

## Uma Avaliação da Sustentabilidade de Softwares com Foco em Aplicativos Móveis

Aline R. Tonini, Milena R. S. Marques, Lisane B. de Brisolara

Centro de Desenvolvimento Tecnológico – Universidade Federal do Rio Grande do Sul  
Pelotas – RS – Brazil

{arttonini, mrsmarques, lisane}@inf.ufpel.edu.br

**Abstract.** *This paper discusses the software sustainability evaluation and proposes a quantitative evaluation approach for mobile applications. The proposed approach considers sustainability as a transversal aspect into the quality model and relates this aspect to the others characteristics from the ISO/IEC 25010:2011 standard. This way, software metrics can be used to estimate the software quality attributes, as well as to evaluate its sustainability. Case studies are presented, in which maintainability, portability and efficiency are evaluated and used to estimate the sustainability of mobile applications.*

**Resumo.** *Este artigo discute a avaliação de sustentabilidade de software e propõe uma abordagem para avaliação quantitativa de aplicativos móveis. A avaliação proposta considera a sustentabilidade como um aspecto transversal no modelo de qualidade e discute a relação da sustentabilidade com as demais características consideradas pela norma ISO/IEC 25010:2011. Desta forma métricas de software podem ser empregadas para estimar características desejadas ao produto de software, bem como para avaliar sua sustentabilidade. Estudos de caso envolvendo aplicativos móveis são apresentados, nos quais manutenibilidade, portabilidade e eficiência são avaliadas e usadas para estimar a sustentabilidade destes aplicativos.*

### 1. Introdução

A tecnologia faz parte do cotidiano das pessoas e das organizações, propiciando uma série de facilidades, dentre elas, um aumento na produtividade. No entanto, o emprego da tecnologia causa também impactos negativos no meio ambiente, por exemplo, aumentando a demanda por recursos energéticos para manter equipamentos funcionando. No contexto atual de avanços tecnológicos, temas como desenvolvimento sustentável e sustentabilidade estão recebendo atenção especial da sociedade.

As áreas de computação e da tecnologia da informação (TI), como principais responsáveis por grande parte dos avanços tecnológicos das últimas décadas, estão preocupadas com o desenvolvimento sustentável. Segundo a WCED (*World Commission on Environment and Development*), “desenvolvimento sustentável é aquele capaz de suprir as necessidades da geração atual, sem comprometer a capacidade de atender às necessidades das futuras gerações” [WCED 1987]. Desta preocupação surge a Engenharia de Software Sustentável (ESS), que tem como objetivo o desenvolvimento sustentável de softwares considerando três pilares (econômico, social e ambiental), e propõe modelos de processo sustentáveis [Naumann et. al 2011]. Porém, possuir um

processo de software sustentável não garante que o produto final produzido pelo mesmo também será sustentável. Portanto, é necessário estudar a avaliação dos produtos de software de forma a determinar o grau de sustentabilidade dos mesmos.

As normas ISO de qualidade de software [ISO 2014] que definem atributos ou características desejadas para um software de qualidade, incluem a eficiência, assim, abrangendo a questão energética. No entanto, a sustentabilidade não se restringe a questão energética e não é totalmente coberta pelas normas existentes, como discutido em [Calero, Moraga e Bertoa 2013][Naumann et al. 2013].

Atualmente, dispositivos móveis estão mudando hábitos e acompanham as pessoas no seu dia-a-dia, além de representarem um importante mercado de equipamentos eletrônicos. Este crescente uso de *smartphones* e *tablets*, e consequentemente de aplicativos para os mesmos, tem impacto ambiental, sobretudo neste mercado altamente competitivo e inovador [Arndt, Dziubaczyk e Mokosch 2014]. No domínio de aplicativos móveis, a preocupação com a eficiência energética é bem discutida. Por exemplo, em [Tonini et al. 2014] uma avaliação da eficiência energética é realizada, porém, sem considerar sustentabilidade e outras características de qualidade. Outro grupo discute a sustentabilidade para aplicativos móveis desenvolvidos para as plataformas da Apple e da Microsoft em [Arndt, Dziubaczyk e Mokosch 2014], mas sem referenciar o modelo de qualidade de software da ISO.

Este artigo propõe uma abordagem para avaliação da sustentabilidade de software com foco em aplicativos móveis. Nesta abordagem, a sustentabilidade é tratada como um aspecto transversal do modelo de qualidade, cujos pilares podem ser relacionados com os atributos de qualidade de produtos da ISO/IEC 25010:2011. Desta forma, métricas de software usadas para avaliar características e subcaracterísticas do software podem ser empregadas para avaliar também sua sustentabilidade. Para demonstrar nossa abordagem, são apresentados estudos de casos de avaliação de sustentabilidade de aplicativos móveis.

O artigo está organizado da seguinte maneira: a Seção 2 revisa a fundamentação teórica, enquanto a Seção 3 discute trabalhos relacionados; a abordagem proposta é apresentada na Seção 4 e os estudos de caso na Seção 5 e por fim, a Seção 6 apresenta conclusões e discute alguns trabalhos futuros.

## 2. Fundamentos

A qualidade de um produto de software é definida por normas e padrões. A norma internacional ISO/IEC 25010:2011 [ISO 2011], define uma série de características esperadas aos produtos de software, as quais são organizadas em um modelo hierárquico de dois níveis. No primeiro nível, encontram-se as características: Funcionalidade, Eficiência, Compatibilidade, Usabilidade, Confiabilidade, Segurança, Manutenibilidade e Portabilidade. No segundo nível, as características são decompostas em subcaracterísticas.

Para determinar o quanto um software possui ou considera essas características, é necessário avaliá-lo através de métricas. As métricas de software podem ser classificadas como estáticas ou dinâmicas, dependendo da forma como estas podem ser extraídas. As métricas estáticas são coletadas a partir de representações do sistema, modelo, programa ou documentação, sem exigir a execução do mesmo, enquanto as

dinâmicas são coletadas através da medição feita com o software em execução. As métricas estão associadas aos atributos de qualidade, de forma que estas medições permitam uma análise quantitativa do software. A escolha das métricas a serem empregadas em uma análise deve basear-se nas características mais relevantes para o produto.

Muitas das métricas são genéricas e podem ser empregadas para avaliar qualquer tipo de software. No entanto, devido as especificidades do software orientado a objetos (OO), métricas específicas foram definidas para avaliá-lo, as quais consideram herança, polimorfismo, dentre outras características. Existem diversas métricas OO, bem como limiares de referência que ajudam na análise dos resultados [Ferreira et al. 2011], [Eclipse 2014], [Chandra e Linda 2010]. Para este trabalho foram escolhidas as métricas DIT (Profundidade da Árvore de Herança) e Instabilidade para estimar a subcaracterística Reusabilidade. Já as métricas WMC (Métodos Ponderados por Classe), LCOM (Falta de Coesão entre Métodos) e NOC (Número de Filhos) foram usadas para a subcaracterística Modularidade. Para a Modificabilidade foram utilizadas as métricas NOM (Número de Métodos) e Distância da Sequência Principal (DMS).

O valor de referência típico para a métrica DIT é 2, ou seja, o ideal é que a profundidade da árvore de herança tenha pelo menos dois níveis [Ferreira et al. 2011]. Já o limiar para a métrica Instabilidade é entre 0 e 1 [Eclipse 2014]. Quanto mais próximo de 0 for o valor encontrado, mais estáveis estão os pacotes, indicando que estes são menos vulneráveis a mudanças. O limiar desejado para WMC deve ser maior que 0 e menor que 15, segundo Chandra e Linda (2010). Já para a métrica LCOM, o limiar de referência está entre 0 e 1 [Eclipse 2014]. Se o valor extraído está próximo de 0, significa que os métodos estão coesos e valores próximos de 1 indicam falta de coesão entre os métodos. Para a métrica NOC, o limiar esperado encontra-se entre 0 e 6, ou seja, espera-se que uma classe tenha no máximo 6 filhos [Chandra e Linda 2010]. Para a métrica NOM, o limiar de referência diz que para códigos em Java o número de métodos em uma classe deve ser no máximo 20 [Herbold, Grabowski e Waack 2011]. Para a métrica Distância da Sequência Principal o limiar está entre 0 e 1. Valores próximos de 1 para esta métrica, indicam que o código deve ser refatorado [Eclipse 2014].

### **3. Trabalhos Relacionados**

Nos últimos anos, vários autores propuseram modelos para ESS. O GREENSOFT foi proposto em [Naumann et al. 2011] como um modelo para desenvolvimento sustentável, o qual inclui a sustentabilidade em todos seus níveis (desenvolvimento, distribuição, uso e gerenciamento). No entanto, este trabalho não inclui a sustentabilidade nas fases de análise de requisitos e de testes, que também tem impacto na sustentabilidade do produto. Mais recentemente, um modelo baseado em dois níveis foi proposto para avaliação de sustentabilidade em [Mahmoud e Ahmad 2013]. Neste modelo, no primeiro nível, a sustentabilidade é considerada em uma fase de análise verde, incluída em um modelo de processo de software baseado em métodos sequenciais e iterativos. No segundo nível, são discutidas características que devem estar presentes no próprio software para que este esteja de acordo com os princípios da TI verde. Estes dois modelos incluem a sustentabilidade no processo, porém, possuir um processo de desenvolvimento sustentável não garante que o produto também seja sustentável.

Preocupados com a sustentabilidade junto ao produto de software, três propostas foram apresentadas [Calero, Moraga e Bertoa 2013][Naumann et al. 2013][Albertao et al. 2011]. Em [Calero, Moraga e Bertoa 2013], foi incluída a sustentabilidade como uma característica do modelo de qualidade de software, com as seguintes subcaracterísticas: consumo de energia, otimização de recursos e durabilidade. Nesta proposta, os três pilares da sustentabilidade não são considerados, e consumo de energia é duplicado já que este já é considerado como uma subcaracterística da eficiência pela norma ISO. Em [Naumann et al. 2013], foi apresentado um modelo de qualidade, que além de critérios comuns de qualidade, considera critérios diretamente e indiretamente relacionados à sustentabilidade. Apesar deste modelo considerar a sustentabilidade, o único pilar considerado é o social, não descrevendo o ambiental e nem o econômico. Os autores não respeitam a estrutura hierárquica da norma de qualidade da ISO/IEC, desta forma, subcaracterísticas de uma mesma característica estão associadas a critérios diferentes, distanciando a avaliação da sustentabilidade, da avaliação de qualidade do software.

Uma avaliação quantitativa para a sustentabilidade de software é discutida em [Albertao et al. 2011], onde é proposto um conjunto de Métricas de Desempenho em Sustentabilidade para prover os aspectos econômicos, sociais e ambientais de um produto. Esta avaliação baseou-se nas características de qualidade da norma ISO/IEC 9126 e definiu um conjunto de métricas para cada atributo de qualidade. Porém, a avaliação proposta é conduzida no nível de características, além dos autores discutirem a questão energética de forma muito abstrata e teórica. Neste artigo, esta análise é estendida de forma a considerar a norma atual, a ISO/IEC 25010:2011, e analisar também os impactos na sustentabilidade no nível de subcaracterísticas. Além disso, diferentemente de [Albertao et al. 2011], este trabalho considera métricas OO e propõe uma abordagem para avaliação da sustentabilidade com foco em aplicativos móveis.

#### **4. Abordagem Proposta**

Numa visão geral, a abordagem de avaliação proposta baseia-se na integração da sustentabilidade como um atributo transversal do modelo de qualidade do produto de software, o qual será chamado de fator de sustentabilidade, e deve ser relacionado a todas características e subcaracterísticas do modelo. Este fator permite discutir impactos de uma dada subcaracterística nos três pilares da sustentabilidade (econômico, social e ambiental). Além disso, a abordagem combina esta análise de impactos com uma avaliação quantitativa do software através do emprego de métricas.

O processo proposto de avaliação de sustentabilidade é ilustrado na Figura 1. Este processo é composto pelos passos: “Escolha de características”, “Escolha de métricas”, “Extração de métricas” e “Análise dos Resultados”. O primeiro passo consiste em selecionar as características de qualidade que são mais relevantes para o software. O segundo passo é extrair as métricas estáticas e dinâmicas para verificar se as características estão contempladas pelo software. Por fim, com base nos resultados e na análise de impactos, a sustentabilidade é avaliada.

De forma geral, a abordagem pode ser aplicada a diferentes tipos de softwares. No entanto, neste artigo, a avaliação de sustentabilidade foi empregada para o domínio de aplicativos móveis. Desta forma, este estudo considera as características portabilidade, manutenibilidade e eficiência, restringindo assim a avaliação a um subconjunto da ISO/IEC 25010. Considerando que os dispositivos móveis representam

um importante mercado que gera um considerável volume de lixo eletrônico, ainda maior que o dos desktops, a portabilidade é desejada pois pode minimizar este impacto negativo. A manutenibilidade foi escolhida por permitir uma maior vida útil do produto e é analisada por três de suas subcaracterísticas: reusabilidade, modificabilidade e modularidade. A eficiência e portabilidade também são consideradas características importantes para este domínio, sobretudo devido às restrições energéticas e ao grande número de plataformas e aparelhos distintos no mercado. A eficiência é avaliada através de duas subcaracterísticas, desempenho e energia (uso da bateria). A portabilidade foi avaliada através de um método empírico, analisando as subcaracterísticas adaptabilidade e conformidade. A conformidade é discutida pela adoção de APIs e pela garantia de suporte a APIs Android já existentes. Já a adaptabilidade é discutida pela dificuldade de portar um aplicativo Android para outras plataformas móveis.

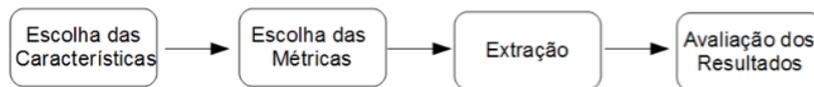


Figura 1: Processo de Avaliação Proposto

#### 4.1. Análise de Impactos

A Manutenibilidade, assim como já discutido em [Albertao et al. 2014], afeta os três pilares da sustentabilidade. Do ponto de vista econômico, um software manutenível, facilita a manutenção e permite que novos *releases* sejam disponibilizados mais rapidamente aos usuários, o que corresponde a um impacto positivo no âmbito social. No ponto de vista ambiental, melhorias na manutenibilidade podem diminuir os gastos energéticos, além de reduzir a necessidade de impressão de documentação durante a manutenção.

Considerando as subcaracterísticas da manutenibilidade, esta análise pode ser detalhada. A subcaracterística reusabilidade de software reduz o esforço de desenvolvimento, além de acelerar o tempo de lançamento no mercado, o que tem impacto econômico, social e ambiental. A modificabilidade facilita a compreensão e modificação do sistema. Já a modularidade aumenta a reusabilidade e em conjunto com a modificabilidade reduz o esforço dos desenvolvedores para modificar o sistema e acelera a manutenção. Melhorias nestas subcaracterísticas diminuem gastos com manutenção e reengenharia (aspectos econômico e ambiental), portando impactando nos três pilares.

No que se refere a eficiência temporal, melhorias podem evitar gastos com otimizações ou ainda investimentos para a aquisição de máquinas com melhor configuração (aspecto econômico). Além disso, melhorias na eficiência temporal permitem a redução do tempo de execução, reduzindo o tempo de espera do usuário pela resposta, podendo acelerar as operações apoiadas pelo software. Por exemplo, em se tratando de um sistema de atendimento ao público, a eficiência pode afetar ainda um maior número de pessoas e não apenas seus usuários diretos (aspecto social). Ainda, considerando o aspecto ambiental, a eficiência temporal evita aquisição de novos

equipamentos, reduzindo o lixo eletrônico gerado pela substituição de equipamentos, além de evitar gastos energéticos associados com implementação e teste de otimizações.

A portabilidade, no que refere ao aspecto econômico, permite o aumento no potencial de mercado e tempo de vida do sistema, evitando que novas versões tenham de ser desenvolvidas para rodar em diferentes ambientes. A portabilidade reduz ainda o custo de emprego de aplicativos, podendo portá-los para diferentes dispositivos (aspecto social). No aspecto ambiental, a portabilidade também minimiza o lixo eletrônico, uma vez que estende o tempo de vida do hardware [Albertao et al. 2011].

#### 4.2. Avaliação Quantitativa de Aplicativos Móveis

Para realizar a avaliação quantitativa das características dos aplicativos móveis, são empregadas métricas estáticas OO, listadas na Tabela 1, além de métricas dinâmicas para a avaliação da eficiência da aplicação. Uma vez que não há estudos que indiquem que as métricas adotadas para avaliação de software tradicional, não possam ser usadas para avaliar *mobile* e nem foram encontradas propostas de métricas específicas para este domínio, nossa abordagem emprega as métricas de software mais conhecidas e para as quais se tem valores de referência. Assim, na abordagem proposta, a reusabilidade é avaliada através das métricas DIT e Instabilidade. Já a modularidade é analisada através de WMC, LCOM e NOC e a modificabilidade pelo NOM e DMS. A eficiência é analisada através do tempo de execução e da energia consumida, que são métricas extraídas após a execução da aplicação. Já a portabilidade será discutida, considerando características da plataforma computacional empregada.

Conforme a abordagem proposta, após a extração das métricas OO, os resultados obtidos são analisados, comparando-os com os valores de referência encontrados na literatura. Assim, se determina se o valor obtido para uma dada métrica é apropriado e pode-se concluir se a característica avaliada está contemplada pela produto ou não. Esta análise quantitativa das características permite analisar os decorrentes efeitos positivos e negativos no fator sustentabilidade associado às características.

**Tabela 1. Métricas Utilizadas e Características Relacionadas.**

Características	Métricas	Descrição
Reusabilidade	DIT	Profundidade da Árvore de Herança
	Instabilidade	Instabilidade de Pacotes
Modularidade	WMC	Métodos Ponderados por Classe
	LCOM	Falta de Coesão entre Métodos
	NOC	Número de Filhos
Modificabilidade	NOM	Número de Métodos
	DMS	Distância da Sequência Principal
Portabilidade	Plataforma Alvo	Plataformas Possíveis de Execução
	Recursos de Hardware	Hardware Necessário para Executa a Aplicação
Eficiência	Incl CPU Time	Tempo de Execução
	Consumo Energético	Consumo de Energia Durante a Execução

Conforme a abordagem proposta, após a extração das métricas OO, os resultados obtidos são analisados, comparando-os com os valores de referência encontrados na literatura. Assim, se determina se o valor obtido para uma dada métrica é apropriado e pode-se concluir se a característica avaliada está contemplada pela produto ou não. Esta análise quantitativa das características permite analisar os decorrentes efeitos positivos e negativos no fator sustentabilidade associado às características.

Para a análise dos resultados das métricas dinâmicas, não há valores de referência para comparação. Assim, ao se obter um valor para tempo de execução, por exemplo, necessita-se também comparar este valor com alguma meta. Esta meta pode ser determinada por um requisito não funcional da aplicação ou pode ser definida pela comparação com outra aplicação disponível, com a mesma funcionalidade e com desempenho adequado. Nestas comparações, o mesmo cenário de uso deve ser considerado e um número significativo de execuções deve ser analisado.

## 5. Estudo de Caso

Nesta seção, a avaliação de sustentabilidade proposta é demonstrada através de dois estudos de caso com aplicativos móveis, com foco na plataforma Android. Cada estudo de caso consiste na avaliação e comparação de dois aplicativos. Em um dos estudos de caso, dois players de música são avaliados e no segundo, dois aplicativos de exploração de sistema de arquivos são analisados. A escolha destas aplicações se deve ao fato de serem cotidianas aos usuários de um dispositivo móvel e seus códigos fontes estarem disponíveis nos repositórios GitHub (2014) e GoogleCode (2014a).

### 5.1 Metodologia

Para a extração das métricas estáticas foi usado o plugin Metrics disponível para a IDE Eclipse, que provê medidas para as métricas abordadas neste trabalho, bem como média e desvio padrão das mesmas. A avaliação das métricas estáticas é realizada através da comparação com limiares que definem faixa de valores aceitáveis para cada métrica.

Já a característica eficiência é avaliada considerando as métricas dinâmicas: tempo de execução e consumo energético. Como estas medidas variam a cada execução dos aplicativos, foram realizadas trinta execuções de cada experimento para validade estatística dos resultados. Após as execuções, é calculada a média e desvio padrão dos valores obtidos. Após os cálculos, para as duas aplicações, estes dois resultados devem ser comparados com um teste estatístico, o qual indicará se existe diferença significativa na eficiência das duas ou se ambas tem a mesma eficiência.

A avaliação de desempenho de aplicativos Android é realizada usando o DDMS (*Dalvik Debugger Managment System*) [Google 2014b] e a biblioteca android.os.Debug que suportam o rastreamento da execução de uma aplicação Android. Os dados do rastreamento são visualizados na ferramenta Traceview [Google 2014c], que provê o tempo de CPU utilizado para a execução do aplicativo.

Para avaliar consumo energético é usado o aplicativo PowerTutor [Zhang et al. 2010], que estima consumo energético de uma aplicação Android que está executando no dispositivo físico. Em nossos experimentos, para todas as execuções, primeiramente, o aplicativo PowerTutor é ativado e após a aplicação em análise é executada.

Para a avaliação da eficiência (temporal e energética), também foi necessário definir os cenários de uso que seriam avaliados nos dois estudos de caso. Para as aplicações de exploração do sistema de arquivos (File Explorer), realizou-se a busca de uma foto no sistema de arquivos do dispositivo, navegando na hierarquia de pastas. O mesmo arquivo foi buscado nos dois aplicativos, seguindo o mesmo roteiro para fins de comparação de eficiência. Para a avaliação dos players, uma música em formato MP3 de 3:01 minutos foi executada. A análise da execução inicia quando a aplicação é inicializada e termina quando a execução da música é concluída e então o botão voltar é pressionado. O mesmo cenário é considerado na análise dos dois players.

## 5.2. Avaliação de aplicativos ‘Exploradores de Arquivos’

Os resultados obtidos para as métricas estáticas retiradas dos aplicativos File-Explorer e Man-File-Explorer estão resumidos na Tabela 2. Nesta tabela são apresentadas as características que estão sendo avaliadas, as métricas utilizadas, e na sequência, apresenta-se o valor da média, desvio padrão e resultado concluído a partir da comparação com os limiares, para cada um dos aplicativos.

Com os resultados obtidos pode-se observar que para todas as medidas do primeiro aplicativo, referenciado como File Explorer, os resultados obtidos encontram-se dentro do limiar definido, isto significa que as características modificabilidade, reusabilidade e modularidade estão sendo contempladas. Para o aplicativo Man-File-Explorer, os valores também são apropriados, exceto para a métrica Instabilidade. Para esta métrica um valor muito próximo de 1 foi obtido, indicando que há instabilidade entre os pacotes do aplicativo. Então, analisando os resultados observa-se que as características modularidade e modificabilidade são suportadas, como indicado pelos valores de WMC, LCOM e NOC, e pelos valores de NOM e DMS, respectivamente. No entanto, apesar da métrica DIT estar dentro do limiar, a característica Reusabilidade apresentou resultados insatisfatórios, pois a reutilização de pacotes instáveis torna-se difícil e inconveniente.

**Tabela 2. Resultados das Métricas Estáticas para os Exploradores de Arquivos**

Característica	Métrica	File-Explorer			Man-File-Explorer		
		Média	$\sigma$	Resultado	Média	$\sigma$	Resultado
Reusabilidade	DIT	1,974	1,6	Típico	1,761	1,324	Típico
	Instabilidade	0,24	0,218	Estável	0,778	0,314	Instável
Modularidade	WMC	8,733	16,775	Bom	7,571	11,371	Bom
	LCOM	0,163	0,266	Coeso	0,157	0,29	Bom
	NOC	0,067	0,327	Bom	0,238	0,526	Bom
Modificabilidade	NOM	3,133	5,451	Bom	3,571	5,5	Bom
	DMS	0,681	0,279	Satisfatório	0,111	0,079	Estável

Os resultados obtidos para a característica eficiência são apresentados na Tabela 3. A partir da realização do teste estatístico "t de Student", verificou-se que as médias do

consumo energético dos aplicativos são iguais, mas os desempenhos dos mesmos se diferem. O aplicativo Man-File-Explorer possui melhor desempenho em comparação ao File-Explorer, o que impacta positivamente na sustentabilidade.

**Tabela 3. Resultados das Métricas Dinâmicas para os Exploradores de Arquivos**

Resultado	File-Explorer		Man-File-Explorer	
	Consumo (J)	Desempenho (ms)	Consumo (J)	Desempenho (ms)
Média	5,2466	5262,7245	4,93	3825,7158
$\sigma$	0,8127	308,1808	0,7799	631,6846

### 5.3. Avaliação de Aplicativos Players

Os valores obtidos para as métricas estáticas dos aplicativos Paranerd-Player e Vanilla-Player, bem como a análise dos mesmos estão sumarizados na Tabela 4. Neste estudo de caso, pode-se observar que para a maioria das métricas analisadas para o aplicativo Paranerd-Player, os resultados obtidos encontraram-se dentro do limiar definido ou dos valores de referência. Vale ressaltar apenas a métrica DMS, que apresentou um valor intermediário entre 0 e 1, indicando que encontra-se em um estado satisfatório, não estando totalmente estável nem instável. Analisando os resultados obtidos através das métricas estáticas, pode ser concluído que todas as características analisadas (modificabilidade, modularidade e reusabilidade) estão contempladas neste aplicativo, embora em diferentes graus.

Para as métricas estáticas extraídas do aplicativo Vanilla-Player, observa-se que para a maioria das métricas analisadas os resultados obtidos encontram-se dentro do limiar. Apenas a métrica WMC encontra-se acima do limiar desejado, onde o resultado obtido é de 22,102 e a referência indica um máximo de 15, indicando que os métodos deste aplicativo devem ser refatorados para que a característica modularidade seja melhorada. As subcaracterísticas modificabilidade e reusabilidade estão suportadas neste aplicativo.

**Tabela 4. Resultados para as Métricas Estáticas dos Players**

Característica	Métrica	File-Explorer			Man-File-Explorer		
		Média	$\sigma$	Resultado	Média	$\sigma$	Resultado
Reusabilidade	DIT	2,325	1,539	Típico	2,67	1,095	Típico
	Instabilidade	0,447	0,264	Satisfatório	0,496	0,309	Satisfatório
Modularidade	WMC	7,55	12,06	Bom	22,102	53,352	Ruim
	LCOM	0,315	0,383	Coeso	0,207	0,352	Coeso
	NOC	0	0	Bom	0,057	0,349	Bom
Modificabilidade	NOM	3,05	3,398	Bom	6,352	11,76	Bom
	DMS	0,553	0,624	Satisfatório	0,43	0,343	Estável

Os resultados para eficiência dos Players estão sumarizados na Tabela 5. Realizando o teste estatístico com as médias para consumo energético, verificou-se que estas são diferentes e o aplicativo Paranerd-Player possui menor consumo energético que o Vanilla-Player. No entanto, para os resultados de desempenho, verificou-se através do teste estatístico que o aplicativo Vanilla-Player possui melhor desempenho.

**Tabela 5. Resultados para Métricas Dinâmicas dos Players**

Resultado	File-Explorer		Man-File-Explorer	
	Consumo (J)	Desempenho (ms)	Consumo (J)	Desempenho (ms)
Média	7,4066	7214,9848	9,7566	2194,3354
$\sigma$	1,361	572,182	1,9333	302,9029

#### 5.4. Análise dos Resultados

Comparando as duas aplicações do tipo exploradores de arquivos, observa-se que o aplicativo Man-File-Explorer possui melhor desempenho que o aplicativo File-Explorer. Porém, para consumo energético as médias obtidas podem ser consideradas estatisticamente iguais, ou seja, ambos aplicativos possuem consumos equivalentes. Analisando as métricas estáticas de ambos, nota-se que os resultados são satisfatórios para os dois aplicativos, apenas o valor da métrica Instabilidade encontra-se fora do limiar desejado no aplicativo Man-File-Explorer.

Realizando a comparação entre os aplicativos de players, observa-se que o aplicativo Vanilla-Player possui melhor desempenho se comparado ao Paranerd-Player. No entanto, o aplicativo Paranerd-Player possui um melhor consumo energético em relação ao Vanilla-Player. Isso reforça a importância de avaliar subcaracterísticas e não apenas a característica eficiência, de forma genérica. Observando as métricas estáticas, ambos possuem resultados satisfatórios, apenas na métrica WMC o aplicativo Vanilla-Player ultrapassou o limiar, o que não acontece no aplicativo Paranerd-Player.

A característica portabilidade foi deixada para ser analisada por fim, pois todos os aplicativos são desenvolvidos para a plataforma Android, que é uma plataforma aberta, desenvolvida pela Google e presente em diversos dispositivos. Todos os aplicativos são desenvolvidos para a API 4.0.3 do Android ou superior. Apesar de não funcionar corretamente em plataformas anteriores, atualmente todos dispositivos vendidos já disponibilizam a API 4.0 ou superior e muitas vezes permitem a atualização do sistema, toda vez que uma nova versão é lançada. Assim, o aplicativo torna-se portátil para qualquer dispositivo Android atual, independente de suas características de hardware. Extrapolando esta análise para outras plataformas móveis, vale destacar que cada plataforma utiliza linguagens e APIs de desenvolvimento diferentes, o que usualmente melhora a eficiência dos aplicativos, mas em contrapartida, prejudica o desenvolvimento para múltiplas plataformas computacionais e conseqüentemente tem impacto negativo na portabilidade dos aplicativos. Apesar dos aplicativos Android serem desenvolvidos em Java, estes utilizam recursos particulares do Android, o que dificulta a portabilidade para outras plataformas.

As subcaracterísticas modificabilidade, modularidade e reusabilidade, como discutido na Seção 4.1, tem impacto positivo na sustentabilidade. Observando a análise destas subcaracterísticas para o aplicativo File-Explorer, conclui-se que a aplicação

contempla estas subcaracterísticas e, portanto, a aplicação incorpora os benefícios para sustentabilidade relacionados a estas características. Para a aplicação Man-File-Explorer apenas o fator sustentabilidade das subcaracterísticas modificabilidade e modularidade podem ser consideradas, uma vez que a aplicação possui reusabilidade limitada. A reusabilidade limitada pode trazer impactos negativos à sustentabilidade do aplicativo, como aumentar o custo de novos desenvolvimentos. No entanto, é consenso que para se ter aplicativos mais eficientes, projetistas abrem mão de soluções reusáveis e utilizam soluções específicas ou otimizadas.

Para as aplicações do tipo player, as características avaliadas para a aplicação Paranerd-Player mostram-se todas satisfatórias, uma vez que todas as métricas analisadas estão dentro do limiar esperado. Já para o aplicativo Vanilla-Player, apenas as métricas que avaliam modificabilidade não apresentam valores esperados, não sendo possível associar seu fator de sustentabilidade ou seus benefícios à sustentabilidade deste aplicativo. A modificabilidade é uma importante subcaracterística da manutenibilidade e uma aplicação com uma baixa manutenibilidade, requer gastos maiores na fase de manutenção, além de prejudicar o entendimento dos programadores que estão realizando esta tarefa. Além disso, ao tornar a fase de manutenção mais longa, consequentemente há maiores gastos com recursos energéticos para que esta se complete. Desta forma, pode-se concluir que o aplicativo Vanilla-Player deve ser refatorado para que os benefícios da modificabilidade possam ser considerados neste aplicativo e impactando positivamente em sua sustentabilidade.

## **6. Conclusões**

Este trabalho propõe uma avaliação da sustentabilidade com foco em aplicativos móveis, a qual baseia-se na análise quantitativa de um conjunto de características de qualidade definidas na norma internacional da ISO/IEC e de interesse para este domínio de aplicação. Neste conjunto, encontram-se as características manutenibilidade, portabilidade e eficiência. A abordagem inclui o emprego de métricas estáticas e dinâmicas para avaliação destas propriedades e de suas subcaracterísticas. A sustentabilidade é considerada como um atributo transversal do modelo atual de qualidade, desde o nível de subcaracterísticas como modificabilidade, modularidade e consumo energético. Neste trabalho são discutidos também de que forma cada uma destas subcaracterísticas de interesse para o domínio de aplicativos móveis impacta os pilares econômico, social e ambiental.

Para demonstrar a abordagem proposta, estudos de caso de avaliação de aplicativos Android foram realizados. Os softwares avaliados foram players de MP3 e exploradores de sistemas de arquivos, que representam aplicações frequentemente usadas em dispositivos móveis. Estes estudos de caso demonstraram que é possível avaliar e discutir a sustentabilidade de aplicativos com base em suas características de qualidade, assim como reforçaram que a análise quantitativa deve incluir subcaracterísticas e não se ater ao nível das características da norma de qualidade. Além disso, não deve se restringir a avaliação de sustentabilidade unicamente a avaliação de eficiência energética, visto que a sustentabilidade é influenciada por outras características de software. Considerando que uma análise de sustentabilidade não pode avaliar métricas isoladamente, este trabalho discute apenas o impacto positivo ou negativo de cada métrica na sustentabilidade, mas não define uma equação para calcular o grau de sustentabilidade de um software.

Este trabalho avaliou apenas aplicativos móveis convencionais. Em aplicações críticas, as características segurança e confiabilidade também devem ser incluídas, por serem de extrema importância neste domínio. Como trabalhos futuros, pretende-se estender a avaliação proposta para incluir outras características relevantes para outros tipos de softwares, ou ainda domínios de aplicação distintos.

### **Agradecimentos**

Os autores agradecem ao CNPQ pelo apoio financeiro (Universal 14/2013 - processo n. 483464/2013-9).

### **Referências**

- Albertao, F., Xiao, J., Lu, C., Zhang, K. Q., Liu, C. (2011) “Measuring the Sustainability Performance of Software Projects”, Em: Sustainable Computing: Informatics and Systems, p. 294-302.
- Arndt, H., Dziubaczyk, B., Motosch, M. (2014) “Impact of Design on the Sustainability of Mobile Applications”, Em: Information Technology in Environmental Engineering, p. 13-24.
- Calero, C., Moraga, M. A., Bertoa, M. F. (2013) “Towards a Software Product Sustainability Model”, Em: Workshop on Sustainable Software for Science: Practice and Experiences, p. 1-4.
- Chandra, D. E., Linda, P. E. (2010) “Class Break Point Determination Using CK Metrics Thresholds”, Em: Global Journal of Computer Science and Technology, 65, p.73-77.
- Eclipse (2014) “Eclipse Metrics Plugin”, <http://metrics.sourceforge.net/>.
- Ferreira, K. A. M., Bigonha, M. A. S., Bigonha, R. S., Mendes, L. F. O., Almeida, H. C. (2011) “Identifying thresholds for object-oriented software metrics”, Em: Journal of Systems and Software, p. 244-257.
- Gitlab (2014) “Repositório GitHub”, <https://github.com/>.
- Google (2014a) “Google Code”, <https://code.google.com/>.
- Google (2014b) “DDMS”, <http://developer.android.com/tools/debugging/ddms.html>.
- Google (2014c) “Traceview”, <http://developer.android.com/tools/help/traceview.html>.
- Herbold, S., Grabowski, J., Waack, S. (2011) “Calculation and optimization of thresholds for sets of software metrics”, Em: Empirical Software Engineering, 16, p. 812-841.
- ISO (2014) “International Organization for Standardization”, <http://www.iso.org/iso/home.html>.
- ISO (2011) “International Organization for Standardization ISO/IEC 25010:2011”, <https://www.iso.org/obp/ui#iso:std:iso-iec:25010:ed-1:v1:en>.
- Naumann, S., Johann, T., Dick, M., Kern, E. (2011) “The GREENSOFT Model: A reference model for green and sustainable software and its engineering”, Em: Sustainable Computing: Informatics and Systems, 1, 294-304.

- Mahmoud, S. S., Ahmad, I. (2013) A Green Model for Sustainable Software Engineering, Em: International Journal of Software Engineering and Its Applications, p.55-74.
- Naumann, S., Kern, E.; Dick, M.; Johann, (2013) “T. Sustainable Software Engineering: Process and Quality Models, Life Cycle, and Social Aspects”, Em: ICT Innovations for Sustainability: Advances In Intelligent Systems and Computing, Springer International Publishing.
- Tonini, A. R., Fischer, L. M., Mattos, J. C. B., Brisolara, L. B. (2013) “Analysis and Evaluation of the Android Best Practices Impact on the Efficiency of Mobile Applications”, Em: III Brazilian Symposium on Computing Systems Engineering, p. 157-158.
- WCED. (1987), Our Common Future, Oxford University Press.
- Zhang, L.; Tiwana, B.; Qian, Z.; Wang, Z.; Dick, R. P.; Mao, M.; Yang, L. (2010) “Accurate Online Power Estimation and Automatic Battery Behavior Based Power Model Generation for Smartphones“, Em: International Conference on Hardware/Software Codesign and System Synthesis, p.105-114.