artigo, descreve o trabalho realizado no âmbito de uma dissertação de mestrado onde se desenvolveu uma aplicação *open-source* para monitorização das calorias consumidas diariamente por um utilizador. Como processo de desenvolvimento recorreu-se ao Scrum Solo. Os requisitos funcionais e não-funcionais da aplicação foram definidos por diversos *product owners*, tendo sido todos implementados. Após um período de testes preliminares concluiu-se que a aplicação desenvolvida apresenta funcionalidades relevantes e diferenciadoras de outras aplicações similares disponíveis na loja da Ubuntu Touch.

Palavras-chave: linux; ubuntu touch; scrum; calorias; saúde.

Abstract

The need to complement the offer of applications in the Ubuntu Touch store represents an opportunity to develop innovative applications of interest to the community of potential users. This article describes the work carried out within the scope of a master's thesis where an open-source application was developed to monitor the calories consumed daily by a user. As a development process, Scrum Solo was used. The application's functional and non-functional requirements were defined by several product owners, all of which were implemented. After a period of preliminary tests, it was concluded that the developed application presents relevant and differentiating features from other similar applications available in the Ubuntu Touch store.

Keywords: linux; ubuntu touch; scrum; calories; health.

1. INTRODUÇÃO

A crescente popularidade dos *smartphones* e as suas cada vez maiores capacidades em termos de processamento de dados e interação com os utilizadores, tornam-nos excelentes ferramentas de suporte e acompanhamento do bem-estar e saúde das populações. Para esse efeito, é importante que existam disponíveis as aplicações adequadas e que cumprem com as necessidades dos seus utilizadores. Por exemplo, para o auxílio no seguimento de uma meta calórica diária existem no

mercado diversas opções de aplicações móveis pensadas para tal. Entre as mais populares estão: a MyFitnessPal, a Lifesum, a Lose It e a ControlMyWeight. Estas possuem recursos grátis, assim como outras funcionalidades ditas *premium*, mediante a subscrição ou pagamento de uma mensalidade. Todavia estas aplicações estão apenas disponíveis para os sistemas operativos Android e iOS, o que é compreensível, atendendo ao facto de que estas plataformas dominam atualmente o mercado de dispositivos móveis, e sendo por isso do interesse dos fabricantes destas aplicações. Por isso, estas plataformas têm lojas de aplicações com uma oferta considerável quando comparadas com as de sistemas operativos com menos popularidade. Não obstante, isto representa uma oportunidade de destaque e investimento. Assim, a aplicação de contagem/monitorização de calorias desenvolvida neste trabalho é motivada para a promoção de estilos de vida saudáveis dos utilizadores do sistema Ubuntu Touch, contribuindo para uma maior adoção deste sistema.

Para o processo de desenvolvimento de software foi seguido aquele que melhor se ajusta aos objetivos do projeto, com a pretensão de atingir a melhor qualidade possível tendo sido feita uma revisão de literatura para fundamentar a opção.

2. ESTADO DA ARTE

2.1. Conceito de qualidade de software

A qualidade pode ser entendida como algo ‘sem defeitos’ ou que simplesmente ‘funciona’, pelo senso comum. Porém, a literatura justifica-a como algo distinto: “[...] *em conformidade com os requisitos especificamente definidos*” (Crosby, 1986, p. 39, 1996, p. 14). Para o desenvolvimento de *software* o termo foi definido ultimamente como a capacidade de um produto de *software* satisfazer as necessidades acordadas e implícitas sob um conjunto de requisitos (Bott et al., 2005). Similarmente, pode ser justificado como a qualidade em que um *software* atende aos requisitos contratados e de como esses requisitos representam com precisão as necessidades, desejos e expectativas das partes interessadas (Brookshear et al., 2012). A definição dos requisitos é traçada pela literatura como sendo uma etapa muito importante pois, muitos dos problemas encontrados nas entregas dos produtos desenvolvidos devem-se a uma má compreensão e definição dos requisitos (Sommerville, 2011; Susanto, 2019). As definições de qualidade no software podem assim incluir os seguintes pontos (Bourque & Fairley, 2014):

- Fundamentos da qualidade do software;
- Definir requisitos;
- Qualidade da especificação dos requisitos;
- Cumprimento dos requisitos;
- Validação e verificação; e

- Revisões e auditorias.

2.2. Metodologias de desenvolvimento de software

As metodologias de desenvolvimento de *software* podem ser categorizadas em dois tipos: ‘*heavyweight*’ e ‘*lightweight*’ (Akbar et al., 2018). Por um lado, as ‘*heavyweight methodologies*’ são alicerçadas em sequências de fases que abarcam: definição de requisitos; construção da solução; teste e desenvolvimento (Khan et al., 2011). Por outro lado, as ‘*lightweight methodologies*’ são relatadas como as metodologias ágeis e focam-se principalmente em pequenos ciclos iterativos e derivam do conhecimento presente na equipa (Khan et al., 2011). Uma não é melhor que a outra, ambas são importantes (Akbar et al., 2018). Dentro destes grupos existem modelos que são frequentemente usados. Na primeira, incluem-se o Incremental Model, o Spiral Model, o Rational Unified Process (RUP) Model e a Component Based Model e na segunda, são usadas o Prototype Model, Rapid Application Development (RAD) Model, Extreme Programming (XP) Model e o Scrum Model (Khan et al., 2011).

2.3. Aplicações móveis de saúde

Como meio para a disseminação de comportamentos saudáveis, as aplicações móveis, procuram assistir os seus utilizadores: i) na gestão do stress; ii) a melhorar o humor; iii) seguir uma dieta saudável; iv) a aumentar a atividade física; v) ou até para deixar de fumar (Dennison et al., 2013). Bastantes destas aplicações são simples, facultando informação especializada para identificar perigos de saúde ou para seguirem as atividades *fitness* (Lupton, 2014).

De acordo com a Direção Geral de Saúde (DGS), a obesidade afeta 1,5 milhões de portugueses (Silva, 2020). O crescimento dos graus de obesidade põe a população em risco de desenvolver morbilidades como o diabetes tipo 2, cancro e dificuldades cardíacas (Camolas et al., 2017). Ora, o uso de uma aplicação móvel de contagem de calorias pode auxiliar na gestão do peso corporal (Solbrig et al., 2017). Investigações realizadas indicam que este método (uso de uma aplicação) apresenta ganhos superiores aos dos métodos mais tradicionais, como por exemplo seguir uma dieta de revista ou apontar num papel o que se ingere (Turner-McGrievy et al., 2013). Este último pede maior tempo para perscrutar as calorias de cada alimento, bem como dos macronutrientes (Turner-McGrievy et al., 2013). Os possíveis ganhos com o uso de uma aplicação são uma maior adesão à supervisão da ingestão calórica, onde os métodos tradicionais não apresentam, em tempo real, *feedback* para suportar e motivar o processo (Burke et al., 2011). Todavia, é também referido nestas investigações que não são identificadas diferenças entre a perda de peso por uso de uma aplicação ou qualquer outro método (Turner-McGrievy et al., 2013). Porém, a perda de peso é articulada com uma maior motivação quando usadas aplicações móveis.

Outro ponto de proveito para o uso de aplicações móveis tem que ver com a própria aplicação em si. Isto é, uma pessoa pode ter motivação para iniciar um processo de perda peso e recorrer a uma aplicação, mas o uso incorreto da aplicação pode levar a uma má experiência, não sobre os resultados desejados, mas do seu uso. Uma investigação qualitativa pretendeu recolher as oportunidades e desafios das aplicações móveis de suporte à mudança de comportamentos em relação à saúde. As principais conclusões indicam os seguintes temas: a) *smartphones* como fontes de informação valiosas; b) aplicações de monitorização de progresso e de alerta; c) aplicações de mudança de comportamento e redes sociais; d) ceticismo sobre os sensores de contexto; e) avisos úteis ou assédio; f) necessidade de eficiência e conveniência; g) motivação e necessidade de mudança de comportamento; h) descartabilidade; i) credibilidade e precisão; j) preocupações com privacidade e segurança; k) deter controlo sobre as aplicações.

Segundo Dennison e colaboradores (2013), o primeiro tópico (a), é o reconhecimento dos *smartphones* como uma fonte de informação proveitosa. No segundo tópico (b) indicam que, por um lado, as aplicações que fornecem instrumentos de supervisão e rastreamento são percebidas como as mais atrativas, ao conseguirem definir objetivos e seguir o progresso em detalhe, bem como a exatidão da informação inserida. Por outro lado, as aplicações que proporcionam estes momentos podem interferir negativamente no humor e na motivação.

A preocupação com as ações sobre a privacidade da saúde individual (c), faz com que o uso de funcionalidades que necessitam de integração com as redes sociais não sejam muito utilizadas. De forma semelhante, a capacidade que os *smartphones* (d) têm em detetar: a localização; os níveis de atividade e a situação social para fornecer conselhos ou sugestões é considerada na investigação mencionado como pouco relevante. Adicionalmente são referidas as notificações (e) úteis e não úteis, que podem fazer sentir os utilizadores como “controlados”, no caso das não úteis. Na mesma linha o sexto tópico (f) indica que os participantes na investigação não gostam de aplicações que alteram o modo como eles usam o *smartphone* ou que aumentam o consumo de bateria ou que ocupam espaço/memória excessiva.

A grande variedade de aplicações (h) grátis conforma que nenhum utilizador esteja preso a uma em particular. Desta maneira, a credibilidade e reputação das fontes (i) são importantes, isto é, serem desenvolvidas por especialistas ao invés de desconhecidos. Já para as questões de segurança (j) é se a informação fornecida pelos utilizadores fica segura e se são eles a deter o controlo sobre a aplicação e não o inverso (k).

2.4. Sistema Linux Ubuntu Touch

O Ubuntu Touch é um sistema operativo (SO) relativamente recente quando comparado com o iOS ou o Android, e tem como mascote o Yumi. É um sistema livre com suporte profissional por parte da UBports Foundation e conta também com o apoio da sua própria comunidade. Foi apresentado

em janeiro de 2013 pela Canonical, a empresa que criou este sistema originalmente então designado como Ubuntu for Phones. O objetivo desta empresa era obter uma experiência unificadora entre as suas versões *desktop*; *smarttv*; *smartphone* e *tablet*. Para isso, criou o ambiente gráfico Unity.

O conceito era essencialmente atingir a “convergência”, uma inovação que só este sistema oferecia na época. Esta visão pretendia que os *smartphones* fossem capazes de ter uma experiência total de computador de secretária, como descrito no *blog* oficial do ubuntu¹: i) facilidade na gestão de janelas multitarefa; ii) serviços integrados e com notificações; iii) aplicações responsivas desenvolvidas quer para toque no ecrã ou com dispositivo de entrada, que se ajustam à forma e tamanho de ecrã; iv) serviço de comunicações usando as aplicações de telefone, mensagens através da interface do *desktop*.

O primeiro dispositivo a ser vendido com o Ubuntu for Phones, foi o BQ Aquaris E4.5, em fevereiro de 2015. A circunstância aparentava ser promissora com novos dispositivos da Meizu a serem colocados no mercado. Todavia, a 5 de abril de 2017 o projeto foi descontinuado causando grande desilusão na crescente comunidade. Em sentido contrário, Marius Grispgard fundador do UBports Project, que se dedicava a colocar este sistema em *smartphones* não oficiais, decidiu continuar o seu desenvolvimento, com as mudanças indispensáveis, como por exemplo a remoção da Ubuntu Software Store. O contínuo crescimento desta comunidade, levou a que em 2019 passa-se a ser reconhecida como uma fundação, a UBports Foundation, sediada na Alemanha. Hoje, é possível comprar este sistema em *smartphones* novos ou instalá-lo em dispositivos suportados por membros da comunidade.

3. TRABALHO REALIZADO

3.1. Escolha do processo para o desenvolvimento da aplicação

Atendendo ao contexto deste projeto, considerou-se as metodologias ágeis, por terem uma estrutura flexível, serem orientadas para as pessoas e com uma direção de desenvolvimento facilmente alterável. Assim, a opção de modelos ficou na categoria *lightweight*: o Prototype Model, RAD Model, XP Model e o Scrum Model. O Prototype Model e o RAD Model, apresentam trunfos com a sua orientação para a entrega do produto dentro do tempo, embora não considerem os requisitos sistematicamente o que, segundo a literatura, pode resultar em produtos com falhas de qualidade (Geambasu et al., 2011; Khan et al., 2011). Neste sentido, a escolha reduz-se aos modelos XP Model e Scrum. Durante o ano de 2021 a digital.ai lançou o seu habitual relatório² sobre as metodologias Agile em uso. A metodologia que predomina é o Scrum com 66%, o XP Model conta apenas com 1%, existindo ainda 6% de empresas que estão a combinar o Scrum com o XP em modo híbrido. A

¹Demonstrado em: <https://ubuntu.com/blog/ubuntu-path-to-convergence>, acessido a 05/07/2022.

²Demonstrado em: <https://itnove.com/wp-content/uploads/2021/07/15th-state-of-agile-report.pdf>, acessido a 05/07/2022.

segunda categoria com maior percentagem é a ScrumBan que incorpora a metodologia KanBan – uso de quadros para controlo de fluxo de trabalho. Posto isto, a metodologia adotada neste projeto foi o modelo Scrum.

Não obstante a eleição do Scrum como modelo a seguir, foi encontrada na literatura uma adaptação deste processo para uso individual – o designado Scrum Solo (Figura 1). Segundo Pagotto et al. (2016), o *product backlog* e o *sprint backlog* mantêm-se iguais; os *sprints* têm a longevidade de uma semana e no final de cada iteração deve ser entregue um protótipo e sempre que necessário realizar reuniões de orientação com o grupo de validação. O processo inicia com a criação do *product backlog* que contém todas as tarefas necessárias para atingir o produto acordado, e por cada sprint semanal escolhem-se as tarefas que vão ser realizadas durante essa semana. Os sprints terminam assim que todas as tarefas estejam concluídas.



Figura 1 - Scrum Solo Diagram (Pagotto et al., 2016).

3.2. Configuração das aplicações de suporte

Para a gestão do *workflow* de todas as atividades do projeto foi utilizado o JIRA Software. Criaram-se ‘épicos’ de forma a agregar as várias tarefas existentes no *product backlog*, para ser possível saber as que estão ‘por fazer’, as que estão ‘em progresso’ e as que estão ‘concluídas’. Adicionalmente, este *software* foi integrado com o repositório da aplicação no GitHub, para que quando fosse concluído um *issue* o JIRA Software atualizasse o quadro dos ‘épicos’. Configurou-se uma máquina virtual na Azure Cloud, para suportar o desenvolvimento da aplicação e, posteriormente, para permitir que voluntários pudessem traduzir a aplicação.

3.3. Retrospectiva

A primeira atividade foi uma sessão de *briefing* online através da plataforma Zoom com integração da aplicação Miro, onde foram colocados quadros com três perguntas na qual os *product owner* colocavam *post-it* com as suas opiniões e ideias. Com esta sessão, procurou-se entender o perfil dos 3 *product owner*.

O primeiro quadro questionava: “sabes o que é o Linux Ubuntu Touch?” Os post-it colocados, indicam que a maioria tem conhecimento desta plataforma. O segundo cartão continha a questão: “descreve como seria uma aplicação de contagem de calorias?” A maioria entende que uma aplicação deste tipo deve ter um design apelativo, que indique a quantidade de calorias ingeridas e a quantidade de calorias em falta. Deve também permitir o registo da ingestão de água e de atividade física. Por fim, a última questão foi para recolher sugestões de nomes para a aplicação. Durante a sessão, ficou imediatamente definido utFoods como o nome.

3.4. Levantamento de requisitos

A segunda atividade foi o levantamento dos requisitos. Esta etapa é referida na literatura como um momento crítico (Sommerville, 2011; Susanto, 2019). As atas foram enviadas aos participantes, onde estes poderiam escrever aquilo que gostariam de ter na aplicação. Das atas enviadas resultaram, inicialmente, vinte e dois requisitos funcionais (RF), sendo que no decorrer do projeto foram pedidos mais dois requisitos:

- RF1. Registar a ingestão diária de alimentos;
- RF2. Registar a ingestão de alimentos do(s) dia(s) anterior(es) ou seguinte(s);
- RF3. Selecionar alimentos pré-definidos;
- RF4. Selecionar alimentos de histórico de alimentos;
- RF5. Criar lista de alimentos;
- RF6. Fazer o scan de códigos de barras;
- RF7. Definir a ingestão por quantidade e porção;
- RF8. Registar a água ingerida;
- RF9. Definir objetivo (manter, ganhar/perder peso);
- RF10. Consultar receitas;
- RF11. Consultar o consumo médio de calorias por mês;
- RF12. Consultar os alimentos consumidos por mês;
- RF13. Atualizar o peso;
- RF14. Atualizar a idade;
- RF15. Atualizar a altura;
- RF16. Registar a pressão arterial baixa e arterial alta;

- RF17. Consultar a evolução do peso;
- RF18. Consultar o índice de Massa Corporal;
- RF19. Consultar a pressão sanguínea arterial;
- RF20. Fazer a gestão do armazenamento;
- RF21. Registrar o tempo de atividade física;
- RF22. Extrair os dados em formato CSV;
- RF23. Lista de alimentos partilhada entre os utilizadores da aplicação; e
- RF24. Copiar as ingestões de um dia para o outro.

Quanto aos requisitos não funcionais (RNF) foram definidos três:

- RNF1. A interface deve de ser de fácil compreensão e intuitiva;
- RNF2. A aplicação deve seguir a *guideline* do sistema; e
- RNF3. Estar disponível em alguns idiomas.

Ficou acordado entre as partes envolvidas que o conceito de qualidade, para este projeto, é o cumprimento total dos requisitos aceites para o *product backlog*. Dessa forma, adotaram-se meios para que a comunicação fosse a melhor possível: foram realizadas reuniões por videochamada e criado um grupo para troca de mensagens em ferramentas de mensagens instantâneas. Também, foram criadas *builds* de teste da aplicação para que fosse possível acompanhar o desenvolvimento, validar ou rejeitar o trabalho realizado até ao momento, bem como identificar potenciais erros que foram apontados e explicados. Por fim, foi também assegurado que a aplicação e o seu uso não causam danos aos utilizadores, bem como ao ambiente.

3.5. *Modelação*

Nesta fase foram desenvolvidos diagramas de casos-de-uso, diagramas de sequência, diagramas de atividades e diagramas de entidades-e-relacionamentos. Estes últimos, foram construídos a pensar nos RF definidos bem como os que são obtidos através da API da OpenFoodsFacts. Adicionalmente, foram realizados *mockups* de interface para ajudar na obtenção de uma representação daquilo que vai ser o aspeto visual e das funcionalidades da aplicação. Estes desenhos, foram inspirados com ideias retiradas das aplicações de contagem de calorias mais populares.

3.6. *Implementação*

O processo de desenvolvimento da aplicação precisou de 7 *sprints* semanais para ficar concluído e foi utilizado como dispositivo móvel de referência um LG Google Nexus 5, e utilizadas as

linguagens de programação: QML; JavaScript; Python e C++. Na primeira semana desenvolveu-se o processo de início de sessão. Na segunda semana foram executadas as operações necessárias para a criação do perfil do utilizador. Na terceira semana implementou-se a base de dados da aplicação e os processos de extração de informação de informação da API da OpenFoodFacts para criar a lista de alimentos com sugestões. Na quarta semana, foi adicionada a página principal e a possibilidade de registar ingestões e de visualizar um gráfico com as calorias consumidas e restantes disponíveis. A quinta semana tratou de criar um menu que dá acesso a outras funcionalidades da aplicação como: i) definições; ii) análise de dados e iii) perfil do utilizador. A sexta semana, adicionou à aplicação a capacidade de digitalizar códigos de barras e de registar notas. Por fim, na última semana tratou-se de traduzir a aplicação para alguns idiomas (português, inglês, francês, espanhol), suporte a temas e licenciamento.

Após estes *sprints* semanais, todas as tarefas presentes no *product backlog* foram concluídas, incluindo os *issues* reportados, detetados em fases de teste e validação. O resultado final, presente na figura 2, em termos de interface ficou um pouco diferente daquele idealizado na fase de modelação, mas ainda assim as linhas pensadas e desejadas foram conseguidas.

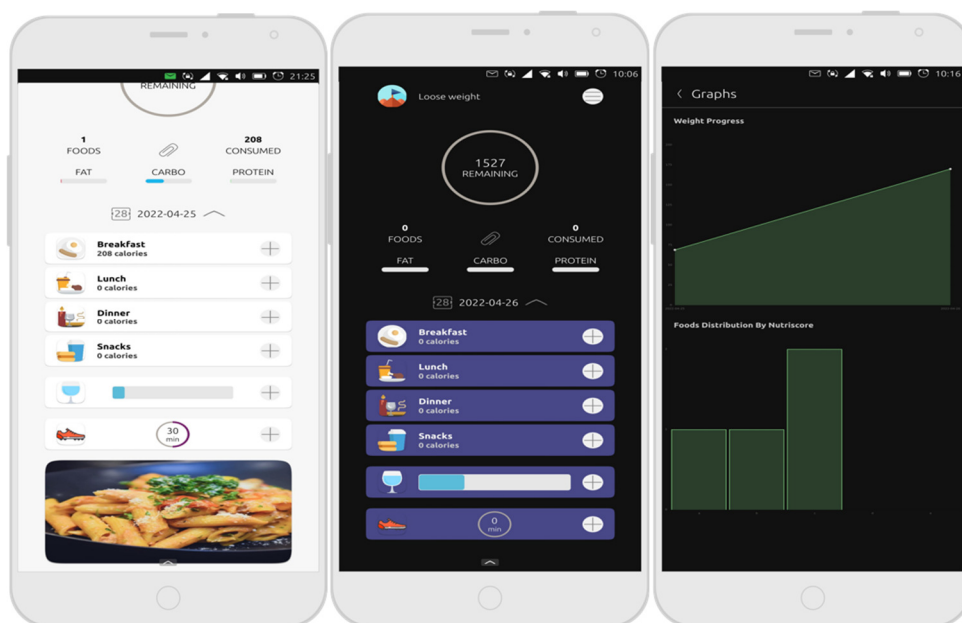


Figura 2 – Capturas de ecrã da aplicação desenvolvida

Durante o processo de desenvolvimento da aplicação foram criadas *builds* com o intervalo [v0.1.0 – v1.5.2], seguindo o esquema *semantic versioning* que usa três dígitos (Raemaekers et al., 2014) e tem a semântica: *major*, *minor* e *patch*. Estas *builds* foram disponibilizadas aos participantes, na qual estes reportaram um total de 14 *issues*.

3.7. Análise de dados

Após a conclusão do desenvolvimento da aplicação, foi realizada uma pequena experiência de uso. Esta decorreu durante 10 dias, entre 06 e 15 de maio de 2022, com a participação de quatro voluntários. O objetivo pretendido não era procurar o cumprimento da meta configurada (manter, ganhar ou perder peso) pois não houve tempo suficiente para isso, mas antes o teste das suas funcionalidades e fornecer *feedback* para melhorias futuras. Assim, dos dados recolhidos, os perfis dos participantes configuram-se do seguinte modo: três indivíduos do sexo masculino e uma do sexo feminino; a média de idade é de 36,5 anos; o peso médio é de 69,5KG e a altura média é de 169cm. Quanto ao nível de atividade física, os dados dizem que apenas 1 dos participantes tem atividade física moderada, enquanto os restantes (3) têm atividade física leve ou muito leve. Quanto às calorias recomendadas pela aplicação, a média de meta calórica diária aconselhada foi de 1979,75 calorias. O valor mais alto atribuído foi de 2224 calorias, a mais baixa de 1831 calorias. O desvio padrão é de 173,41. Uma das informações mais relevantes desta análise é se o objetivo diário das calorias foi cumprido. Para cada participante a recomendação calórica fornecida pela aplicação nunca foi atingida, tendo apenas sido registada um excesso em dois dias (Figura 3).

		cal	rec_cal	calories_difference	is_objective_completed
id_user	date				
1	2022-05-06	387	1888	1501	False
	2022-05-07	186	1888	1702	False
3	2022-05-09	250	2224	1974	False
	2022-05-10	3164	2224	-940	False
4	2022-05-06	763	1831	1068	False
	2022-05-07	848	1831	983	False
	2022-05-08	1171	1831	660	False
	2022-05-09	915	1831	916	False
	2022-05-10	1091	1831	740	False
	2022-05-11	654	1831	1177	False
	2022-05-12	1080	1831	751	False
	2022-05-14	1564	1831	267	False
5	2022-05-06	2123	1976	-147	False
	2022-05-07	784	1976	1192	False
	2022-05-09	1289	1976	687	False
	2022-05-11	79	1976	1897	False
	2022-05-12	242	1976	1734	False
	2022-05-13	272	1976	1704	False
	2022-05-14	1757	1976	219	False

Figura 3 – Meta Calórica Diária

Por um lado, os poucos registos efetuados podem ser entendidos à luz do que é mencionado pela literatura sobre este tipo de aplicações. Isto é, requerem que seja despendido pouco esforço na introdução dos registos para maximizar o uso da aplicação (Dennison et al., 2013). Por outro lado,

olhando para a figura abaixo, o não cumprimento da meta diária não significa que o problema esteja na aplicação em si, podendo estar na falta de motivação individual para o bem-estar pessoal (Turner-McGrievy et al., 2013). Porém, a perda de motivação pode também ser provocada pelo uso de uma aplicação, caso ela transgrida alguns tópicos considerados importantes pelo utilizador (Dennison et al., 2013).

4. CONCLUSÃO

Este artigo descreve um trabalho de dissertação de mestrado onde foi desenvolvida uma aplicação móvel de contagem de calorias para o sistema Ubuntu Touch. Foi seguida a Scrum Methodology, mais concretamente a Scrum Solo. Criou-se um *product backlog* com todas as tarefas a cumprir, e em cada *sprint* semanal escolheram-se as tarefas que foram realizadas. Desta metodologia, fizeram parte também um conjunto de atividades, atores e artefactos (Pagotto et al., 2016). A preferência pelo Scrum é fundamentada por ser uma metodologia predominante (*15th State of Agile Report*, 2021) e ajustada ao trabalho desenvolvido. As funcionalidades foram definidas por um grupo de pessoas participantes neste trabalho, acordando-se que a implementação de todos os requisitos definidos era o critério principal de qualidade deste projeto. A aplicação utiliza APIs gratuitas da OpenFoodsFacts e da TheMealDB para obter informação de alimentos com código de barras, e de sugestões de refeições respetivamente.

Findo o desenvolvimento, a aplicação foi utilizada durante 10 dias por 4 participantes. A análise dos dados impede conclusões assertivas, dado o baixo número de participantes e registos, mas permite uma aproximação a alguns tópicos descritos pela literatura, como por exemplo o pouco esforço para efetuar o registo de informação neste tipo de aplicações, onde é o utilizador que necessita de introduzir os dados (Dennison et al., 2013). Neste sentido, nenhum dos participantes conseguiu cumprir com a meta calórica diária recomendada, o que pode colocar em destaque a falta de motivação para o bem-estar pessoal (Turner-McGrievy et al., 2013).

Durante o desenvolvimento foram encontradas algumas dificuldades no processo de obter informação sobre este sistema. A maior dificuldade foi a implementação da funcionalidade da leitura de código de barras, uma vez que o sistema não possui no seu *software development kit* (SDK) uma opção para tal. Foi necessário procurar soluções e quando encontradas, sentiram-se dificuldades na integração com o sistema Ubuntu Touch, mas foram colocados todos os esforços para ultrapassar a situação.

Algumas limitações da aplicação são referentes à API da OpenFoodsFacts e da TheMealDB. A primeira, apesar de conter uma base de dados considerável de alimentos não inclui refeições tradicionais e sem códigos de barra. A segunda, na sua versão *free* proíbe a distribuição em lojas de

aplicações online o que condiciona a colocação desta aplicação de imediato na Store do Ubuntu Touch, e fornece refeições aleatórias em quantidade reduzida.

É necessário considerar para melhorias futuras, concretamente neste tipo de aplicações, a existência de uma base de dados de alimentos e refeições extensa. Nesse sentido, um melhoramento futuro seria de criar uma plataforma online onde as comunidades de utilizadores pudessem adicionar e editar alimentos/refeições. Outra melhoria futura, está relacionada com o armazenamento na *cloud* dos dados gerados e assim permitir que os utilizadores pudessem utilizar mais do que um dispositivo. Todavia, o *feedback* recebido sobre a aplicação no geral foi positivo, e destacada a funcionalidade de *scan* de código de barras, de ser possível fazer gestão do armazenamento de dados e exportar os mesmos em formato CSV para serem utilizados noutras aplicações.

Com este trabalho, procurou-se contribuir para a adoção e divulgação do sistema Ubuntu Touch, aumentar a documentação existente sobre esta temática e adicionalmente promover estilos de vida mais saudáveis.

AGRADECIMENTOS

Este trabalho foi financiado por fundos nacionais através da FCT - Fundação para a Ciência e Tecnologia no âmbito do Projeto UIDB/00319/2020.

REFERÊNCIAS

- Akbar, M. A., Sang, J., Khan, A. A., Amin, F. E., Nasrullah, Hussain, S., Sohail, M. K., Xiang, H., & Cai, B. (2018). Statistical Analysis of the Effects of Heavyweight and Lightweight Methodologies on the Six-Pointed Star Model. *IEEE Access*, 6, 8066–8079. <https://doi.org/10.1109/ACCESS.2018.2805702>
- Bott, F., Coleman, A., Eaton, J., & Rowland, D. (2005). *Professional Issues in Software Engineering*. Taylor and Francis.
- Bourque, P., & Fairley, R. (2014). *Guide to the Software Engineering Body of Knowledge, Version 3.0*.
- Brookshear, J. G., Smith, D. T., & Brylow, D. (2012). *Computer science: An overview (11th ed)*. Addison-Wesley.
- Burke, L. E., Conroy, M. B., Sereika, S. M., Elci, O. U., Styn, M. A., Acharya, S. D., Seveck, M. A., Ewing, L. J., & Glanz, K. (2011). The Effect of Electronic Self-Monitoring on Weight Loss and Dietary Intake: A Randomized Behavioral Weight Loss Trial. *Obesity (Silver Spring, Md.)*, 19(2), 338–344. <https://doi.org/10.1038/oby.2010.208>
- Camolas, J., Gregório, M., Mendes de Sousa, S., & Graça, P. (2017). *Obesidade: Otimização da abordagem terapêutica no serviço nacional de saúde*. Direcção-Geral da Saúde.
- Crosby, P. B. (1986). *Quality without tears: The art of hassle-free management (3rd print)*. McGraw-Hill.
- Crosby, P. B. (1996). *Quality is still free: Making quality certain in uncertain times*. McGraw-Hill.
- Dennison, L., Morrison, L., Conway, G., & Yardley, L. (2013). Opportunities and Challenges for Smartphone Applications in Supporting Health Behavior Change: Qualitative Study. *Journal of Medical Internet Research*, 15(4), 1–12. <https://doi.org/10.2196/jmir.2583>
- Digital.ai (Ed.). (2021). 15th State of Agile Report (pp. 1–23). <https://itnove.com/wp-content/uploads/2021/07/15th-state-of-agile-report.pdf>
- Geambasu, C., Jianu, I., Jianu, I., & Gavrilă, A. (2011). Influence Factors For The Choice Of A Software Development Methodology. *Accounting and Management Information Systems*, 10(4), 479–494.

- Khan, A. I., Qurashi, R. J., & Khan, U. A. (2011). A Comprehensive Study of Commonly Practiced Heavy and Light Weight Software Methodologies. *IJCSI International Journal of Computer Science Issues*, 8(2), 441–450.
- Lupton, D. (2014). Apps as Artefacts: Towards a Critical Perspective on Mobile Health and Medical Apps. *Societies*, 4(4), 606–622. <https://doi.org/10.3390/soc4040606>
- Pagotto, T., Fabri, J. A., Lerario, A., & Goncalves, J. A. (2016). Scrum solo: Software process for individual development. 2016 11th Iberian Conference on Information Systems and Technologies (CISTI), 1–6. <https://doi.org/10.1109/CISTI.2016.7521555>
- Raemaekers, S., van Deursen, A., & Visser, J. (2014). Semantic Versioning versus Breaking Changes: A Study of the Maven Repository. 2014 IEEE 14th International Working Conference on Source Code Analysis and Manipulation, 215–224. <https://doi.org/10.1109/SCAM.2014.30>
- Silva, C. (2020, October 31). 1,5 milhões de obesos em Portugal: DGS quer consultas nos centros de saúde. *Jornal de Notícias*. <https://www.jn.pt/nacional/15-milhoes-de-obesos-em-portugal-dgs-quer-consultas-nos-centros-de-saude-12984375.html>
- Solbrig, L., Jones, R., Kavanagh, D., May, J., Parkin, T., & Andrade, J. (2017). People trying to lose weight dislike calorie counting apps and want motivational support to help them achieve their goals. *Internet Interventions*, 7, 23–31. <https://doi.org/10.1016/j.invent.2016.12.003>
- Sommerville, I. (2011). *Software engineering* (9th ed). Pearson.
- Susanto, A. (2019). System Development Method with The Prototype Method. 8(07), 141–144.
- Turner-McGrievy, G. M., Beets, M. W., Moore, J. B., Kaczynski, A. T., Barr-Anderson, D. J., & Tate, D. F. (2013). Comparison of traditional versus mobile app self-monitoring of physical activity and dietary intake among overweight adults participating in an mHealth weight loss program. *Journal of the American Medical Informatics Association*, 20(3), 513–518. <https://doi.org/10.1136/amiajnl-2012-001510>