



Universidade Federal de Pernambuco
Graduação em Ciência da Computação
Centro de Informática

Thiago Borges Jordani

**Práticas energeticamente eficientes de
desenvolvimento mobile.**

Trabalho de Graduação

Recife

2018

Universidade Federal de Pernambuco
Centro de Informática
Departamento de Ciência da Computação

Práticas energeticamente eficientes de desenvolvimento móvel.

Trabalho apresentado ao Programa de Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Bacharel em Ciência da Computação

Aluno: Thiago Borges Jordani

Orientador: Fernando Castor

Recife

Dezembro 2018

Agradecimentos

Agradeço primeiramente à Deus, por todas as oportunidades e desafios que me foram dados e que me guiaram no ciclo que chega ao fim com este trabalho. Agradeço também ao meu orientador, Fernando Castor, que teve toda a paciência e disponibilizou do seu tempo para me auxiliar na produção deste trabalho. Da mesma maneira, agradeço a todos os outros professores desta instituição de ensino que foram responsáveis na construção da formação acadêmica que tenho hoje.

Por fim, agradeço aos meus pais, Plínio e Mércia, que me apoiaram e me deram todo o suporte nesta caminhada, sendo duros quando necessário, e nunca deixaram que dificuldade alguma me atrapalhasse. Agradeço, em especial, por todo o seu apoio e ajuda na realização do desenvolvimento deste trabalho.

Resumo

Uma das principais ferramentas do tempo contemporâneo, apesar do seu surgimento exclusivamente corporativo e preços que dificilmente atrairiam o interesse da população geral, o smartphone mantém uma crescente relevância na sociedade moderna. A substituição do uso de aparelhos tradicionais já ocorre desde o ano de 2016, acompanhada por um ascendente tempo médio de uso dessa já não tão nova tecnologia.

A identidade portátil dos smartphones, no entanto, faz seu tempo hábil de uso ser diretamente relacionado ao tempo de duração da carga de sua bateria. Além disso, apesar da existência de guias oficiais de desenvolvimento energético-eficientes para as principais plataformas, suas instruções não possuem indicação do tipo de economia esperado ao aplicá-las e não se sabe se estas cobrem toda a gama de possíveis táticas de desenvolvimento móvel energeticamente eficiente.

Desta maneira, neste trabalho, serão estudadas as técnicas de desenvolvimento energeticamente eficientes/conscientes, relacionando-as com as principais técnicas de economia devidamente corroboradas pela produção científica recente. Buscando-se, desta forma, não apenas a validade das técnicas implementadas, tendo sua eficiência devidamente atestada por produção científica, mas, também, novas oportunidades de pesquisa surgirão a partir do que já é implementado na comunidade de desenvolvimento de aplicativos móveis.

Para tanto, serão utilizadas as práticas de pesquisa ditadas pela metodologia nomeada como mini-revisão sistemática, devido à sua fácil reprodutibilidade e pela possibilidade de expansão, através da aplicação de uma revisão sistemática completa, caso resultados interessantes sejam encontrados.

Palavras Chave: Eficiência energética, desenvolvimento móvel

Abstract

One of the main contemporary tools, despite its early exclusivity to corporate usage and prices that would hardly attract the interest of the general population, the smartphones maintains a growing relevance in modern society. The usage replacement of the most traditional apparels already occurs since the year 2016, accompanied by an ascending average time of use of this not so new technology.

The portable uniqueness of smartphones, however, makes their usage time to be directly linked to the duration of its battery charge. In addition, despite the existence of official energy-efficient development guides for the major mobile platforms, their instructions do not indicate the kind of savings expected by applying them and it is not known whether they cover the full range of possible energy-efficient mobile development tactics.

Thus, this work will study the techniques of energy efficient/conscious development, relating them to the main energy-saving techniques duly corroborated by the recent scientific production. Seeking, in this way, not only the validation of the techniques implemented, having their efficiency duly certified by scientific production, but also to find new research opportunities from what is already implemented in the application development communities.

To do so, the research will utilize practices as dictated by the methodology named as systematic mini-review, due to their easy reproducibility and the possibility of expansion, through the application of a complete systematic review, if interesting results are found.

Keywords: Energy efficiency, mobile development

Lista de Figuras

Figura 1. Guia iOS - Estado ocioso vs Consumo da CPU.

Figura 2. Guia iOS - Custo fixo vs custo dinâmico.

Figura 3. Disposição dos artigos encontrados

Figura 4. Artigos por categoria de tipo de conhecimento

Figura 5. Distribuição das recomendações quanto ao seu respaldo acadêmico

Lista de Tabelas

Quadro 1. Recomendações gerais - iOS

Quadro 2. Recomendações de networking - iOS

Quadro 3. Monitoramento de localização - iOS

Quadro 4. Demais recomendações - iOS

Quadro 5. Recomendações de networking - Android

Quadro 6. Outras recomendações - Android

Quadro 7. Resultados FSE

Quadro 8. Resultados ICSE

Quadro 9. Resultados MSR

Quadro 10. Categorização de Artigos

Quadro 11. Respaldos referentes à academia e às documentações oficiais.

Índice

| | |
|--|-----------|
| 1 INTRODUÇÃO | 9 |
| 1.1 Contextualização e Motivação | 9 |
| 1.2 Objetivos | 10 |
| 1.3 Estrutura do Documento | 11 |
| 2 FUNDAMENTAÇÃO TEÓRICA | 12 |
| 2.1 Ambiente Mobile | 12 |
| 2.2 Autonomia energética | 13 |
| 2.3 O sistema iOS | 14 |
| 2.3.1 Informações gerais | 15 |
| 2.3.2. Networking | 17 |
| 2.3.3 Monitoramento de localização | 18 |
| 2.3.4 Demais recomendações | 20 |
| 2.4 O sistema Android | 21 |
| 2.4.1 Networking | 22 |
| 2.4.2 Outras recomendações | 23 |
| 3 DESENVOLVIMENTO | 26 |
| 3.1 Metodologia de pesquisa: Revisão Sistemática | 26 |
| 3.1.1 Coleta de dados | 28 |
| 3.1.1.1 Planejamento | 28 |
| 3.1.1.2 Execução | 29 |
| 3.1.1.3 Resultados Obtidos | 29 |
| 3.2 Análise do Conteúdo | 34 |
| 4 RESULTADOS | 39 |
| 4.1 Definições | 39 |
| 4.1.1 Recomendações de desenvolvedores | 39 |
| 4.1.2 Resultados Acadêmicos | 40 |
| 4.2 Respostas | 42 |
| 4.2.1 Recomendações quanto ao seu respaldo na academia | 42 |
| 4.2.2 Respaldos quanto às recomendações oficiais | 43 |
| 4.2.3 Incompletude das recomendações oficiais | 45 |

| | |
|-------------------------------|-----------|
| 5 CONSIDERAÇÕES FINAIS | 47 |
| 5.1 Ameaças à validade | 48 |
| 5.2 Trabalhos Futuros | 48 |
| Referências | 50 |
| Apêndice | 55 |

1 INTRODUÇÃO

1.1 Contextualização e Motivação

Ao longo dos anos recentes, os smartphones, tablets e demais aparelhos mobile têm mais do que firmado sua presença nos mercados e cotidianos ao redor do planeta. Tendo ultrapassado pela primeira vez os desktops como principal ferramenta de uso da web no mundo em 2016 [1], uma marca alcançada no Brasil desde 2014 [2], estes aparelhos apresentam cada vez mais importância no dia a dia. Paralelamente, o uso diário médio destes aparelhos tem crescido de forma significativa, com fontes citando um uso médio superior a 4 horas por dia [3]; seu uso, no entanto, depende diretamente da duração da carga de sua bateria, tornando o tempo entre recargas algo de grande valor para os usuários [14].

O tema de desenvolvimento sustentável, ou Green Computing, já é citado por pesquisadores desde, pelo menos, 2008 [4] e, com a chegada destes novos aparelhos, altamente restringidos pela disponibilidade de energia, não é surpresa a popularidade do tema consumo de energia em aparelhos móveis. No entanto, mesmo com todo um esforço de pesquisa, observou-se, no ano de 2013, que cerca de 18% de todos os comentários relativos a aplicativos presentes na Play Store, da plataforma Google, continham críticas quanto ao consumo de energia [5]. Tamanha aceitação e a frequência, praticamente anual, de atualização desses sistemas, faz com que este se mantenha um popular tema de estudo na academia, sob as mais diferentes visões de pesquisa [6,7].

Neste momento, tanto a Apple quanto a Google, maiores nomes no mercado quanto a sistemas operacionais mobile [8], disponibilizam em suas documentações guias de recomendações para uso eficiente de energia em seus respectivos sistemas, iOS e Android [9,10]; no entanto, não sabemos se estas recomendações se aplicam à realidade prática das plataformas, se existe uma homogeneidade entre elas ou mesmo se elas estão em consonância com a produção recente dos pesquisadores da área.

1.2 Objetivos

Este trabalho terá por objetivo analisar propostas de desenvolvimento mobile energético-eficientes das plataformas mais utilizadas na atualidade, apresentando para o desenvolvedor, dessa maneira, técnicas de desenvolvimento com eficiência energética devidamente demonstrada. Assim, caberá a ele, apenas, selecionar as técnicas que melhor se aplicam a sua necessidade.

Para uma coleta, apresentação e interpretação de dados eficiente, este projeto se baseará no método científico de pesquisa chamado *Systematic Review(SR)*; Utilizado com frequência para a definição de evidências do mais alto nível no campo da medicina [11]. Conforme descrito por Bolchini et al, apesar do método não se adaptar perfeitamente devido à diferenças intrínsecas dos dois campos de estudo, visto que as amostragens realizadas para estudos de engenharia de software precisam de conhecimento técnico, SR pode ser um formato favorável à centralização de conhecimentos disponíveis para o desenvolvedor.

Devido à rápida evolução das tecnologias de desenvolvimento e o tempo restrito para o desenvolvimento deste projeto, no entanto, será utilizada a modalidade chamada Mini-Review [12], selecionando as devidas restrições de pesquisa para que o resultado final seja factível dentro do tempo estimado para sua realização e mantenha sua relevância para o leitor final(desenvolvedor).

Desta maneira, seguindo o protocolo selecionado, este projeto tentará responder a seguinte pergunta de pesquisa:

RQ1. Até que ponto as recomendações de economia de energia feitas por desenvolvedores de aplicativos para as principais plataformas de computação móvel têm respaldo na literatura científica recente na área de Engenharia de Software?

Complementarmente, será respondida também:

RQ2. Quais técnicas propostas na literatura científica recente sobre economia de energia não aparecem entre os guias oficiais para as principais plataformas de computação móvel?

1.3 Estrutura do Documento

Os capítulos seguintes serão apresentados seguindo a seguinte estrutura:

- Capítulo 2: Introdução de um breve histórico dos smartphones, seguido da sua forte relação com a necessidade de um desenvolvimento energeticamente eficiente. Além disso, serão apresentadas as recomendações oficiais das plataformas iOS e Android, servindo como ponto de apoio e comparação para os diversos estudos realizados na academia.
- Capítulo 3: Descreve a metodologia aplicada neste trabalho, a mini-revisão sistemática, discorrendo sobre sua importância para o meio científico, apresentando e suas formalidades e, por fim, introduzindo o material obtido para o estudo a ser realizado.
- Capítulo 4: Apresenta o estudo do material obtido como resultado da metodologia descrita no capítulo anterior, categorizando-o de maneira que facilite sua utilização de acordo com as necessidades da mini-revisão sistemática.
- Capítulo 5: Sintetiza os resultados obtidos da pesquisa, apresentando possíveis melhorias para o trabalho desenvolvido e apontando para possíveis recomendações de pesquisa complementares ao conhecimento adquirido.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Ambiente Mobile

Lançado pela IBM no ano de 1993, o Simon, ao apresentar a inovadora união entre as funcionalidades do telefone padrão e as dos PDAs, ficou conhecido por ser o primeiro smartphone da história; Devido a tecnologia disponível na época e objetivo de atingir um público específico, no entanto, este aparelho pouco se assemelha aos smartphones do nosso dia a dia.

O período que Sarwar e Soomro(2013) definem como a primeira fase da “Era dos smartphones”, iniciado a partir do lançamento do Simon, é caracterizado por seu foco empresarial, com smartphones que apresentam features voltadas para o mundo corporativo e preços inacessíveis para o mercado consumidor geral. Nesta era, a empresa Blackberry torna-se conhecida por apresentar o aparelho mais inovador deste momento.

Por outro lado, no ano de 2007, a Apple anuncia pela primeira vez um smartphone ao público geral, iniciando-se assim a popularização dos smartphones entre o mercado consumidor não-corporativo e a segunda fase definida por Sarwar e Soomro. Nesta fase, também, é notável o investimento da Google neste mercado com o anúncio do seu sistema operacional (SO) open source, Android.

No ano de 2008, por fim, inicia-se a terceira fase da era dos smartphones finalizando a separação entre o mercado corporativo e a população em geral; Em grande parte devido a massificação de aparelhos com o SO Android e diversas melhorias nos sistemas.

Dentre estas melhorias apresentadas na terceira fase, o surgimento das lojas de aplicativos pode ser considerado um dos maiores marcos na história dos smartphones. Tendo-se iniciado no ano de 2008, com o surgimento da App Store no iOS e do Android Market, as ferramentas disponíveis ao usuário aumentaram de maneira incomparável e gerou um mercado que, desde sua criação, já rendeu um valor superior à 70 bilhões de dólares [16].

O domínio dos sistemas criados pela Apple e Google culminou em uma representação

praticamente inexpressiva de outros sistemas operacionais móveis [17, 18]. Desta maneira, o estudo dos sistemas Android e iOS garante a este estudo uma cobertura superior à 99% dos aparelhos no mercado.

2.2 Autonomia energética

Com sua praticidade e grande número de funcionalidades, os dispositivos móveis se tornaram aparelhos de alta importância no dia a dia da sociedade contemporânea. Tal importância têm facilmente se refletido em um grande impacto do uso diário destes, com estimativas atuais que chegam a indicar um uso médio de até 4 horas entre adultos norte americanos [3].

Por outro lado, o uso especialmente portátil dos smartphones torna a sua utilização efetiva diretamente proporcional à duração de suas baterias. Esta correlação faz com que o tempo de vida da bateria continue como um grande impulsionador da satisfação dos usuários [14] e, deste modo, o mercado de smartphones, juntamente com o de carros elétricos, têm impulsionado a pesquisa de diversas novas tecnologias em baterias [13].

Além disso, conforme encontrado pelo portal Hackernoon, aproximadamente 90% do manuseio de dispositivos móveis é relacionado à utilização de aplicativos [3], fazendo com que otimizações de software sejam tão importantes quanto a evolução e otimização do hardware destes aparelhos. Ainda mais, foi observado tanto entre a comunidade acadêmica [22] quanto entre desenvolvedores [23] que a iluminação da tela, uso da CPU e, a utilização dos componentes de hardware de rede estão entre os maiores consumidores de energia dos aparelhos, o que refletirá tanto nas recomendações dos sistemas quanto nos principais focos de pesquisa do meio acadêmico.

Segundo resultados de “Li et al”(2014), dentre os gastos não ociosos do sistema Android entre as aplicações testadas, cerca de 91.4% são relacionados à chamadas de API do sistema, tendo como mais expressiva porcentagem destes gastos, aproximadamente 40%, as chamadas da API de networking. Por outro lado, dos gastos relacionados ao código do app, um quantidade significativa é relacionada à manipulação de dados pelo desenvolvedor. No

entanto, é encontrado que a maior parte dos gastos de energia se dá durante o estado ocioso dos aplicativos, sendo responsável por cerca de 60% dos gastos energéticos de uma aplicação android. Mais especificamente, “Wan et al”(2015) estima que 41% das aplicações podem estar gastando até o dobro de energia a mais que o necessário em suas soluções.

Devido à abrangência de gastos de energia relacionados à implementação de aplicativos móveis e sua necessidade crítica de uso eficiente de energia, tem-se observado na academia a busca pelos mais diversos meios e abordagens para a economia de energia em apps. Pode-se facilmente observar a busca por designs mais energeticamente eficientes [13, 14], relacionados à diminuição dos gastos em estados ociosos, assim como uma busca por auxiliar a implementação de chamadas de API menos gastosas [15] e, também, a procura de usos mais energeticamente conscientes de estruturas de dados [25,26], por exemplo.

Por fim caberá sempre ao desenvolvedor buscar o uso de métodos energeticamente eficientes, mas sempre equilibrando a economia de energia e os benefícios que a implementação pode trazer ao desenvolvedor e ao usuário. “Sahin et al”, por exemplo, apesar de encontrar que a utilização de certos métodos de ofuscação de código de aplicativos Android podem causar maior ou menor impacto energético nos aplicativos, chegou a conclusão de que os resultados encontrados não são significativos o suficiente para incentivar que os desenvolvedores deixem de usufruir da sua utilidade.

2.3 O sistema iOS

A Apple, responsável pelo sistema em questão, apresenta uma documentação bastante centralizada das suas recomendações de eficiência energética e experiência do usuário [9]. Desta maneira, pode-se facilmente encontrar as práticas recomendadas para os desenvolvedores.

Seguindo a apresentação da documentação oficial disponibilizada, discorrer-se-á sobre os principais desperdícios de energia no aparelho e como os evitar, seguido das recomendações especializadas, separadas por tipos de tarefas ou uso de hardware específico. Por praticidade, as recomendações serão divididas entre uso e requisições de rede(networking) e monitoramento de localização, principalmente por seus usos e recomendações mais intensos e

extensos, seguidos das demais recomendações, que possuem usos mais restritos e recomendações menos detalhadas.

Para melhor entendimento das práticas recomendadas, estas serão classificadas entre recomendações de “Implementação eficiente”, que dependem principalmente do esforço do desenvolvedor em aplicar as técnicas recomendadas, recomendações utilizando “Métodos especializados”, que dependem da aplicação de métodos disponibilizados pela plataforma, recomendações de “Arquitetura econômica”, onde, mais do que utilizar uma biblioteca ou implementar uma recomendação específica, as recomendações se refletem em utilizar uma configuração mais econômica ou mesmo utilizar um design de projeto mais eficiente, e, por fim, de caráter “Informativo”, onde o guia apresenta informações de interesse do desenvolvedor, sem que estas tenham relação, estritamente, com uma redução no consumo de bateria do aparelho.

2.3.1 Informações gerais

Inicialmente, como esperado e como pode ser observado na figura 1, quanto maior o grau de utilização da CPU dos aparelhos, maior será o consumo energético. Além disto, mesmo que não haja uso intensivo da CPU, o seu uso ininterrupto pode impedir que o aparelho entre no modo de descanso, onde apresenta a maior economia de energia.

Figura 1 - Guia iOS - Estado ocioso vs Consumo da CPU.

| | |
|--------------|-----------------------------------|
| Idle | 10x greater power draw over sleep |
| 1% CPU use | 10% greater power draw over idle |
| 10% CPU use | 2x power draw over idle |
| 100% CPU use | 10x power draw over idle |

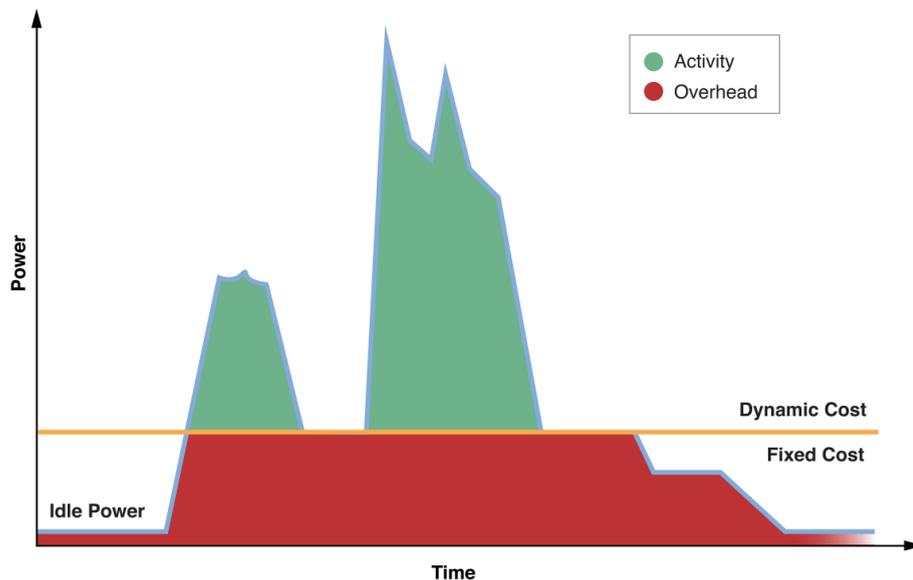
Fonte:

<https://developer.apple.com/library/content/documentation/Performance/Conceptual/EnergyGuide-iOS/FundamentalConcepts.html>

Em segundo lugar, como ferramenta do sistema de economia de energia, boa parte dos hardwares de uso opcional do aparelho, como os de rede, bluetooth ou gps por exemplo, são mantidos desligados ou em estados de baixa energia; Sendo ativados apenas quando seu uso for requisitado por aplicativos. Fora isto, o hardware é mantido energizado por um período de

tempo(overhead) após sua utilização, à espera de novas tarefas, com o objetivo de economizar a energia necessária especificamente para a sua inicialização. Desta maneira, o custo de energização e overhead dos hardwares são chamados de custos fixos dos sistema e seu comportamento esperado pode ser observado na figura 2.

Figura 2: Guia iOS - Custo fixo vs custo dinâmico.



Fonte:

<https://developer.apple.com/library/content/documentation/Performance/Conceptual/EnergyGuide-iOS/FundamentalConcepts.html>

Por fim, as informações gerais da documentação iOS culminam em uma série de recomendações de comportamento esperada dos aplicativos e, desta maneira, dos seus desenvolvedores. Para evitar que haja desperdício conforme as situações citadas anteriormente, os desenvolvedores devem, quando possível, realizar tarefas de acordo com quadro 1.

QUADRO 1: Recomendações gerais - iOS

(continua)

| Proposta | Descrição | Classificação |
|-------------------------------|--|-------------------------|
| Trabalhar menos em background | Avisar ao sistema conclusão da atividade, atrasar downloads quando possível, ... | Implementação eficiente |

QUADRO 1: Recomendações Gerais - iOS

(conclusão)

| | | |
|--|--|-------------------------|
| Trabalhar menos em background | Pausar utilização de hardware quando este for desnecessário (bluetooth, gps...) | Arquitetura econômica |
| Priorização com classes de qualidade de serviço(QoS) | Auxiliam o sistema na execução de tasks de maneira energeticamente eficiente | Métodos especializados |
| Otimização de temporizadores | Utilizar Notificações para atualizações em tempo real e GCD ¹ para sincronização de execuções, evitando o uso de temporizadores | Métodos especializados |
| | Timeouts, intervalos de tolerância e invalidação de timers desnecessários | Arquitetura econômica |
| Reduzir tarefas de entrada e saída(IO) | Diminuir a frequência com implementação de caches, realizar operações em pacotes, operar com arquivos maiores, ... | Implementação eficiente |

Fonte: Adaptado de Energy Efficiency Guide for iOS Apps

1. Grand Central Dispatch - ferramenta para a utilização de threads e execução paralela no sistema iOS

2.3.2. Networking

Conforme citado, a utilização da rede implica em gasto relativo à energização do hardware de rede e de overhead relativo à espera de novas requisições, além desta funcionalidade apresentar gastos variáveis de acordo com as condições de conectividade do aparelho. Uma baixa qualidade de sinal pode levar a requisições mais demoradas ou falhas, que requerem repetição de requisições e gastam energia extra. Em piores casos, ainda, utilização do hardware de dados do celular (3g/4g) gasta energia extra em relação ao uso de Wifi.

Desta maneira, a Apple disponibiliza uma série de ferramentas e sugestões de modo que o desenvolvedor possa minimizar o número de requisições ou atrasar estas para que sejam realizadas em situações mais oportunas para o sistema, conforme descrito no quadro 2.

QUADRO 2: Recomendações de Networking - iOS

| Proposta | Descrição | Classificação |
|-----------|--|-------------------------|
| Minimizar | Reduzir qualidade e tamanho de mídia, comprimir os dados e evitar operações durante más condições de rede | Implementação eficiente |
| | As APIs NSURLCache e NSURLSession são utilizadas para evitar transferências redundantes | Métodos especializados |
| Atrasar | Operações em lote devem ser utilizadas para evitar overhead do hardware de rede | Implementação eficiente |
| | A API NSURLSession permite a criação de sessões em background, que são executadas no momento mais apropriado | Métodos especializados |
| VoIP | Implementação utilizando o framework PushKit. | Métodos especializados |

Fonte: Adaptado de Energy Efficiency Guide for iOS Apps

2.3.3 Monitoramento de localização

De maneira semelhante às tarefas de networking, o monitoramento de localização do aparelho é dependente da utilização do hardware de gps, o que resulta na mesma necessidade de gastos tanto com energização do hardware quanto no overhead de espera por novas requisições.

Além disso, monitoramento de localização possui o fator agravante de ser comumente utilizado de maneira recorrente; Não só a localização atual do aparelho é importante, muitas vezes a distância percorrida ou outros fatores são mais importantes para o usuário, como em apps de corrida, por exemplo. Essa recorrência das atualizações de localizações podem impedir que o aparelho entre em modo de descanso, onde há maior economia de energia.

Dessa maneira, em sua documentação, a Apple disponibiliza uma série de recomendações ao desenvolvedor, para as mais diversas situações, conforme descrito no

quadro 3 a seguir.

QUADRO 3: Recomendações de Monitoramento de localização - iOS

| Proposta | Descrição | Classificação |
|---|--|------------------------|
| Atualizações Rápidas | Uso do método “requestLocation” para atualizações pontuais. | Métodos especializados |
| Interrupção de atualizações e tipos de atividades | Manter atualizações apenas enquanto necessário. Desligar, também, se a precisão desejada estiver indisponível. | Arquitetura econômica |
| | Atribuir um tipo de atividade permite que o sistema defina a frequência e o melhor momento para realizar atualizações | Métodos especializados |
| Interrupção de atualizações e tipos de atividades | "pausesLocationUpdatesAutomatically" possibilita que o sistema pause as atualizações automaticamente, em momentos oportunos. | Métodos especializados |
| | “allowDeferredLocationUpdatesUntilTraveled:timeout:” permite que o sistema atrase atualizações até que uma distância mínima tenha sido percorrida. | Métodos especializados |
| Locais específicos ou mudanças significativas | A monitoração de iBeacons pode ser utilizado para locais específicos | Métodos especializados |
| | Métodos de "visit" podem ser utilizados para monitorar regiões visitadas com frequência ou por longos períodos de tempo | Métodos especializados |
| | Utilizar os métodos para mudanças significativas quando a precisão não é crítica. | Métodos especializados |

Fonte: Adaptado de Energy Efficiency Guide for iOS Apps

2.3.4 Demais recomendações

Por fim, as demais recomendações disponíveis são igualmente referentes à utilização de assets do aparelho com grande custo energético, como a utilização de hardware secundários ou referentes à iluminação e animações na tela do aparelho, um dos componentes de maior consumo energético [22]. No entanto, por serem mais concisas, estas recomendações podem ser agrupadas mais facilmente e estão descritas no quadro 4.

QUADRO 4: Demais recomendações - iOS

(continua)

| Tema | Proposta | Descrição | Classificação |
|--|---|--|-------------------------|
| Recomendações de Gráficos, animações e vídeo | Evitar animações e gráficos alheios | Redução número de views, do uso de opacidade e de animações em telas não visíveis ao usuário | Implementação eficiente |
| | | Utilizar frameworks recomendados (spritekit, metal, scenekit) | Métodos especializados |
| | | Evitar layers adicionais, ao exibir vídeo em tela cheia. | Implementação eficiente |
| Recomendações de Movimento | Reduzir frequência de atualizações | Definir um intervalo eficiente entre atualizações e desabilitar quando não forem necessárias | Arquitetura econômica |
| Recomendações de Notificações | Evitar notificações remotas | Utilizar notificações locais quando possível, evitando gastos com hardware de rede. | Implementação eficiente |
| | Adicionar prioridade à notificações remotas | Priorizar notificações de modo que sejam lançadas em momentos convenientes. | Arquitetura econômica |

QUADRO 4: Demais recomendações - iOS

(conclusão)

| | | | |
|--|--------------------------|---|-------------------------|
| Recomendações de Interação com periféricos | Evitar uso desnecessário | Escanear apenas enquanto necessário, minimizar processamento de aparelhos duplicados, encontrar e observar características e serviços específicos, desconectar quando o uso deixar de ser necessário. | Implementação eficiente |
| | | Utilizar métodos de notificação ao invés de buscar constantemente os valores desejados. | Métodos especializados |
| Recomendações de Modo baixa energia | Adaptação | Além das medidas tomadas pelo próprio sistema, o desenvolvedor deve se responsabilizar por adaptar o aplicativo quando o modo for ativado pelo usuário. | Informativo |

Fonte: Adaptado de Energy Efficiency Guide for iOS Apps

2.4 O sistema Android

Ao contrário da documentação direcionada à economia de energia apresentada por sua concorrente, a Apple, a Google disponibiliza para seus desenvolvedores uma plataforma geral de guidelines de desenvolvimento¹. Dentre os temas apresentados nesta plataforma, otimização do tempo de bateria do aparelho é apresentado apenas como um subtema entre os disponíveis para os interessados neste assunto [10].

Ainda de maneira diferente da documentação disponível para o sistema iOS, ao invés de um guia extensivo de implementações, a Google apresenta uma série de recomendações

para networking, separadas por tipos de requisições, seguidas de recomendações específicas para lidar com os estados de disponibilidade energética do aparelho e suas restrições.

2.4.1 Networking

O guia disponibilizado para o sistema android classifica as operações de networking entre três possíveis definições [19]:

- Iniciada pelo usuário: Requisições diretamente relacionadas à interação com o app, este tipo de requisição tem maior prioridade por estar diretamente relacionada à experiência do usuário.
- Iniciada pelo aplicativo: Dados do aplicativo que não influenciam diretamente a experiência do usuário, possibilitando o planejamento para a realização de tais requisições.
- Iniciada pelo servidor: Dados alterados no servidor do aplicativo. Necessárias apenas quando for do interesse do usuário.

Desta maneira, pode-se sintetizar as recomendações de networking do sistema android de acordo com o quadro 5.

QUADRO 5: Recomendações de networking - Android

(continua)

| Proposta | Descrição | Classificação |
|---------------------------|---|-------------------------|
| Pré-carregamento de dados | Pré-carregar dados do aplicativo em lote para evitar várias requisições independentes | Implementação eficiente |
| Checar conectividade | Evitar requisições quando a rede não estiver disponível | Implementação eficiente |
| Reduzir conexões | Utilizar uma conexão para múltiplos propósitos | Implementação eficiente |

QUADRO 5: Recomendações de networking - Android

(conclusão)

| | | |
|---|--|-------------------------|
| Transferência em lote e atraso de requisições | GCM Network Manager, Job Scheduler, Sync Adapter. Ferramentas do sistema com checagem automática de qualidade da rede. | Métodos especializados |
| Alertar através de notificações | Utilizar Google Cloud Messaging(GCM) ao invés de buscar atualizações constantes do servidor | Métodos especializados |
| Otimizações | Comprimir dados, cache local, otimização cache de pré-carregamento | Implementação eficiente |

Fonte: Adaptado de *Optimizing User-Initiated Network Use*

2.4.2 Outras recomendações

O guia, por fim, disponibiliza soluções para monitoramento do estado da bateria, sugerindo uma diminuição da frequência de atualizações, mas, deixando a caráter do desenvolvedor como tratar estas situações. Primeiramente, são apresentados dois estados possíveis do aparelho, “Doze” e “App Standby”, que foram introduzidos pela google com o objetivo de economizar bateria enquanto o usuário não estiver interagindo diretamente com o aparelho.

Como estes estados em si restringem a liberdade de execução de tarefas pelo app, não são definidas recomendações de economia de energia específicas para esses estados, mas sim como eles afetam os serviços disponíveis para os desenvolvedores. Apesar da existência de informações a respeito de como interagir com o app nestes estados e como conceder maiores liberdades(whitelisting) ao app, estas recomendações fogem ao escopo definido para o projeto e, devido a peculiaridade do sistema, não possui grande valor comparativo junto às recomendações do iOS.

Semelhantemente, o guia android se estende por recomendações ao desenvolvedor de acompanhamento do estado da bateria, estado de carregamento do aparelho ou mesmo o estado e tipo de “ancoragem”. Neste caso, no entanto, o guia não apresenta recomendações explícitas para economia de energia, cabe estritamente ao desenvolver escolher como ampliar

ou restringir a precisão e utilização de features do aplicativo, como o objetivo de equilibrar o tempo de vida do aparelho e a funcionalidade oferecida ao usuário.

Além disso, apesar de não pertencer ao guia anteriormente citado, pode-se encontrar recomendações simples ao longo de sua documentação geral. A documentação relativa à implementação de geofencing, por exemplo, informa a possibilidade de configurar a responsividade de notificações, controlando a frequência de suas atualizações e, desta maneira, aumentando ou diminuindo seu consumo de energia [20].

Dentre esta documentação também estão disponíveis informações sobre configurações para controle de uso de energia das ferramentas empregadas, tais quais as informações disponíveis para observação de mudanças de localização [21]. Neste caso, é recomendado o uso de priorização das requisições, indo do mais baixo custo de energia “PRIORITY_NO_POWER” ao mais alto e, por consequência, mais preciso “PRIORITY_HIGH_ACCURACY”.

Este tipo de informação, no entanto, deixa a critério do desenvolvedor a escolha da qualidade de serviço e não apresenta nenhum tipo de recomendação explícita quanto ao seu uso, possivelmente dificultando o acesso do desenvolvedor à tais ferramentas, podendo ser categorizadas de acordo com o quadro 6.

QUADRO 6: Outras recomendações - Android

(continua)

| Tema | Proposta | Descrição | Classificação |
|-----------------------------------|-----------------------------|---|---------------|
| Estados Doze e App Standby | Informações e Alternativas | Limitações ocasionadas pelos estados descritos | Informativo |
| | | Possíveis métodos para contornar as restrições impostas | Informativo |
| Estados de bateria e carregamento | Informações e Recomendações | Informações complementares de estado de energia, cabendo ao desenvolvedor adaptar o aplicativo. | Informativo |

QUADRO 6: Outras recomendações - Android

(conclusão)

| | | | |
|----------------|-------------|--|-----------------------|
| Geolocalização | Otimizações | Utilizar frequência de atualizações e precisão apropriadas à solução | Arquitetura econômica |
|----------------|-------------|--|-----------------------|

Fonte: Adaptado de Google Developers

3 DESENVOLVIMENTO

Como pode-se observar no capítulo anterior, a autonomia energética de smartphones é um dos pontos mais importantes para a experiência do usuário e existem as mais diversas maneiras de torná-los mais eficientes.

Além disso, também foram apresentadas as recomendações oficiais das principais plataformas de Smartphone, iOS e Android, apesar das otimizações do hardware e do sistema operacional, em geral, serem de responsabilidade das empresas responsáveis se faz necessário que os desenvolvedores também apliquem técnicas eficientes em seus aplicativos; Mais especificamente quase 20% de todos apps avaliados no Google Play contêm reclamações relacionadas ao consumo de energia [5].

Apesar disso, valores concretos, definindo que tipo de ganhos ou perdas resultarão das escolhas de desenvolvimento, não são apresentadas ao desenvolvedor, de modo que ele possa escolher onde conceder ou não a utilização de um método mais eficiente em troca de uma melhor experiência do usuário.

Dessa maneira, buscaremos apoio dos resultados obtidos pela academia com o objetivo de obter informações mais concretas e, possivelmente, mais variadas quanto ao desenvolvimento energeticamente eficiente.

3.1 Metodologia de pesquisa: Revisão Sistemática

Seguindo o objetivo deste projeto de validação das instruções relativas à um desenvolvimento mobile energeticamente eficiente, utilizar-se-há a Revisão Sistemática, uma modalidade de pesquisa considerada do mais alto nível em relação a obtenção de evidências.

A Revisão Sistemática é caracterizada por seu desenvolvimento metódico, consistindo na aplicação de um processo formal de pesquisa, de modo que se possa obter, qualificar e, finalmente, selecionar fontes de dados; Tal processo tem por objetivo inibir o enviesamento das fontes de dados para melhor qualidade dos resultados.

Mais especificamente, os protocolos esperados para uma revisão sistemática esperam que os seguintes passos sejam seguidos:

1. Definição de uma Pergunta de pesquisa.
2. Definição da estratégia de busca
3. Realização da estratégia de busca para obtenção de artigos
4. Seleção entre os artigos obtidos para utilização
5. Extração de dados dos artigos
6. Qualificação dos artigos quanto aos dados extraídos
7. Análise e combinação dos dados obtidos.
8. Avaliação dos resultados em relação à pergunta definida no primeiro passo.

A metodologia de um projeto de revisão sistemática ainda possui a necessidade de diversos subprocessos de definição de protocolos para sua realização, obtenção de conteúdos e análise, além da presença de múltiplos observadores/avaliadores, com o objetivo de evitar um enviesamento através de um processo formal e controlado [11].

O processo de desenvolvimento de uma revisão sistemática, no entanto, necessita um esforço conjunto e de tempo, para que seja obtida a maior quantidade de fontes possível, tornando-se impraticável para o desenvolvimento deste trabalho de conclusão. Desta maneira, de modo que este projeto seja realizável dentro de suas limitações de tempo e equipe, utilizaremos o processo chamado de “mini-review”, ou a mini revisão sistemática.

No processo de desenvolvimento de uma mini-revisão, o processo de busca da literatura deve ser explícito e reproduzível, possuindo o diferencial de utilização de limites arbitrários, para que seja possível uma redução do seu escopo de busca de informações [12]. Tal processo nos permite selecionar, por exemplo, apenas artigos relativamente recentes e pertencentes a publicações com um certo grau de confiabilidade.

Por fim, este projeto, em sua proposta inicial, possuía por objetivo responder às seguintes perguntas de pesquisa:

1. Quais são as principais recomendações, com apoio científico, para cada uma das principais plataformas mobile?
2. Quais os processos de validação para as recomendações apresentadas ?
3. Que níveis de economia de energia pode-se obter utilizando as recomendações

propostas?

4. Como encontrar as recomendações apresentadas e novas recomendações que possam surgir da academia científica?

O processo da revisão, no entanto, requer a definição de uma pergunta principal, guia, que servirá de base para o encaminhamento dos passos seguintes do projeto. Desta maneira, foi levantada a questão “Até que ponto as recomendações de economia de energia feitas por desenvolvedores de aplicativos para as principais plataformas de computação móvel têm respaldo na literatura científica recente na área de Engenharia de Software?”, possibilitando a geração de conhecimento a respeito do que é produzido no mercado e validando sua implementação com os resultados produzidos na academia.

3.1.1 Coleta de dados

3.1.1.1 Planejamento

Inicialmente, como dita o processo de revisão sistemática, fica definida como requisito do processo de aquisição de artigos para esta pesquisa, a utilização de publicações de conferências em língua inglesa, dada a universalização de publicações neste idioma. Além disso, como o próprio processo necessita garantir a confiabilidade de seu conteúdo, serão utilizados apenas artigos pertencentes à conferências com alto h-index, métrica de avaliação da academia baseada nas quantidades de citações dos artigos.

Além disso, a constante evolução das linguagens e tecnologias para o desenvolvimento de aplicativos móveis nos permite selecionar apenas publicações relativamente recentes, mas, mantendo a relevância de seu conteúdo. Para tanto, o ano de 2015 foi escolhido, arbitrariamente, como ano base da pesquisa.

Por fim, serão realizadas um busca quantitativa de artigos relevantes, utilizando palavras chave pré-definidas, e uma análise qualitativa da relevância dos resultados obtidos, através da leitura de seu abstract e, quando necessário, da sua introdução. Desta maneira, fica definido um processo que seguirá os seguintes passos:

1. Seleção de Publicações, em língua inglesa, com alto h5-index

2. Filtragem por conteúdo publicado a partir do ano de 2015
3. Resultados de busca pelos termos: “battery”, “power”, “green” e “energy”
4. Seleção dos artigos finais de acordo com sua relevância para o tema da pesquisa

3.1.1.2 Execução

Entre as conferências possíveis para o estudo, foram selecionadas *ACM/IEEE International Conference on Software Engineering*(ICSE), *ACM SIGSOFT International Symposium on Foundations of Software Engineering*(FSE) e *Mining Software Repositories*(MSR), devido aos seus h5-index/h5-median, 74/102, 50/82 e 43/76 respectivamente [31], e por indicação, dada frequência de artigos relacionados ao tema dentre suas publicações.

Depois de selecionadas as conferências, selecionou-se como base de pesquisa a plataforma DBLP, bibliografia de ciências da computação contendo mais de 4,3 milhões de publicações, por mais de 2,1 milhões de autores [32]. O processo foi realizado utilizando a ferramenta de busca da própria plataforma, que, ao ser utilizada com os termos de busca específicos de de cada conferência(ICSE, MSR, FSE), já apresenta resultados agrupados pelo ano de cada publicação.

Por fim, foi realizada uma busca pelos termos chave selecionados para este trabalho, “battery”, “power”, “green” e “energy”, onde cada termo foi pesquisado manualmente e separadamente, utilizando a ferramenta padrão de busca do navegador de modo que foram encontrados 25 resultados. Além disso, o termo “sustainable” foi excluído dos termos de busca pela alta probabilidade de apresentar resultados falso-positivos, porém, um dos artigos encontrados entre as publicações possuía este termo e foi adicionado como exceção.

3.1.1.3 Resultados Obtidos

Entre os artigos da FSE, foram encontrados um total de sete artigos diferentes, sendo cinco deles referentes à busca pela palavra “Energy” e os dois restantes à busca pela palavra

“Power”, conforme descrito no Quadro 7; Além disso, três artigos foram do ano de 2015, um do ano de 2016 e três do ano de 2017.

Neste resultado, no entanto, foi possível observar, através da análise manual, que os dois artigos referentes à palavra “Power” e do ano de 2017, *The power of "why" and "why not": enriching scenario exploration with provenance* e *Strong agile metrics: mining log data to determine predictive power of software metrics for continuous delivery team*, são casos falso-positivos, ficando, por este motivo, de fora do grupo de interesse deste projeto. Além disso, o artigo intitulado *Improving energy consumption in Android apps*(Improving), apesar estar relacionado ao tema e, desta maneira, não ser um falso-positivo, ficou de fora por ser uma proposta de pesquisa, não um artigo completo.

Quadro 7: Resultados FSE

(continua)

| Ano | Palavra Chave | Artigos |
|------|---------------|--|
| 2015 | Energy | <p>Optimizing energy consumption of GUIs in Android apps: a multi-objective approach.</p> <p>Nyx: a display energy optimizer for mobile web apps.</p> <p>Improving energy consumption in Android apps.¹</p> |
| 2016 | Energy | A portable interface for runtime energy monitoring. |
| 2017 | Power | <p>The power of "why" and "why not": enriching scenario exploration with provenance.²</p> <p>Strong agile metrics: mining log data to determine predictive power of software metrics for continuous delivery teams.²</p> |

Quadro 7: Resultados FSE

(conclusão)

| | | |
|------|--------|--|
| 2017 | Energy | μ Droid: an energy-aware mutation testing framework for Android. |
|------|--------|--|

Observações:

1. Desconsiderado por ser uma proposta de projeto.
2. Resultados falso-positivos

No caso da ICSE, por outro lado, foram encontrados oito resultados correspondentes aos termos de pesquisa selecionados; Devido à ocorrência de um artigo específico, *The Green Lab: Experimentation in Software Energy Efficiency*(Green Lab), contendo duas palavras chaves diferentes, no entanto, foram obtidos sete artigos únicos, conforme descrito no quadro 8.

Dentre os artigos encontrados, seis artigos foram resultados da palavra chave “Energy” e dois artigos foram resultados da palavra chave “Green”. Além destes, a análise manual de todos as publicações permitiu encontrar o artigo *A programming model for sustainable software*, que é relevante para a pesquisa, mas não possui nenhuma das palavras chaves desejadas; Interessantemente, o artigo possui a palavra “Sustainable”, que pode ser relevante ao tema, mas, foi deixada de fora por comumente apresentar resultados falso-positivos.

Destes oito artigos, porém, o artigo *Green Lab* foi categorizado como falso positivo e removido da seleção por ser uma proposta de trabalho, nos moldes do artigo *Improving* da publicação anterior. Além disso, o artigo *StressCloud: A Tool for Analysing Performance and Energy Consumption of Cloud Applications* foi excluído da seleção por, apesar de possuir conteúdo relativo à economia de energia, seu conteúdo não pode ser diretamente ou indiretamente relacionado ao consumo no desenvolvimento de aplicativos mobile.

Quadro 8: Resultados ICSE

(continua)

| Ano | Palavra Chave | Artigos |
|------|---------------|---|
| 2015 | Energy | Optimising Energy Consumption of Design Patterns. |

Quadro 8: Resultados ICSE

(conclusão)

| | | |
|------|--------|--|
| 2015 | Energy | StressCloud: A Tool for Analysing Performance and Energy Consumption of Cloud Applications. ³ The Green Lab: Experimentation in Software Energy Efficiency. ^{1,3} |
| | Green | The Green Lab: Experimentation in Software Energy Efficiency. ^{1,3} |
| | - | A programming model for sustainable software. ² |
| 2016 | Energy | Energy profiles of Java collections classes. Automated energy optimization of HTTP requests for mobile applications. |
| | Green | An empirical study of practitioners' perspectives on green software engineering. |
| 2018 | Energy | EARMO: an energy-aware refactoring approach for mobile apps. |

Observações:

1. Artigo duplicado; Palavras chave *green* e *energy*
2. Artigo exceção da busca, adicionado por análise manual do conteúdo
3. Resultado falso-positivo

Já no caso da MSR, conforme apresentado no quadro 9, dez resultados foram encontrados através da busca utilizando as palavras-chave selecionadas, sendo dois artigos correspondentes à palavra “power”, seis artigos referentes à palavra “energy” e os dois finais à palavra “green”. Pode ser interessante notar, no entanto, que os artigos referentes à palavra green(Going Green: An Exploratory Analysis of Energy-Related Questions e GreenOracle:

estimating software energy consumption with energy measurement corpora.) foram encontrados devido ao nome escolhido para a solução, Going Green e GreenOracle respectivamente; Além disso, ambas as pesquisas também foram resultado da busca utilizando o termo energy, caso que também ocorre em uma das soluções apresentadas na ICSE, The Green Lab.

Desta maneira, oito resultados únicos foram encontrados entre últimos anos da conferência MSR, no entanto, foi encontrado que o artigo *Why Power Laws? An Explanation from Fine-Grained Code Changes*, resultado da palavra power para o ano de 2015, não é relacionado ao desenvolvimento energeticamente eficiente, sendo então um falso-positivo. Por fim, foram selecionados sete artigos dentro dos requisitos necessários para participação nesta pesquisa.

Quadro 9: Resultados MSR

(continua)

| Ano | Palavra Chave | Artigos |
|------|---------------|--|
| 2015 | Power | Why Power Laws? An Explanation from Fine-Grained Code Changes. ¹ |
| | Energy | Mining Energy-Aware Commits. |
| | | Going Green: An Exploratory Analysis of Energy-Related Questions. ² |
| | Green | Going Green: An Exploratory Analysis of Energy-Related Questions. ² |
| 2016 | Power | How android app developers manage power consumption?: an empirical study by mining power management commits. |
| | Energy | GreenOracle: estimating software energy consumption with energy measurement corpora. ² |

Quadro 9: Resultados MSR

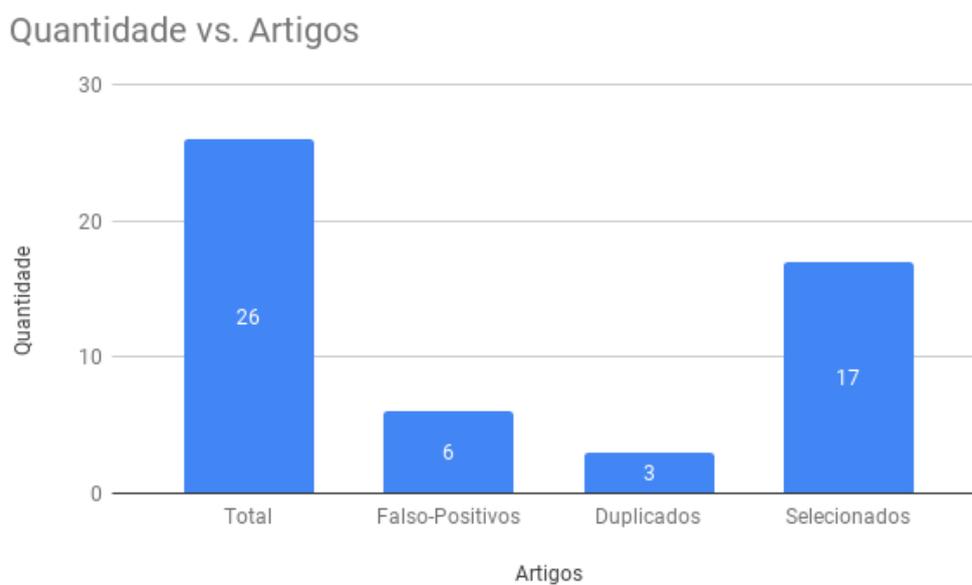
(conclusão)

| | | |
|------|--------|---|
| 2016 | Energy | Characterizing energy-aware software projects: are they different? |
| | Green | GreenOracle: estimating software energy consumption with energy measurement corpora. ² |
| 2017 | Energy | A study on the energy consumption of Android app development approaches. |
| 2018 | Energy | What are your programming language's energy-delay implications? |

Observações:

1. Resultado falso-positivo.
2. Artigos duplicados; Palavras chave green e energy

3.2 Análise do Conteúdo

Figura 3: Disposição dos artigos encontrados

Conforme apresentado no subcapítulo anterior, e descrito na figura 3, dos 26 resultados da pesquisa inicial, seis foram considerados falso-positivos por não apresentar uma conexão forte com o tema desejado ou por não possuir conteúdo de onde seja possível inferir conhecimentos para este projeto; Além destes, três artigos aparecem como resultado duplicado das pesquisas, desta maneira, foram encontrados para análise neste trabalho os seguintes, 17, artigos relacionados:

1. A portable interface for runtime energy monitoring. (*Imes, Bergstrom e Hoffman, 2016*)
2. A programming model for sustainable software. (*Zhu, Lin e Liu, 2015*)
3. A study on the energy consumption of Android app development approaches. (*Oliveira, Oliveira e Castor, 2017*)
4. An empirical study of practitioners' perspectives on green software engineering. (*Manotas, Bird, Zhang, Sheperd, Jaspan, Sadowski, Pollock e Clause, 2016*)
5. Automated energy optimization of HTTP requests for mobile applications. (*Li, Lyu, Gui e Halfond, 2016*)
6. Characterizing energy-aware software projects: are they different? (*Chowdhury e Hindle, 2016*)
7. EARMO: an energy-aware refactoring approach for mobile apps. (*Morales, Saborido, Khomh, Chicano e Antoniol, 2018*)
8. Energy profiles of Java collections classes. (*Hasan, King, Hafiz, Sayagh, Adams e Hindle, 2016*)
9. Going Green: An Exploratory Analysis of Energy-Related Questions. (*Malik, Zhao e Godfrey, 2015*)
10. GreenOracle: estimating software energy consumption with energy measurement corpora. (*Chowdhury e Hindle, 2016*)
11. How android app developers manage power consumption?: an empirical study by mining power management commits. (*Bao, Lo, Xia, Wang e Tian, 2016*)
12. Mining Energy-Aware Commits. (*Moura, Pinto, Ebert e Castor, 2015*)

13. Nyx: a display energy optimizer for mobile web apps. (*Li, Tran e Halfond, 2015*)
14. Optimising Energy Consumption of Design Patterns. (*Noureddine e Rajan, 2015*)
15. Optimizing energy consumption of GUIs in Android apps: a multi-objective approach. (*Linares-Vásquez, Bavota e Bernal-Cárdenas, 2015*)
16. What are your programming language's energy-delay implications? (*Georgiou, Kechagia, Louridas e Spinellis, 2018*)
17. μ Droid: an energy-aware mutation testing framework for Android. (*Jabbarvand e Malek, 2017*)

Estes artigos, no entanto, não possuem um tema único sob qual todos possam ser agrupados. Dentre eles pode-se encontrar, por exemplo, técnicas específicas de desenvolvimento com foco na economia de energia, ferramentas de auxílio ao desenvolvimento “energético-conscientes”, estudos sobre o desenvolvedor e os projetos com interesse neste tema, entre outros.

O processo da revisão sistemática, no entanto, necessita que os artigos sejam comparáveis, de modo que possam ser utilizados para inferir conhecimentos relevantes ao tema desejado. Desta maneira, se faz necessária a definição de categorias que melhor caracterizem e agrupem estes estudos, facilitando a inferência de informações desejadas para este trabalho de conclusão e mini-revisão.

Algumas diferenciações podem ser definidas mais facilmente para os artigos em questão; Os artigos 6, 9, 11 e 12, por exemplo, analisam o comportamento e as características de projetos energeticamente conscientes e/ou de seus desenvolvedores, utilizando de dados obtidos a partir de plataformas amplamente utilizadas, como o Github e o StackOverflow, por exemplo. De certa forma, semelhantemente, mas, sendo um resultado relativamente único nesta pesquisa, o artigo 4 apresenta, através de pesquisas, quantitativa e qualitativa, uma série de informações sobre desenvolvedores e seus respectivos conhecimentos, interesses e dificuldades quanto ao desenvolvimento energeticamente.

Os artigos 1, 2, 5, 7, 10, 13, 14, 15 e 17, por outro lado, apresentam ferramentas, interfaces, ou mesmo, melhorias do ambiente de desenvolvimento que tem por objetivo

auxiliar o desenvolvedor na produção de código e projetos energeticamente mais eficientes, no entanto, esses artigos podem ser separados em categorizações mais eficientes quanto suas abordagens. Os artigos 1, 10 e 17, por exemplo, apresentam soluções de auxílio ao acompanhamento do consumo e detecção de falhas, enquanto os artigos 2, 5, 7, 13, 14 e 15 apresentam soluções de otimização direta do projeto em desenvolvimento.

Os artigos restantes(3, 8 e 16), por fim, apresentam estudos relativos à avaliação de métodos de desenvolvimento, apresentando ao desenvolvedor possibilidades de implementação utilizando certas linguagens ou métodos disponíveis e seus respectivos impactos energéticos. O artigo 16, mais especificamente, apresenta conhecimento generalista, sem um foco específico para o desenvolvimento móvel, mas será mantido por possuir informações relevantes, possuindo informações sobre linguagens de possível utilização no desenvolvimento Android.

Dada a categorização apresentada acima, representada graficamente no Quadro 10 deste projeto, realizada de acordo com o tipo de solução apresentada por cada artigo, realizar-se-há uma segunda categorização, baseada nos tipos de conhecimento inferíveis e sua relação com a pergunta de pesquisa definida para esta mini-revisão.

Nesta segunda, três categorias serão possíveis:

- “Profissional”, onde o conhecimento gerado pelos pesquisadores é baseado em dados obtidos a partir de soluções já empregadas por desenvolvedores de aplicativos num âmbito profissional.
- “Acadêmico”, onde o conhecimento gerado sobre técnicas energeticamente eficientes pelos pesquisadores é inferido a partir de resultados experimentais.
- “Auxiliar”, onde o conhecimento gerado é relevante para o desenvolvedor energeticamente consciente, mas os conhecimentos têm um caráter estritamente auxiliativo, sem uma proposta clara de implementação mais eficiente.

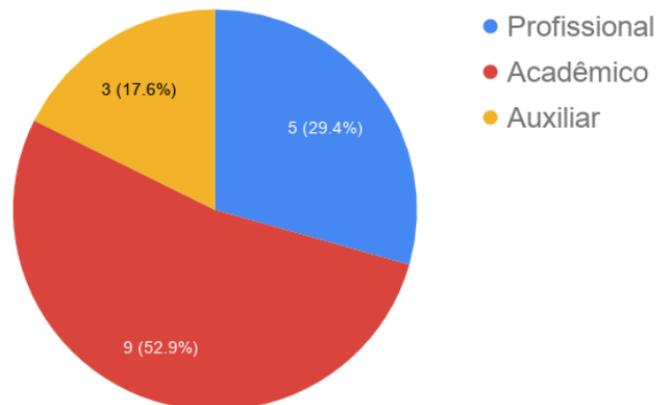
Relacionando esta à categorização anterior, os artigos da categoria de artigos “Descritivos”, apesar das diferentes metodologias, apresentam dados relativos aos desenvolvedores e projetos quanto ao seu relacionamento com o desenvolvimento energeticamente eficiente, portanto serão também os artigos relacionados ao conhecimento

Profissional.

Os artigos categorizados como de “Ferramentas Auxiliares”, por outro lado, apresentam ferramentas de auxílio ao desenvolvedor energeticamente consciente, sem apresentar um dado específico quanto a técnicas mais(ou menos) eficientes, caracterizando-se então com conhecimento Auxiliar.

Dos restantes, os artigos caracterizados como de “Ferramentas de Otimização” podem ser utilizados para a inferência de conhecimentos menos específicos quanto à economia de energia, como, por exemplo, o agrupamento de requisições HTTP [24] ou a utilização de esquemas de cores eficientes para telas OLED [25,31]. Juntando-se a esses, os artigos “Avaliativos”, que geram dados específicos sobre a eficiência de certas técnicas de desenvolvimento, formam os caracterizados pelo conhecimento Acadêmico e culminam na distribuição apresentada na figura 4.

Figura 4: Artigos por categoria de tipo de conhecimento



Fonte: Próprio autor

4 RESULTADOS

Dada a pergunta de pesquisa definida, “Até que ponto as recomendações de economia de energia feitas por desenvolvedores de aplicativos para as principais plataformas de computação móvel têm respaldo na literatura científica recente na área de Engenharia de Software?”, precisamos extrair os dados relevantes de cada categoria.

Primeiramente, com o objetivo de selecionar informações referentes às recomendações descritas em “recomendações de economia de energia feitas por desenvolvedores”, serão extraídas informações dos artigos categorizados como “Profissionais”, onde pode-se observar um maior foco no estudo das técnicas economicamente conscientes/eficientes aplicadas por desenvolvedores de aplicativos e software em geral.

Em sequência, os artigos categorizados como “Acadêmicos” serão utilizados para respaldar cientificamente, ou não, as recomendações apresentadas, visto que os artigos “Auxiliares, apesar de estarem relacionados, não apresentam sugestões de implementações mais eficientes.

4.1 Definições

4.1.1 Recomendações de desenvolvedores

Dentre os artigos selecionados como Profissionais, apesar das diferenças metodológicas entre cada um deles, pode-se inferir que as principais recomendações apresentadas são:

- Controle inteligente do uso de hardware do aparelho.
- Ajustes de bugs responsáveis por aumento do consumo.
- Controle de utilização de recursos de acordo com o estado de carregamento da bateria do aparelho, nível da bateria e modo de economia de energia.

- Utilização de bibliotecas e funcionalidades mais energeticamente eficientes.
- Redução de consumo em estado ocioso.
- Evitar Anti-padrões de Projeto.
- Manipulação de dados mais curtas.
- Redução na frequência e número de chamadas de função.
- Otimização do aplicativo quanto ao uso de “Wake Locks”.
- Redução de frequência ou voltagem do hardware.

4.1.2 Resultados Acadêmicos

Dentre os artigos selecionados como representantes do conhecimento acadêmico, encontram-se os que respaldam seus resultados diretamente, diferenciando explicitamente implementações mais econômicas.

Oliveira, Oliveira e Castor, por exemplo, apresentam diversas situações onde a utilização das linguagens Java, Javascript e C++ para o desenvolvimento Android podem ser mais ou menos energeticamente eficientes, com resultados de implementações híbridas que podem ser até 3 vezes mais eficientes que implementações nativas em Java.

Semelhantemente, Hasan, King e Hafiz apresentam resultados bastante relevantes quanto à utilização de classes de coleção da linguagem Java, demonstrando em soluções open source que a escolha de classes com melhor consumo para a função deseja pode atingir melhorias no consumo energético de até 39%, enquanto uma má escolha de classe pode levar à uma degradação de até 309%. Estes artigos, no entanto, apesar de terem suas especificidades quanto aos métodos estudados, são resumidos em um ponto único de embasamento, da “Aplicação de implementações mais eficientes”.

Em contrapartida, os artigos previamente categorizados como de Ferramentas de otimização não têm por objetivo direto demonstrar a eficiência de uma prática específica, mas, os testes apresentados para demonstração de suas soluções servem de prova para conceitos reais de eficiência energética, qualificando suas práticas como respaldo para práticas de desenvolvimento.

A ferramenta proposta por Li et al, por exemplo, utiliza a prática de agrupamento de requisições HTTP com o objetivo de otimizar o uso do hardware de rede pelo aplicativo, obtendo, em média, melhoras de até 15% do consumo de bateria em nível de aplicação. Semelhantemente, ferramentas como a Nyx, apresentada por Li, Tran e Halfond, e a GEMMA, apresentada por Linares-Vásquez et al, demonstram soluções de otimização do consumo de energia em estados de espera ao utilizar paletas de cores mais frias para interfaces gráficas; obtendo melhorias, em média, de até 66% do consumo energético.

O EARMO, de Morales et al, por outro lado, é uma ferramenta automatizada de auxílio ao refatoramento com o objetivo de remoção de anti-padrões de projeto. Apesar deste tipo de solução não ser apresentada entre os guias oficiais, é apresentada a possibilidade de aumento do tempo de vida do app em até 29 minutos. Da mesma maneira, Nouredine e Rajan apresentam a possibilidade de ganhos de até 25,47% com a otimização de implementações de padrões de projeto.

Além disso, observa-se que a ferramenta apresentada por Jabbarvand e Malek, pertencente aos artigos categorizados como de conhecimento Auxiliar, apesar de não propor melhorias específicas ou apresentar comparações de implementações, apresenta o funcionamento de sua ferramenta com base na proposta de correções de bugs de energia, que serve como embasamento de soluções de desenvolvimento. O mesmo pode ser observado na ferramenta apresentada por Zhu, Lin e Liu, que, apesar de não apresentar resultados específicos de melhoria quanto ao consumo de energia, apresenta uma ferramenta de auxílio ao desenvolvimento consciente quanto ao estado da bateria do aparelho; possibilitando o desenvolvimento com o objetivo de aumentar o seu tempo de uso máximo.

Desta maneira, pode-se afirmar que, dentro do escopo selecionado para este trabalho, as seguintes práticas de desenvolvimento possuem respaldo na literatura científica recente:

- Minimização do uso de hardware do aparelho.
 - Realização de requisições HTTP em pacotes, com o objetivo de minimizar a utilização do hardware de rede
- Otimização de aplicativos em estado ocioso.
 - Utilização de cores frias como modo de economia em telas OLED
- Aplicação de implementações mais eficientes.

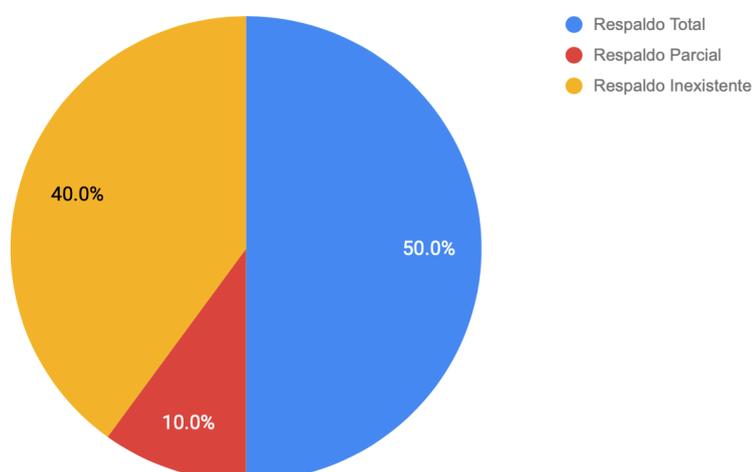
- Utilização de estruturas de dados otimizadas, de acordo com a função a ser executada
- Utilização de linguagens de programação otimizadas para a tarefa desejada
- Controle de utilização de recursos de acordo com o estado do aparelho.
- Otimização de implementação de padrões de projeto.
- Correção de anti-padrões de projeto.
- Correção de bugs de energia.

4.2 Respostas

4.2.1 Recomendações quanto ao seu respaldo na academia

A partir das informações inferidas, pode-se, finalmente, observar os relacionamentos entre as propostas discutidas e implementadas no mercado profissional de desenvolvimento mobile com o conhecimento gerado na academia, assegurando resultados de redução no consumo de energia.

Na comparação dos conhecimentos, percebe-se que algumas propostas apresentadas possuem equiparação automática com o conhecimento acadêmico, enquanto outras possuem apenas respaldo parcial ou nenhum respaldo dentre os artigos encontrados nas publicações e nos anos selecionados; As proposições de desenvolvedores apresentam uma distribuição de respaldo conforme apresentado na Figura 5.

Figura 5. Distribuição das recomendações quanto ao seu respaldo acadêmico

Mais especificamente, as propostas de “Ajustes de bugs de energia”, “Redução de consumo em estado ocioso”, “Evitar Anti-Padrões de projeto”, “Utilização de bibliotecas e funcionalidades mais energeticamente eficientes.” e “Controle de utilização de recursos de acordo com o estado de carregamento da bateria do aparelho, nível da bateria e modo de economia de energia.” apresentam respaldo total por possuírem contexto condizentes com os da academia.

“Controle inteligente do uso de hardware do aparelho”, por outro lado, apresenta um resultado parcial pois, apesar de existir a demonstração da melhoria por redução do número de requisições totais no aplicativo, o uso inteligente envolve uma série de outros cuidados relativos à utilização dos mais diversos hardwares do aparelho.

Por fim, as recomendações “Manipulação de dados mais curtas.”, “Redução na frequência e número de chamadas de função.”, “Otimização do aplicativo quanto ao uso de ‘Wake Locks’” e “Redução de frequência ou voltagem do hardware.” não apresenta respaldo entre os artigos encontrados.

4.2.2 Respaldos quanto às recomendações oficiais

Complementarmente aos resultados apresentados pela resolução da pergunta de pesquisa, quanto ao embasamento das propostas de desenvolvedores em cima do

conhecimento gerado pela academia, pode-se utilizar o conhecimento disponível nos guias oficiais de cada plataforma para gerar uma avaliação mais ampla quanto à validade formal de recomendações apresentadas nos ambientes de desenvolvimento.

Como pode-se observar no quadro 11, apesar da diferença entre o número de recomendações disponíveis e dos resultados factuais que podem ser esperados, grande parte da cobertura disponível nos guias oficiais também está disponível entre o conhecimento gerado pela academia. As exceções quanto à presença na produção científica recente são a breve recomendação na utilização inteligente de transações I/O, presente na documentação do iOS, e as recomendações específicas para a utilização de *Wake Locks*, ferramenta unicamente utilizada nos sistemas Android.

QUADRO 11: Respaldos referentes à academia e às documentações oficiais. (continua)

| Artigo | Respaldos | | |
|---|----------------------------------|--|--|
| | Academia | iOS | Android |
| Controle inteligente do uso de hardware do aparelho | Parcial(Redução de uso do Wi-Fi) | Total | Total |
| Ajustes de bugs responsáveis por aumento do consumo. | Total | Inexistente | Inexistente |
| Controle de utilização de recursos de acordo com o estado de carregamento da bateria do aparelho, nível da bateria e modo de economia de energia. | Total | Parcial(Modo econômico) | Total |
| Utilização de bibliotecas e funcionalidades mais energeticamente eficientes. | Total | Parcial(Bibliotecas) | Parcial(Bibliotecas) |
| Redução de consumo em estado ocioso. | Parcial(Cores da Tela) | Parcial(Evitar trabalho em background) | Parcial(Evitar trabalho em background) |

QUADRO 11: Respaldos referentes à academia e às documentações oficiais. (conclusão)

| | | | |
|---|-------------|------------------------------|-------------|
| Evitar Anti-padrões de projeto. | Total | Inexistente | Inexistente |
| Manipulação de dados mais curtas. | Inexistente | Total | Inexistente |
| Redução na frequência e número de chamadas de função. | Inexistente | Inexistente | Inexistente |
| Otimização do aplicativo quanto ao uso de “Wake Locks”. | Inexistente | Não Relacionado ¹ | Total |
| Redução de frequência ou voltagem do hardware. | Inexistente | Inexistente | Inexistente |

Observações:

1. Wake Lock é uma ferramenta utilizada especificamente na plataforma Android

Interessantemente, apesar do caráter oficial esperado das recomendações da Apple e da Google, e mesmo como a reduzida quantidade de fontes inerente da aplicação da mini-revisão, a produção da academia apresenta os novos conhecimentos a respeito de bugs de energia e de correção de anti-padrões de desenvolvimento dentro do tema de desenvolvimento mobile energeticamente eficiente.

Por fim, se faz importante notar que, mesmo com a união dos respaldos provenientes do conhecimento recente gerado na academia e conhecimento disponibilizados em guias oficiais para as principais plataformas mobile, duas recomendações dos desenvolvedores não puderam ser respaldadas.

4.2.3 Incompletude das recomendações oficiais

Conforme pode-se observar, apesar de escopo reduzido utilizado para esta pesquisa,

algumas das indicações respaldadas através de experimentos científicos não encontram recomendações semelhantes dentre a documentação oficial.

Mais especificamente, apesar dos guias oficiais incentivarem a utilização de métodos especializados mais econômicos, a produção da academia também demonstra que a utilização de uma linguagem de programação diferente ou a utilização de uma estrutura de dados menos eficiente pode causar um impacto não trivial quanto ao consumo de energia.

Além disso, a utilização de telas OLED, tecnologia já existente em vários aparelhos que utilizam o sistema Android e, recentemente, anunciada para os mais novos aparelhos da Apple, permite que o desenvolvedores implementem telas com esquemas de cores mais econômicas.

Por fim, os artigos acadêmicos oferecem soluções de otimização de padrões de projeto, correções de anti-padrões e ferramentas de validação de suítes de teste quanto a sua habilidade de encontrar de bugs de energia, situações que podem ser observadas durante a produção de um aplicativo, mas as plataformas oficiais não apresentam nenhuma indicação destes problemas ao desenvolvedor.

5 CONSIDERAÇÕES FINAIS

Dada a importância da autonomia energética de aplicações mobile, este trabalho teve por objetivo analisar as principais práticas energeticamente eficientes de desenvolvimento. Como apresentado, no entanto, apesar de apresentar conteúdo relativamente robusto, as recomendações oficiais de economia de energia dos principais ambientes mobile não apresenta uma relação clara quanto aos ganhos relativos à aplicação de suas técnicas; Além disso, no ambiente profissional de desenvolvimento de aplicações, a maior parte das dúvidas de implementação e otimização acabam sendo sanadas com o uso de ferramentas como o StackOverflow, onde a veracidade das soluções dependem principalmente de um esforço de grupo, sem comprovações formais de sua validade.

Desta forma, este projeto buscou responder, através da aplicação da metodologia de mini-revisão sistemática, até que ponto as soluções de desenvolvimento energeticamente eficientes, apresentadas por desenvolvedores, possuem respaldo na bibliografia científica recente da área de Engenharia de Software.

Seguindo o processo da mini-revisão, foi definida a pergunta de pesquisa, seguida pela definição das fontes e dos métodos de busca de conteúdo. Utilizando a implementação dos métodos definidos, foram encontrados 26 resultados, dos quais 17 foram selecionados para a realização final do processo. Destes selecionados, foi possível separar o conhecimento de forma que obteve-se dados tanto em relação aos métodos aplicados pelos desenvolvedores mobile quanto os dados necessários para validar seu respaldo acadêmico.

Concluindo o objetivo da mini-revisão sistemática, observou-se que pelo menos 60% das propostas encontradas apresentam respaldo parcial, em relação ao conhecimento gerado pela comunidade científica. Ao expandir o material de base, adicionando o conteúdo recomendado pelos guias oficiais de cada plataforma, foi encontrado, por fim, um embasamento de pelo menos 80% entre as propostas selecionadas.

Além disso, observou-se que, apesar do uso de um escopo reduzido neste projeto, uma boa parte das soluções propostas pela academia estão ausentes do conteúdo disponível nos guias oficiais.

5.1 Ameaças à validade

Apesar do escopo reduzido existente na metodologia de produção de uma mini-revisão, uma série de recomendações de desenvolvimento foram definidas e devidamente respaldadas de acordo com as metodologias energeticamente eficientes validadas pela academia. Esta redução de escopo, no entanto, atua como um possibilitador da existência de técnicas definidas por desenvolvedores mas sem a devida validação da economia.

No caso da otimização do uso de “wake locks”, por exemplo, não foi possível encontrar respaldo dentre os artigos encontrados com as regras definidas para a mini-revisão, no entanto, a busca minuciosa por artigos relacionados já revela a existência de resultado falso-negativo encontrado entre as publicações do FSE 2016 [44]. Desta maneira, uma busca minuciosa, com a possível expansão dos termos e do escopo de pesquisa, pode apresentar um resultado completamente diferente quanto ao respaldo das soluções apresentadas pelos desenvolvedores.

Além disso, uma pesquisa mais abrangente quanto as sugestões de economia de energia provenientes das comunidades de desenvolvedores devem gerar uma quantidade maior e mais diversificada de possibilidades para a validação e possível aplicação no desenvolvimento de aplicativos móveis.

5.2 Trabalhos Futuros

Além dos problemas citados no subcapítulo anterior, a constante evolução das linguagens e ambientes de programação gera a necessidade da constante atualização dos conhecimentos no ramo. Apesar do recente lançamento da linguagem Kotlin para desenvolvimento android, não existe material de comparação da sua performance em relação às outras linguagens disponíveis para o desenvolvimento, por exemplo.

Desta maneira, uma Revisão Sistemática completa, com o seu minucioso processo de

busca pela maior quantidade de fontes possível, pode ser o meio mais eficiente de encontrar a maior quantidade de métodos energeticamente eficientes respaldados, de modo que nenhum método sugerido por desenvolvedores fique sem cobertura.

A incompletude de recomendações observável nos guias oficiais, também, representa uma possível necessidade dos desenvolvedores quanto a encontrar todas as possíveis técnicas documentadas, oficial ou academicamente, de economia de energia. A centralização de todo o material produzido neste tema, pode ser grande serventia para o desenvolvedor energeticamente consciente.

Por fim, é observável que a grande maioria dos artigos científicos nesse ramo são referente estritamente ao desenvolvimento Android, de modo que o desenvolvimento iOS fica sujeito apenas às recomendações oficiais ou à implementação de propostas para o sistema concorrente, na expectativa de que eles funcionem de forma semelhante. Desta maneira, um grande vazio de conhecimento sobre a plataforma se encontra em aberto para o desenvolvimento de trabalhos futuros.

Referências

- [1] GIBBS, S. "Mobile web browsing overtakes desktop for the first time". Disponível em: <<https://www.theguardian.com/technology/2016/nov/02/mobile-web-browsing-desktop-smartphones-tablets>>. Acessado: Setembro 2017.
- [2] GOMES, H. S. "Smartphone passa PC e vira aparelho nº 1 para acessar internet no Brasil". Disponível em: <<http://g1.globo.com/tecnologia/noticia/2016/04/smartphone-passa-pc-e-vira-aparelho-n-1-para-acessar-internet-no-brasil.html/>>. Acessado: Setembro 2017.
- [3] HACKERNOON. **How Much Time Do People Spend on Their Mobile Phones in 2017?** Disponível em: <<https://hackernoon.com/how-much-time-do-people-spend-on-their-mobile-phones-in-2017-e5f90a0b10a6>>. Acessado: Novembro, 2017
- [4] WANG, D. "Meeting green computing challenges". In: **Proceedings of 10th electronics packaging technology conference (EPTC 2008)**, Singapore. Acessado: Novembro 2017
- [5] WILKE, C.; RICHLI, S.; GOTZ, S.; PIECHNICK, C.; ASSMANN, U. "Energy Consumption and Efficiency in Mobile Applications: A User Feedback Study". **2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing**. Acessado: Novembro 2017
- [6] OLIVEIRA, W.; OLIVEIRA, R; CASTOR, F. "A Study on the Energy Consumption of Android App Development Approaches". **Proceedings of the 14th International Conference on Mining Software Repositories**. Buenos Aires, Argentina, 2017. Acessado: Novembro 2017
- [7] FUENTES, L.; GÁMEZ, N.; PINTO, M. "HADAS Green Assistant: designing energy-efficient applications". CoRR, abs/1612.08095. Disponível em: <<https://arxiv.org/abs/1612.08095>>. Acessado: Novembro 2017
- [8] JAMES, V. "99.6 percent of new smartphones run Android or iOS" <<https://www.theverge.com/2017/2/16/14634656/android-ios-market-share-blackberry-2016>>. Acessado: Setembro 2017
- [9] "Energy Efficiency and the User Experience". Disponível em: <<https://developer.apple.com/library/content/documentation/Performance/Conceptual/EnergyGuide-iOS/index.html>>. Acessado: Setembro 2017

- [10]Google Developers. **Optimizing Battery Life**. Disponível em:
<<https://developer.android.com/training/monitoring-device-state/index.html>>. Acessado: Outubro 2017
- [11]BIOLCHINI, J.; MIAN P.G.; NATALI, A.C.; TRAVASSOS G. “Systematic Review in Software Engineering” <<https://www.cos.ufrj.br/uploadfile/es67905.pdf>>, Acessado: Agosto 2018
- [12] GRIFFITHS, P. (2002). “Evidence informing practice: introducing the mini-review. British journal of community nursing”. Acessado: Agosto 2018
- [13]Swedish Electromobility Centre. **Emerging Battery Technologies towards 2025**, 2015. Disponível em:
http://hybridfordonscentrum.se/wp-content/uploads/2016/01/T3.11-Batteries-present-and-future-challenges_Final-report.pdf. Acessado: Outubro, 2017.
- [14]STRATEGY ANALYTICS. **Camera and Battery Features Continue to Drive Consumer Satisfaction of Smartphones in US**, 2017. Disponível em:
<https://www.strategyanalytics.com/strategy-analytics/news/strategy-analytics-press-releases/strategy-analytics-press-release/2017/05/31/camera-and-battery-features-continue-to-drive-consumer-satisfaction-of-smartphones-in-us>. Acessado: Outubro, 2017.
- [15] SARWAR, M.; SOOMRO, T. R.. “Impact of Smartphone’s on Society”. **European Journal Of Scientific Research**. [s.i], p. 216-226. mar. 2013. Disponível em:
<https://www.researchgate.net/profile/Tariq_Soomro/publication/236669025_Impact_of_Smartphone's_on_Society/links/00b7d518ce4de2509d000000/Impact-of-Smartphones-on-Society.pdf>. Acessado: Novembro, 2017
- [16] ZOLOTAREVA, K. **The history of the app stores: from 2008 to 2017**. 2017. Disponível em:
<<https://thetool.io/2017/evolution-app-stores-infographic>>. Acessado: Novembro, 2017.
- [17]”Global mobile OS market share in sales to end users from 1st quarter 2009 to 2nd quarter 2018”. Disponível em:
<<https://www.statista.com/statistics/266136/global-market-share-held-by-smartphone-operating-systems/>>. Acessado: Outubro 2018
- [18]”What operating system does your smartphone use?” Disponível em:
<<https://www.statista.com/statistics/716053/most-popular-smartphone-operating-systems-in-us/>>. Acessado: Dezembro 2017

- [19]Google Developers. **Reducing Network Battery Drain**. Disponível em:
<<https://developer.android.com/topic/performance/power/network/index.html>>. Acessado: Outubro 2017
- [20]Google Developers. **Creating and Monitoring Geofences**. Disponível em:
<<https://developer.android.com/training/location/geofencing.html#BestPractices>>. Acessado: Dezembro 2017
- [21]Google Developers. **Changing Location Settings**. Disponível em:
<<https://developer.android.com/training/location/change-location-settings.html>>. Acessado: Dezembro 2017
- [22]TAWALBEH, M.; EARDLEY, A.; TAWALBEH, L. “Studying the Energy Consumption in Mobile Devices”. **Procedia Computer Science**, [s.l.], v. 94, p.183-189, 2016. Elsevier BV.
<http://dx.doi.org/10.1016/j.procs.2016.08.028>. Acessado: Dezembro 2017
- [23]ATANÁSIO, P. **Power Profiling & Measurement**. [s.i.]: Inlocomidia, 2017. Color. Acessado: Dezembro 2017
- [24]LI, D. et al. “Automated energy optimization of HTTP requests for mobile applications”. **Proceedings Of The 38th International Conference On Software Engineering**, Austin, Texas, p.249-260, maio 2016.
[doi>10.1145/2884781.2884867]. Acessado: Outubro 2018
- [25]LINARES-VÁSQUEZ M. et al. “Optimizing energy consumption of GUIs in Android apps: a multi-objective approach”. **In Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering (ESEC/FSE 2015)**. ACM, New York, NY, USA, 143-154. Acessado em: Outubro 2018
- [26]PINTO, G.; LIU, K.; CASTOR, F.; LIU, Y. D. “A Comprehensive Study on the Energy Efficiency of Java’s Thread-Safe Collections”, **2016 IEEE International Conference on Software Maintenance and Evolution (ICSME)**, Raleigh, NC, 2016, pp. 20-31. Acessado em: Outubro 2018
- [27]HASAN, S. et al. Energy profiles of Java collections classes. **In Proceedings of the 38th International Conference on Software Engineering (ICSE '16)**. ACM, New York, NY, USA, 225-236. Acessado em: Outubro 2018
- [28] Google Scholar. **Top Publications in Software Systems**. Disponível em:
https://scholar.google.com/citations?view_op=top_venues&hl=en&vq=eng_softwaresystems Acessado em: Outubro 2018.
- [29] DBLP. Disponível em: <https://dblp.org/faq/What+is+dblp.html> Acessado em: Agosto de 2018

- [30]JIMES, C.; BERGSTROM, L.; HOFFMANN, H. “A Portable Interface for Runtime Energy Monitoring”. **In Proceedings of the 2016 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering (FSE 2016)**. ACM, New York, NY, USA, 968-974. Acessado em: Outubro 2018
- [31]LI, D.; TRAN, A.; HALFOND, W. “Nyx: a display energy optimizer for mobile web apps”. **In Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering (ESEC/FSE 2015)**. ACM, New York, NY, USA, 958-961. Acessado em: Outubro 2018
- [32]JABBARVAND, R.; MALEK, S. “ μ Droid: an energy-aware mutation testing framework for Android”. **In Proceedings of the 2017 11th Joint Meeting on Foundations of Software Engineering (ESEC/FSE 2017)**. ACM, New York, NY, USA, 208-219. Acessado em: Outubro 2018
- [33]MORALES, R. et al. “EARMO: An Energy-Aware Refactoring Approach for Mobile Apps”. **In Proceedings of the 40th International Conference on Software Engineering (ICSE '18)**. ACM, New York, NY, USA, 59-59. Acessado em: Outubro 2018
- [34]MANOTAS, I. et al. “An Empirical Study of Practitioners’ Perspectives on Green Software Engineering”. **In Proceedings of the 38th International Conference on Software Engineering (ICSE '16)**. ACM, New York, NY, USA, 237-248. Acessado em: Outubro 2018.
- [35]NOUREDDINE, A.; RAJAN, A. “Optimising energy consumption of design patterns”. **In Proceedings of the 37th International Conference on Software Engineering - Volume 2 (ICSE '15)**, Vol. 2. IEEE Press, Piscataway, NJ, USA, 623-626. Acessado em: Outubro 2018.
- [36]ZHU, H. S.; LIN, C.; LIU, Y. D. “A programming model for sustainable software”. **In Proceedings of the 37th International Conference on Software Engineering - Volume 1 (ICSE '15)**, Vol. 1. IEEE Press, Piscataway, NJ, USA, 767-777. Acessado em: Outubro 2018.
- [37]CHOWDHURY, A. S.; HINDLE, A. “Characterizing energy-aware software projects: are they different?”. **In Proceedings of the 13th International Conference on Mining Software Repositories (MSR '16)**. ACM, New York, NY, USA, 508-511. Acessado em: Outubro 2018
- [38]MALIK, H.; ZHAO, P.; GODFREY, M. “Going green: an exploratory analysis of energy-related questions”. **In Proceedings of the 12th Working Conference on Mining Software Repositories (MSR '15)**. IEEE Press, Piscataway, NJ, USA, 418-421. Acessado em: Outubro 2018

- [39]CHOWDHURY, A. S.; HINDLE, A. “GreenOracle: estimating software energy consumption with energy measurement corpora”. **In Proceedings of the 13th International Conference on Mining Software Repositories (MSR '16)**. ACM, New York, NY, USA, 49-60. Acessado em: Outubro 2018
- [40]BAO, L.; LO, D.; XIA, X.; WANG, X.; TIAN, C. “How android app developers manage power consumption?: an empirical study by mining power management commits”. **In Proceedings of the 13th International Conference on Mining Software Repositories (MSR '16)**. ACM, New York, NY, USA, 37-48. Acessado em: Outubro 2018
- [41]MOURA, I.; PINTO, G.; EBERT, F.; CASTOR, F. “Mining energy-aware commits”. **In Proceedings of the 12th Working Conference on Mining Software Repositories (MSR '15)**. IEEE Press, Piscataway, NJ, USA, 56-67. Acessado em: Outubro 2018
- [42]GEORGIU, S.; KECHAGIA, M.; LOURIDAS, P; SPINELLIS, D. 2018. “What are your programming language's energy-delay implications?”. **In Proceedings of the 15th International Conference on Mining Software Repositories (MSR '18)**. ACM, New York, NY, USA, 303-313. Acessado em: Outubro 2018
- [43]SAHIN, C.; TORNQUIST, P.; MCKENNA, R.; PEARSON, Z.; CLAUSE, J. "How Does Code Obfuscation Impact Energy Usage?", **2014 IEEE International Conference on Software Maintenance and Evolution**, Victoria, BC, 2014, pp. 131-140. Acessado: Dezembro 2017
- [44]LIU, Y.; XU, C.; CHEUNG, S.; TERRAGNI, V. “Understanding and detecting wake lock misuses for Android applications.”, **In Proceedings of the 2016 24th ACM SIGSOFT International Symposium on Foundations of Software Engineering (FSE 2016)**. ACM, New York, NY, USA, 396-409. Acessado: Dezembro 2018

Apêndice

QUADRO 10: Categorização de Artigos

| Categoria | Artigo | Proposta |
|-----------------------------------|--|---|
| Artigos Descritivos | An empirical study of practitioners' perspectives on green software engineering. | Estudo sobre a visão de desenvolvedores experientes a respeito do desenvolvimento energeticamente consciente. |
| | Characterizing energy-aware software projects: are they different? | Caracterização de projetos energeticamente conscientes, através da análise de projetos no GitHub |
| | Going Green: An Exploratory Analysis of Energy-Related Questions. | Avaliar as dúvidas e conhecimentos de desenvolvedores através da análise de perguntas no StackOverflow |
| | How android app developers manage power consumption? | Caracterização de projetos e abordagens de desenvolvimento energeticamente-eficiente. |
| | Mining Energy-Aware Commits. | Caracterização de abordagens de desenvolvimento energeticamente-eficiente. |
| Artigos de Ferramentas Auxiliares | A portable interface for runtime energy monitoring | Interface de abstração para o acesso aos dados de ferramentas de monitoramento de energia. |
| | GreenOracle: estimating software energy consumption with energy measurement corpora. | Ferramenta de avaliação de código quanto ao seu consumo energético com uma abordagem de Big Data |
| | μDroid: an energy-aware mutation testing framework | Ferramenta de validação de conjuntos de testes quanto à |

| | | |
|--------------------------------------|--|--|
| | for Android. | sua capacidade de revelar falhas energéticas. |
| Artigos de Ferramentas de Otimização | A programming model for sustainable software. | Framework de auxílio à utilização de recursos do aparelho dentro de um <i>budget</i> de energia e/ou temperatura pré-definidos |
| | Automated energy optimization of HTTP requests for mobile applications. | Investigação e agrupamento de requisições automatizados com o objetivo de otimizar a utilização de hardware de rede. |
| | EARMO: an energy-aware refactoring approach for mobile apps. | Ferramenta analisa o consumo energético da aplicação, encontra a ocorrência de anti-padrões de software e gera possibilidades de refatoramento com o objetivo de melhorar o consumo e removê-los. |
| | Nyx: a display energy optimizer for mobile web apps. | Ferramenta analisa e gera versões mais energeticamente eficientes de aplicações web quanto a sua exibição em telas OLED |
| | Optimising Energy Consumption of Design Patterns. | Avaliação e otimização automatizados em tempo de compilação para otimização energética de padrões de projeto. |
| | Optimizing energy consumption of GUIs in Android apps: a multi-objective approach. | Ferramenta analisa designs e hierarquias de telas de aplicativos, gerando diversas versões energeticamente mais eficientes para telas OLED, levando em consideração o contraste das cores e a adequação aos usuários.. |

| | | |
|---------------------|--|---|
| Artigos Avaliativos | A study on the energy consumption of Android app development approaches. | Avaliação do consumo energético de desenvolvimento Android nas linguagens Java, Javascript e C++ |
| | Energy profiles of Java collections classes. | Avaliação do consumo energético de diversas classes dos tipos <i>List</i> , <i>Map</i> e <i>Set</i> |
| | What are your programming language's energy-delay implications? | Avaliação de diversas linguagens de programação quanto ao seu <i>Energy Delay Product</i> , razão ponderada entre o consumo de energia e performance. |