



Pós-Graduação em Ciência da Computação

**“ gDFS: um mecanismo de otimização para  
redução de consumo de energia em  
smartphones através de uma estratégia baseada  
em grupos de aplicações”**

**Por**

***Antônio Correia de Sá Barreto Neto***

**Dissertação de Mestrado**



Universidade Federal de Pernambuco  
posgraduacao@cin.ufpe.br  
www.cin.ufpe.br/~posgraduacao

RECIFE, 2014





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

Antônio Correia de Sá Barreto Neto

**“gDFS: Um mecanismo de otimização para redução de consumo de energia em smartphones através de uma estratégia baseada em grupos de aplicações”**

*Trabalho apresentado ao Programa de Pós-graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco como requisito parcial para obtenção do grau de Mestre em Ciência da Computação.*

Orientador: Abel Guilhermino da Silva Filho

RECIFE, 2014

Catálogo na fonte  
Bibliotecária Monick Raquel Silvestre da Silva, CRB4-1217

B273g Barreto Neto, Antônio Correia de Sá.

gDFS: um mecanismo de otimização para redução de consumo de energia em smartphones através de uma estratégia baseada em grupos de aplicações / Antônio Correia de Sá Barreto Neto. – Recife: O Autor, 2014.

116 f.: il., fig., tab.

Orientador: Abel Guilhermino da Silva Filho.

Dissertação (Mestrado) – Universidade Federal de Pernambuco. CIN. Ciência da Computação, 2014.

Inclui referências.

1. Engenharia da computação. 2. Sistemas embarcados. 3. Low power. I. Silva Filho, Abel Guilhermino (orientador). I. Título.

621.39

CDD (22. ed.)

UFPE-MEI 2014-143

Dissertação de Mestrado apresentada por **Antônio Correia Sá Barreto Neto** à Pós-Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, sob o título “**gDFS: Um mecanismo de otimização para redução de consumo de energia em smartphones através de uma estratégia baseada em grupos de aplicações**” orientada pelo **Prof. Abel Guilhermino da Silva Filho** e aprovada pela Banca Examinadora formada pelos professores:

---

Profa. Patricia Cabral de Azevedo Restelli Tedesco  
Centro de Informática / UFPE

---

Prof. Remy Eskinazi Sant'Anna  
Departamento de Engenharia Elétrica/ UPE

---

Prof. Abel Guilhermino da Silva Filho  
Centro de Informática / UFPE

Visto e permitida a impressão.  
Recife, 12 de agosto de 2014.

---

**Profa. Edna Natividade da Silva Barros**

Coordenadora da Pós-Graduação em Ciência da Computação do  
Centro de Informática da Universidade Federal de Pernambuco.

*Eu dedico este trabalho a todos os meus amigos, meus pais e meu irmão, sem a ajuda dos quais não poderia ter chegado até aqui.*

# Agradecimentos

Eu gostaria de começar agradecendo a meus pais que sempre acreditaram no meu potencial e investiram nele, apoiando-me em todas as etapas de minha vida, de tal forma que hoje foi possível chegar até aqui.

Agradeço também ao Professor Doutor Abel Guilhermino da Silva Filho que, em 2011, em uma conversa em sua sala, acreditou que meus conhecimentos prévios me possibilitariam construir pesquisas relevantes e aceitou o desafio de conduzir a realização deste trabalho.

Meus agradecimentos a Marco Aurélio Tomaz Mialaret Júnior, pelo total apoio no desenvolvimento das teorias matemáticas necessárias durante a realização deste trabalho, a fim de conduzir e ensinar-me os fundamentos matemáticos necessários para que eu obtivesse êxito nos meus resultados, além de auxiliar-me nas análises estatísticas dos dados, agindo como um verdadeiro crítico até a conclusão do mesmo.

Agradeço ao professor Doutor Antônio Correia de Sá Barreto Filho, por também contribuir de forma significativa na elaboração das teorias matemáticas usadas durante a realização deste trabalho.

Agradeço também a Bruno Cartaxo que, desde os tempos de faculdade, sempre trabalhou em parceria comigo, apoiando-me nos meus desafios profissionais, e incentivou-me a prosseguir com meus estudos e realizar a seleção de mestrado.

Agradeço a Edwin Marinho, pela preocupação e disponibilidade em ajudar-me no final deste trabalho, por vezes até mesmo disponibilizando a sua casa para que eu pudesse realizar a escrita de minha dissertação.

Agradeço a Rafael Nunes do projeto de pesquisa da Samsung por apoiar-me e construir alguns circuitos que foram usados na etapa de coleta de dados deste trabalho.

Agradeço a Antônio Cavalcanti, Giovane Boaviagem, Pâmela Tays, Leandro Araújo, José Anderson, Gabriela Bulhões, Tatiana Bittecourt e a todos os outros que fazem parte do projeto Samsung, por dividirem comigo a árdua tarefa de realizar pesquisas a fim de prover aos dispositivos móveis mecanismos para a redução do consumo energético.

Agradeço também a Tamine Chaves Gomes que, nos momentos em que eu estava bastante desanimado com as tentativas mal-sucedidas de chegar aos resultados obtidos neste trabalho, apoiou-me e deu-me o ânimo necessário, acreditando no meu potencial.

Enfim, agradeço a todos os outros amigos que, mesmo de forma indireta, contribuíram, pois sei que sem eles a realização deste trabalho ficaria mais difícil. . . .



*"O engenheiro pensa o mundo justo,  
mundo que nenhum véu encobre."  
(O Engenheiro, João Cabral de Melo Neto)*



# Resumo

Houve, nos últimos anos, um aumento bastante significativo do uso de smartphones. No final de 2011, havia 6 bilhões de assinantes, de acordo com a união internacional de telecomunicações. Isso é equivalente a 87% da população mundial. O aumento nas vendas de smartphones em 2012 foi de 694,8 milhões, o que significa que, apenas nesse ano, houve um aumento de 11,58% na quantidade de assinantes que utilizam esse tipo de dispositivo. Android é o sistema operacional dominante para os novos smartphones vendidos em 2012, tendo 68,8% da fatia de mercado, com uma venda da ordem de 497,1 milhões de dispositivos. Assim, torna-se extremamente necessário aumentar o tempo de disponibilidade e aperfeiçoar as tecnologias envolvidas na fabricação desse tipo de dispositivo. Entretanto, dentre essas tecnologias, a de bateria foi a que menos evoluiu. Assim, faz-se necessário estudar mecanismos que possibilitem uma melhora no consumo energético dos smartphones, de modo a melhorar o tempo de disponibilidade dos mesmos. Este trabalho teve por objetivo, então, estudar mecanismos de otimização energética por meio de uma técnica de chaveamento de frequência, possibilitando o ajuste da mesma de uma forma mais inteligente que o modo feito por um Kernel Android padrão. Explorou-se o fato de as aplicações com funcionalidade semelhante possuírem uma faixa de frequências aceitáveis de operação, e ela foi configurada de tal maneira, que o smartphone se mantenha em uma frequência ótima durante todo o tempo de uso do aplicativo, chegando a níveis de economia da ordem de 22%.

**Palavras-chave:** Android. DVFS. CPUFreq. consumo de energia. chaveamento de frequência. otimização energética



# Abstract

The increasing usage of smartphones in the last years was very noticeable. At the end of 2011, there was 6 billion of subscribers according to International Union of Telecommunications. This is equivalent to 87% of the world population. There was an increase of 694,8 million of smartphones sales in 2012, so, this fact represent a growth of 11,58% in the number of subscribers that utilize this kind of device. Android is the main operation system in the new smartphones sold in 2012, representing 68,8% of the market share with a sale of approximately 487,1 million of devices. Based on these facts, it is extremely necessary an improvement on the availability time of these devices, because, analysing the evolution of technologies used in the manufacture of this kind of device, the battery technology has evolved much less than the others. So, is extremely necessary, studying mechanisms that enable an improvement in the energy consumption of smartphones such that the availability time of these devices become better. Based in these facts, this work has the main goal studying optimization mechanisms through a frequency scaling technique allowing the adjustment of it in a more intelligent way than those implemented in the standard Android Kernel. We used the fact of the applications with similar functionality have a range of acceptable frequencies, and we configured the device in a such way that it maintains the same frequency during its execution. Using this methodology, we got a power reduction of approximately 22%.

**Keywords:** Android. DFS. CPUFreq. energy consumption. frequency switching. energy optimization



# Lista de Figuras

Figura 1 – Evolução das tecnologias dos dispositivos presentes nos smartphones . . .	14
Figura 2 – Experimento com mudança de frequência . . . . .	17
Figura 3 – Experimento sem mudança de frequência . . . . .	17
Figura 4 – Arquitetura do CpuFreq . . . . .	38
Figura 5 – Ciclo de vida do serviço Android . . . . .	41
Figura 6 – Plano experimental desenvolvido para a realização da pesquisa . . . . .	45
Figura 7 – Circuito de Medição . . . . .	47
Figura 8 – Procedimento para coleta de dados . . . . .	48
Figura 9 – Comparação das iterações durante análise gráfica . . . . .	50
Figura 10 – Procedimento para realização da análise estatística . . . . .	51
Figura 11 – Procedimentos para realização da análise numérica . . . . .	55
Figura 12 – Interação entre os procedimentos de análise . . . . .	58
Figura 13 – Ambiente Experimental . . . . .	63
Figura 14 – Frequências ideal e aceitáveis . . . . .	68
Figura 15 – Comparação entre consumos da solução proposta e os governors do Kernel Android . . . . .	69
Figura 16 – Frequências ideal e aceitáveis x governors padrão do Android . . . . .	70
Figura 17 – Frequência ideal x governors padrão do Android . . . . .	71
Figura 18 – Frequência ideal x governors padrão do Android . . . . .	72
Figura 19 – Frequências ideal e aceitáveis para Youtube usando 3G . . . . .	73
Figura 20 – Frequência ideal x governors padrão do Android para Youtube usando 3G . . . . .	74
Figura 21 – Frequências ideal e aceitáveis para Dailymotion usando Wi-Fi . . . . .	75
Figura 22 – Frequência ideal x governors padrão do Android para Dailymotion usando Wi-Fi . . . . .	75
Figura 23 – Frequências ideal e aceitáveis para Dailymotion usando 3G . . . . .	76
Figura 24 – Frequência ideal x governors padrão do Android para Dailymotion usando 3G . . . . .	77
Figura 25 – Frequências ideal e aceitáveis para Vimeo usando Wi-Fi . . . . .	78
Figura 26 – Frequência ideal x governors padrão do Android para Vimeo usando Wi-Fi . . . . .	78
Figura 27 – Frequências ideal e aceitáveis para Vimeo usando 3G . . . . .	79
Figura 28 – Frequência ideal x governors padrão do Android para Vimeo usando Wi-Fi . . . . .	80

Figura 29 – Frequências ideal e aceitáveis para cálculo de rota usando Google Maps	83
Figura 30 – Frequência ideal x governors padrão do Android para cálculo de rota usando Google Maps . . . . .	84
Figura 31 – Frequências ideal e aceitáveis para cálculo de rota com iteração usando Google Maps . . . . .	85
Figura 32 – Frequência ideal x governors padrão do Android para cálculo de rota com iteração usando Google Maps . . . . .	86
Figura 33 – Frequências ideal e aceitáveis para cálculo de rota usando Waze . . . . .	87
Figura 34 – Frequência ideal x governors padrão do Android para cálculo de rota usando Waze . . . . .	88
Figura 35 – Frequências ideal e aceitáveis para cálculo de rota com iteração usando Waze . . . . .	89
Figura 36 – Frequência ideal x governors padrão do Android para cálculo de rota com iteração usando Waze . . . . .	90
Figura 37 – Frequências ideal e aceitáveis para exibição da página da Adobe . . . . .	93
Figura 38 – Frequência ideal x governors padrão do Android para exibição do site Adobe usando WiFi . . . . .	94
Figura 39 – Frequências ideal e aceitáveis para renderizar a página da Adobe usando 3G . . . . .	95
Figura 40 – Frequência ideal x governors padrão do Android para exibição do site Adobe usando 3G . . . . .	96
Figura 41 – Overview da solução final . . . . .	100
Figura 42 – Descrição da solução final . . . . .	101
Figura 43 – Comparação entre potências instantâneas com e sem mudança de frequência . . . . .	104
Figura 44 – Plano Experimental . . . . .	105

# Lista de Tabelas

Tabela 1 – Escala de Significância de Fisher. . . . .	23
Tabela 2 – Consumo das Tags HTML . . . . .	34
Tabela 3 – Layout do arquivo de coleta . . . . .	49
Tabela 4 – Layout do arquivo de coleta com potência . . . . .	49
Tabela 5 – Exemplo do Resultado do Teste Lilliefors para as séries de potência do Youtube . . . . .	52
Tabela 6 – Resultado do teste Lilliefors para as séries de potência do Youtube usando WiFi . . . . .	65
Tabela 7 – Consumo em Joules de cada uma das iterações . . . . .	66
Tabela 8 – Comparação entre as séries de potência instantânea produzidas pela execução do Youtube nas frequências disponíveis usando a rede de dados WiFi . . . . .	66
Tabela 9 – Tempo de execução do Google Maps apenas para cálculo de rota . . . . .	82
Tabela 10 – Comparação entre frequências com relação à potência instantânea do Google Maps para o cálculo de rota . . . . .	83
Tabela 11 – Tempo de execução do Google Maps para cálculo de rota e iteração . . . . .	84
Tabela 12 – Comparação entre frequências em relação à potência instantânea do Google Maps para cálculo de rota e iteração . . . . .	85
Tabela 13 – Tempo de execução do Waze para cálculo de rota . . . . .	87
Tabela 14 – Comparação entre frequências em relação à potência instantânea do Waze para o cálculo de rota . . . . .	87
Tabela 15 – Tempo de execução do Waze para cálculo de rota e iteração . . . . .	89
Tabela 16 – Comparação entre frequências em relação à potência instantânea do Waze para cálculo de rota e iteração . . . . .	89
Tabela 17 – Tempo de execução do Browser para exibir a página da Adobe . . . . .	92
Tabela 18 – Comparação entre frequências em relação à potência instantânea para exibição do site da Adobe na Wi-Fi . . . . .	93
Tabela 19 – Comparação entre frequências em relação à potência instantânea para exibição do site da Adobe na 3G . . . . .	95
Tabela 20 – Tempo de execução do Browser para exibir página do Baidu: Wi-Fi . . . . .	96
Tabela 21 – Tempo de execução do <i>Browser</i> para exibir página do Baidu: 3G . . . . .	97
Tabela 22 – Frequências ideais dos grupos de aplicações . . . . .	99



# Lista de Acrônimos

<b>CMOS</b>	Complementary Metal Oxide Semiconductor
<b>E/S</b>	Entrada e Saída
<b>USB</b>	Universal Serial Bus
<b>Tx</b>	Transmit Data
<b>Rx</b>	Receive Data
<b>CPUFreq</b>	CPu Frequency Scaling
<b>I2C</b>	Inter-Integrated Circuit
<b>3G</b>	Rede de dados móvel
<b>RRC</b>	Radio Resource Control
<b>HTML</b>	HyperText Markup Language
<b>DPM</b>	Dynamic Power Management
<b>MLE</b>	Maximum-Likelihood Estimation
<b>GSM</b>	Global System for Mobile Communications
<b>P2P</b>	Peer-to-Peer
<b>WCDMA</b>	Wide-Band Code-Division Multiple Access
<b>WLAN</b>	Wireless Local Area Network
<b>DVFS</b>	Dynamic Voltage Frequency Scaling
<b>GWh</b>	Giga Watts



# Sumário

<b>1</b>	<b>Introdução</b>	<b>13</b>
1.1	Objetivos	14
1.2	Motivação	15
<b>2</b>	<b>Referencial Teórico</b>	<b>19</b>
2.1	Estudos sobre Técnicas de Medição de Energia	19
2.2	Teste de Hipóteses	22
2.3	Valor-p	23
2.4	Teste de Aderência	23
2.5	Teste Lilliefors	25
2.6	Teste de Wilcoxon-Man-Whitney	25
2.7	Método de integração Numérica	26
2.8	Estudos sobre Otimização do Consumo	28
2.9	CPUFreq: Escalonamento de Frequência no <i>Kernel</i> Android	37
2.10	Serviços Android	39
<b>3</b>	<b>Metodologia</b>	<b>43</b>
3.1	Plano Experimental	45
3.1.1	Etapa 1: Caracterização da aplicação	46
3.1.2	Etapa 2: Análise gráfica para seleção das amostras a serem comparadas	49
3.1.3	Etapa 3: Análise estatística das amostras selecionadas	50
3.1.4	Etapa 4: Análise numérica das amostras selecionadas	54
3.2	Resumo das aplicações analisadas para um grupo de aplicações	56
3.3	Interação entre os procedimentos de análise	58
<b>4</b>	<b>Resultados</b>	<b>61</b>
4.1	Ambiente Experimental	62

4.2	Análise do aplicativo de <i>Stream</i> de Vídeo Youtube usando rede de dados Wi-Fi . . . . .	64
4.3	Análise de Aplicativos de <i>Stream</i> de Vídeo . . . . .	72
4.3.1	Análise Youtube . . . . .	73
4.3.2	Análise Dailymotion . . . . .	74
4.3.3	Análise Vimeo . . . . .	77
4.3.4	Conclusão das Análises de Aplicativos de <i>Stream</i> de Vídeo . . . . .	80
4.4	Análise de Aplicativos de Navegação GPS . . . . .	81
4.4.1	Análise Google Maps: Apenas Cálculo de Rota . . . . .	82
4.4.2	Análise Google Maps: Cálculo de Rota e Iteração . . . . .	84
4.4.3	Análise Waze: apenas cálculo de rota . . . . .	86
4.4.4	Análise Waze: cálculo de rota e iteração . . . . .	88
4.4.5	Conclusão das Análises de Aplicativos de Navegação GPS . . . . .	90
4.5	Análise de aplicativos de navegação de páginas . . . . .	91
4.5.1	Análise Adobe: Wi-Fi . . . . .	92
4.5.2	Análise Adobe: 3G . . . . .	94
4.5.3	Análise Baidu: Wi-Fi e 3G . . . . .	96
4.5.4	Conclusão das Análises para exibição de páginas no aplicativo Browser . . . . .	97
4.5.5	Conclusão do mecanismo proposto . . . . .	98
4.6	Arquitetura da Solução . . . . .	100
<b>5</b>	<b>Considerações Finais . . . . .</b>	<b>103</b>
	<b>Referências . . . . .</b>	<b>111</b>



# 1 Introdução

De acordo com o Instituto Gartner, a projeção para vendas de dispositivos (*PCs, tablets, ultramobiles e mobile phones*) está delineada para atingir 2,5 bilhões de unidades neste ano, um crescimento percentual de 7,6% em relação a 2013. Ainda segundo esse Instituto, no mercado de sistemas operacionais, o uso de Android ultrapassará 1 bilhão de usuários este ano e, em 2017, ele representará 75% dos dispositivos que usarão o sistema operacional da Google.

Ainda segundo relatório da Google (BESSA, 2012), 14% da população brasileira utilizam *smartphones* e, destes, 40% usam esses dispositivos todos os dias da semana. A importância desses dispositivos é tamanha, que 27% dos entrevistados preferem ficar sem televisão a não ter seus *smartphones*.

No entanto, de acordo com estudos do Green Car Congress (CONGRESS, 2010), as baterias de lítio responsáveis por alimentar esses dispositivos não evoluíram no mesmo passo que eles, pois, segundo estimativas, em 2020, o mercado demandará 195 *GWh* de energia, enquanto que, de acordo com o desenvolvimento das tecnologias de bateria atuais, só serão fornecidos 50 *GWh* de energia, gerando, assim, um *gap* de 145 *GWh* de energia.

Assim, outros estudos desenvolvidos por Paradiso e Starner (PARADISO; STARNER, 2005) já apontam o descompasso do crescimento entre as tecnologias dos componentes envolvidos na fabricação dos *smartphones* e a tecnologia envolvida na produção das baterias. De acordo com esses estudos, a velocidade do processador evoluiu na ordem de 100 vezes, enquanto as tecnologias de bateria evoluíram na ordem de 10 vezes, como mostra a Figura 1.

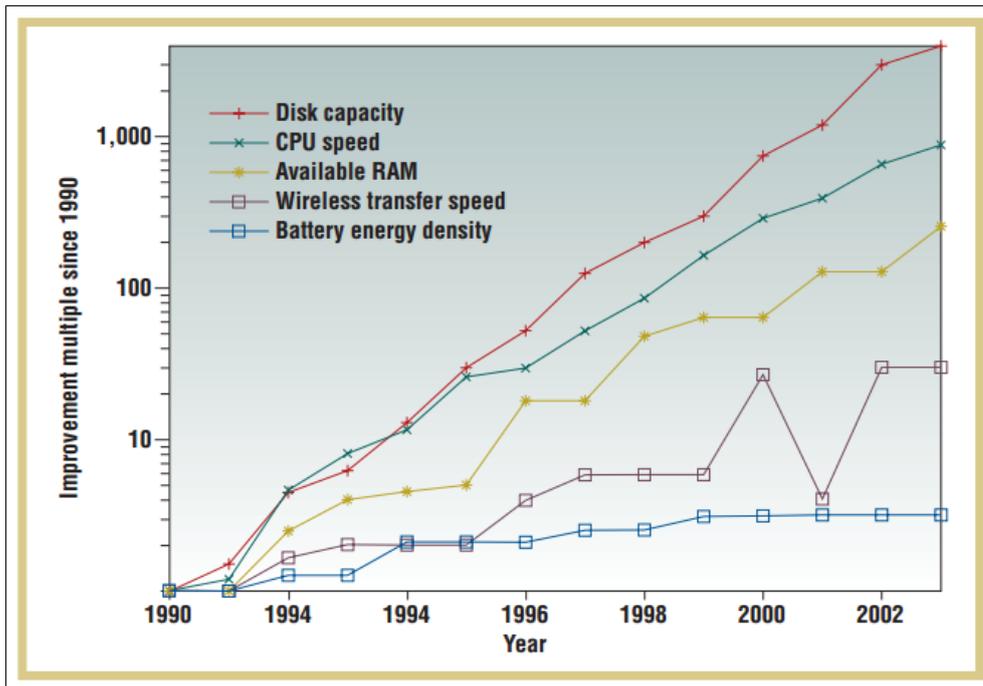


Figura 1 – Evolução das tecnologias dos dispositivos presentes nos smartphones

## 1.1 Objetivos

Visto, então, o grande uso de dispositivos móveis atualmente, na realização das mais diversas atividades, e o potencial crescimento da fatia de mercado da plataforma Android, este trabalho tem por objetivo estabelecer um método de análise do consumo energético dessas plataformas e, a partir disso, propor otimizações, visando à redução do consumo energético dos *smartphones*, uma vez que as baterias de lítio não acompanharam toda a evolução desses dispositivos, de modo a não reduzir a *performance* do ponto de vista dos usuários. Desse modo, foram estabelecidos, neste trabalho de pesquisa, os seguintes objetivos gerais:

1. construir um método de análise do consumo energético de dispositivos móveis;
2. analisar métodos de redução do consumo energético;
3. construir métodos a fim de encontrar pontos ótimos para o *tradeoff*, redução do consumo *versus* manutenção da *performance*.

Para atingir esses objetivos, durante a elaboração deste trabalho, foram vistos diversos métodos de otimização energética voltados para aplicações e/ou uso eficiente dos dispositivos presentes nos *smartphones*. Após serem realizados diversos testes e análises, chegou-se à conclusão de que uma maior economia seria alcançada se não fossem realizados muitos chaveamentos de frequência.

Desse modo, a fim de não realizar muitas mudanças de frequência, foram estabelecidos **grupos** de aplicações capazes de nortear os chaveamentos de frequência a fim de, com isso, reduzir o consumo energético do *smartphone* e, assim, **encontrar frequências com menor consumo para esses grupos de aplicações**.

Para alcançar esses objetivos, buscou-se o estudo realizado por Negata *et al.* (NAGATA; YAMAGUCHI; OGAWA, 2012), que leva em consideração não apenas a redução do consumo energético como também a manutenção da *performance* da aplicação. De acordo com esses autores, o processo de ajuste de frequência da CPU deve levar em conta o *feedback* da aplicação para ajustar a frequência do processador conforme a *performance* requerida.

Dessa forma, foram estabelecidos os seguintes objetivos específicos para esta pesquisa:

1. encontrar métodos estatísticos que pudessem avaliar o consumo do *smartphone* estudado durante a realização deste trabalho;
2. encontrar métodos numéricos que pudessem estimar a energia consumida por esse dispositivo a fim de mensurar a economia do dispositivo móvel e, assim, poder apontar quanto cada uma das abordagens consome;
3. utilizar o métodos estatísticos e numéricos a fim de encontrar pontos de menor consumo e, dessa forma, poder propor otimizações.

## 1.2 Motivação

Ashwini *et al.* (ASHWINI; THAWANI; SRIKANT, 2006) mostram que o chaveamento de frequência explora o fato de que a frequência de pico de um processador com tecnologia *CMOS* é proporcional à tensão de alimentação, enquanto que a quantidade de energia dinâmica requerida para a execução de uma determinada carga de trabalho é proporcional ao quadrado da tensão de alimentação. Dessa relação, tem-se que uma redução na tensão de alimentação, enquanto há uma diminuição na frequência de operação, permite uma redução quadrática no consumo energético desse dispositivo ao custo do aumento do tempo de execução da atividade.

De acordo com Hennessy *et al.* (HENNESSY; PATTERSON; ASANOVIĆ, 2012), a distribuição de potência, a remoção de calor e a prevenção de pequenas áreas com uma grande quantidade de calor dissipado têm se tornado extremamente difíceis de se obter com o avanço na tecnologia de *transistors*, fazendo que a potência dissipada seja uma das maiores restrições no uso dessa tecnologia. No entanto, os microprocessadores modernos passaram a oferecer as seguintes técnicas para tentar melhorar a eficiência energética:

**Fazer nada bem:** Os microprocessadores atuais são capazes de reconhecer quando estão ociosos, entrando automaticamente em um estado de baixo consumo de energia.

**Escalonamento dinâmico de frequência e tensão (DVFS):** Nesse caso, o microprocessador é inteligente o suficiente para chavear para uma frequência de mais baixo consumo, caso seja detectado um nível baixo de carga de trabalho no processador.

**Projeto para o caso típico:** Os dispositivos, como memória e armazenamento em massa, são postos em estado de baixo consumo, caso seja detectado muito tempo de ociosidade do microprocessador.

Desse modo, a técnica escolhida para realizar a redução do consumo energético do *smartphone* foi a de *DVS* pois, como foi mencionando por Ashwini *et al.* (ASHWINI; THAWANI; SRIKANT, 2006), uma redução da frequência de operação do processador pode chegar a uma redução no consumo na mesma proporção.

Ao realizarem-se os estudos sobre chaveamento de frequência e compararem-se as séries temporais geradas quando se chaveava constantemente a frequência em relação às séries geradas, notou-se uma diferença sutil quando esse fato não ocorria. Para evidenciar-se essa diferença, foram realizados dois experimentos. Cada um deles durou 30 minutos e foi feito em cada um dos seguintes cenários: 1) com chaveamento de frequência de 1 em 1 minuto; e 2) com uma frequência de operação constante. Os resultados são mostrados na Figura 2 e na Figura 3.

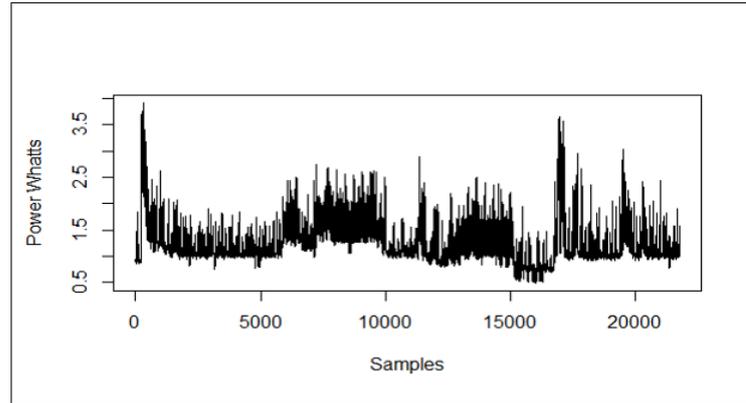


Figura 2 – Experimento com mudança de frequência

Observando-se a Figura 2, nota-se que, nas amostras no intervalo de 10000 a 15000, a potência instantânea se mantém no intervalo entre quase 1 *Watt* e quase 2,5 *Watts*. Por outro lado, conforme se pode ver na Figura 3, a potência se mantém entre 0,5 *Watt* e 1,5 *Watts*.

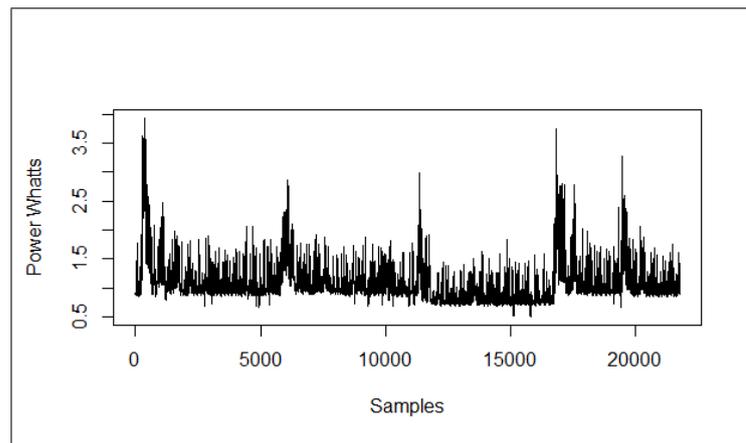


Figura 3 – Experimento sem mudança de frequência

Após visualizar essa diferença, viu-se que ela chega a **6%** em experimentos de 30 minutos. Apesar de ela conter um viés, pois no experimento realizado há uma chamada ao *Kernel* para realizar essa mudança, ao contrário do que acontece com os *governors* padrão do *Kernel*, sentiu-se a necessidade de estabelecer frequências ideais, de modo a não realizar chaveamentos a todo instante. Para se trabalhar com frequências ideais e não realizar chaveamentos a todo instante, resolveu-se adotar a estratégia de criar **grupos** de aplicações que passariam a operar em uma mesma frequência ideal ou mesmo em uma faixa de frequências aceitáveis. Essa estratégia será desenvolvida ao longo deste trabalho.



## 2 Referencial Teórico

Após notar as possibilidades de melhoria descritas na Seção 1.2, precisava-se investigar como os pesquisadores da área de otimização do consumo estavam realizando as medições e as otimizações energéticas.

Por isso empreendeu-se uma pesquisa a fim de verificarem-se os estudos recentes mais relevantes nas áreas de medição e otimização energética. Foram selecionados, então, 14 artigos, sendo 4 na área de técnicas de medição e 10 na área de otimização, uma vez que este trabalho tem por objetivo primeiro buscar uma forma de otimizar o consumo energético de um *smartphone* utilizando a plataforma Android.

A seguir são descritos os estudos encontrados na área de técnicas de medição que são relevantes para a realização deste trabalho.

### 2.1 Estudos sobre Técnicas de Medição de Energia

Na área de técnicas de medição de energia, conforme se registrou acima, elencaram-se 4 artigos que se mostraram interessantes para a discussão realizada neste trabalho sobre cálculo do consumo energético do dispositivo móvel. Eles constituíram um ponto de partida para analisar-se o que estava sendo produzido sobre o tema de medição do consumo energético.

No estudo realizado por Kalic *et al.* (KALIC; BOJIC; KUSEK, 2012), faz-se uma análise baseada no percentual de bateria gasto e no *throughput* de dados na realização das atividades de *download* e *upload*, usando-se as tecnologias *Wi-Fi*, *3G* e *Bluetooth*. Os autores mostram, nesse artigo, que a energia gasta pelas tecnologias *3G* e *Wi-Fi* são muito semelhantes, mas o *throughput* de dados da segunda é muito maior que o da primeira, ao contrário da tecnologia *Bluetooth*, que possui um gasto bem menor quando comparado às outras duas tecnologias, fazendo a bateria durar até 4 horas a mais. Esses autores propõem funções lineares que relacionam o tempo gasto e a energia consumida para as tecnologias *Bluetooth*, *Wi-Fi* e *3G*. Eles também propõem funções lineares que relacionam a quantidade de dados transferida à energia consumida para *download* e *upload*, usando as tecnologias acima citadas.

Já em Harjula *et al.* (HARJULA; KASSINEN; YLIANTTILA, 2012), é feita uma análise do consumo energético do *smartphone* devido às mensagens de sincronização. Esse tipo de mensagem inclui transações de rede consistindo de pequenas mensagens enviadas e recebidas, transmitidas em pacotes únicos, como: mensagens de manutenção, mensagens instantâneas e busca de serviço. De acordo com os autores desse artigo, é necessário

contabilizar a energia gasta por esses tipos de comunicação e, assim, eles propõem um modelo de consumo de energia avançado para dispositivos móveis. Eles mostram que, de acordo com Balasubramanian *et al.* (BALASUBRAMANIAN; BALASUBRAMANIAN; VENKATARAMANI, 2009), as redes 3G e GSM possuem um alto consumo energético e pontuam que isso vem se tornando um grande problema em redes que possuem sinalização muito frequente, como é o caso das redes P2P.

Desse modo, Harjula *et al.* (HARJULA; KASSINEN; YLIANTTILA, 2012) constroem um modelo energético baseado em dois elementos de consumo: transferência de sinalização e mídia. De acordo com o modelo proposto nesse estudo, o consumo energético total de mandar ou receber uma transmissão de dados é dividido da seguinte forma:

**Ramp Energy:** representa a energia gasta para realizar a transição do estado *idle* para o estado de transferência de dados;

**Transfer Energy:** representa a energia gasta na transferência da informação em si;

**Tail Energy:** constitui a energia gasta durante os estados de potência intermediários depois da transmissão.

Dessa forma, esses autores propõem um modelo definido pela equação (2.1)

$$P(t) = \max[P_{sig}(t), P_{med}(t)] \quad (2.1)$$

O modelo foi implementado usando-se Matlab comparado quanto à sua acurácia, utilizando-se medições reais e chegou-se a bons resultados, pois, na rede WCDMA, 72% dos valores simulados estão dentro dos pontos de amostra com uma significância de 5%. Entretanto, vale a pena destacar que o modelo proposto aplica-se apenas às mensagens de sinalização.

Por sua vez, Xiao *et al.* (XIAO; KALYANARAMAN; YLA-JAASKI, 2008) fizeram um estudo sobre o consumo energético do site *mobile* do Youtube, realizando, assim, uma comparação dos serviços do Youtube que cobrem os recursos de tecnologia de acesso à rede, tecnologias de *download* e tecnologias de armazenamento do *smartphone* N95 da Nokia. Esses autores também empreenderam um estudo comparativo entre WCDMA e WLAN na realização das seguintes atividades: *download* seguido de execução e execução local. Baseados nos resultados dessas análises, eles propuseram a criação de um modelo para caracterizar o consumo energético do site *mobile* do Youtube.

Nas análises realizadas por Xiao *et al.* (XIAO; KALYANARAMAN; YLA-JAASKI, 2008), foi considerada, além da energia gasta pelo programa de análise e outras *threads* de serviço, a temperatura da bateria. Nesse estudo foi feita a medição do consumo para transferência do vídeo pela rede, para exibição local e para a interação do dois. A fim de procederem a essas análises, esses autores utilizaram o **Nokia Energy Profiler**.

Por outro lado, Gupta *et al.* (GUPTA *et al.*, 2012) realizaram a medição do consumo energético durante as diversas atividades envolvidas para o estabelecimento de uma ligação em uma rede 2G/3G. Esses autores analisaram três cenários para fazer essa mensuração: a) acesso à rede; b) sinalização durante a chamada de voz; e c) sinalização quando o telefone está em movimento e passa por uma área sem sinal de telefonia. Eles trabalharam com potência média, e os resultados mais relevantes desse estudo são os seguintes: a) o consumo energético durante o acesso à rede é em torno de 22 *Joules*; b) há um aumento substancial do consumo durante o processo de sinalização comparado à chamada em si; e c) o consumo energético na rede 3G é superior ao consumo da rede 2G na realização das mesmas atividades.

Como é possível perceber-se, muitos dos estudos de medição do consumo trabalham com modelagem e simulação, ou potência média, e esses dois fatos precisam ser considerados com bastante cuidado, pois existem trabalhos, como o realizado por Huang *et al.* (HUANG; YANG, 2013), que mostram que o estudo das séries da potência elétrica geradas durante o uso de um *smartphone* não é tão simples, porque as séries não possuem as condições de estacionariedade bem definidas na maioria dos casos.

Alawnah *et al.* (ALAWNAH; SAGAHYROON, 2013) fazem previsões de demanda de energia elétrica. Esse trabalho apresenta uma revisão das técnicas de previsão de demanda de energia elétrica e os vários tipos de metodologias e modelos estão incluídos na literatura. Previsão de carga pode ser dividido em três categorias: a) as previsões de curto prazo, que vão geralmente de uma hora a uma semana; b) as previsões médias, que vão geralmente de uma semana a um ano; e c) as previsões de longo prazo, que são de mais de um ano. Assim, com base nos vários tipos de estudos apresentados nesses documentos, as técnicas de previsão de carga podem ser apresentadas em três grandes grupos: técnica de previsão tradicional, técnica tradicional modificada e técnica *soft computing*.

Também em Huang *et al.* (HUANG; YANG, 2013), é feita uma previsão da carga de curto prazo. As séries obtidas nesse artigo possuem características não lineares e não estacionárias. Tais características implicam dificuldades na construção de um modelo adequado para prever com precisão as variações de carga. Esses autores usam os métodos de regressão estatística, modelos de séries temporais estocásticos e computação evolutiva para gerar os modelos, chegando a resultados interessantes.

Entretanto, ao se realizar a análise das técnicas de medição que estão sendo usadas atualmente, notou-se uma forte limitação nelas. Isso se deve à forma como os autores acima citados analisam os dados coletados, pois, como é possível perceber pelos estudos descritos anteriormente, todos trabalham com experimentos simulados, ou, quando realizam experimentos reais, não possuem um método estatístico e numérico para fazerem a análise das amostras coletadas. Isso porque eles descrevem o consumo por meio de uma regressão linear e, como é mostrado ao longo deste trabalho e pelos estudos realizados por

Huang *et al.*(HUANG; YANG, 2013), as séries de potência encontradas não podem ser descritas dessa forma.

Devido a isso, viu-se a necessidade de, antes de fazer qualquer busca por metodologias que levassem a uma maior economia energética, estabelecer métodos estatísticos e numéricos apropriados para empreender a análises dos experimentos realizados durante esta pesquisa. Dessa forma, recorreu-se a algumas teorias, a saber:

- Teste de Hipóteses para averiguar estatisticamente as melhorias a serem realizadas;
- Valor-P para auxiliar na construção do resultado do teste de hipóteses;
- Teste de Aderência a fim de identificar se as amostras coletadas provinham de alguma distribuição de probabilidade conhecida;
- Teste de Lilliefors, um teste de aderência não-paramétrico, após identificar-se que não era possível realizar qualquer tipo de teste que envolvesse média e desvio-padrão e com isso a impossibilidade de usar testes paramétricos devido às características das séries de potência obtidas;
- Teste de Wilcoxon-Mann-Whitney, também não-paramétrico, para comparar duas amostras obtidas quanto à igualdade ou atestar que uma era estatisticamente maior que a outra.

Além dessas teorias, precisava-se estabelecer um método de análise numérica a fim de mensurar as otimizações que viessem a ser feitas durante a realização deste trabalho. Assim, a seguir, descreve-se cada uma das teorias associadas aos métodos estatísticos e numéricos utilizados durante a realização deste trabalho.

## 2.2 Teste de Hipóteses

Em estatística, para se inferir alguma coisa a respeito de uma amostra, como a média da população da qual ela veio, a distribuição de probabilidade que ela segue e a variância da população da qual essa amostra vem, utiliza-se a inferência estatística. Para comprovar-se a veracidade de uma afirmação feita, usa-se um teste de hipóteses. Desse modo, os testes de hipóteses dispõem uma metodologia que permite verificar se os dados amostrais trazem evidências que apoiam ou refutam uma hipótese estatística formulada.(BUSSAB; MORETTIN, 2010)

A metodologia definida para a construção de um teste de hipóteses é a seguinte(BUSSAB; MORETTIN, 2010):

1. Fixa-se a hipótese  $H_0$  a ser testada e a hipótese alternativa.

2. Usa-se a teoria estatística e a informação da amostra para decidir qual estatística/estimador será usada para testar a hipótese  $H_0$ . A partir disso, obtêm-se as propriedades desse estimador (média, desvio-padrão e distribuição).
3. Fixa-se a probabilidade máxima de cometer um erro de aceitar a hipótese nula sendo ela falsa ( $\alpha$ ). E utiliza-se esse valor para construir a região crítica (região de rejeição de  $(H_0)$ ).
4. Usam-se as observações da amostra para calcular o valor da estatística do teste.
5. Se o valor da estatística não pertencer à região crítica, não se rejeita  $H_0$ ; caso contrário,  $H_0$  é rejeitada.

## 2.3 Valor-p

De acordo com o que foi descrito na Seção 2.2, o método de construção de um teste de hipóteses parte da fixação do nível de significância  $\alpha$ . Desse modo, pode-se argumentar que esse procedimento pode levar à rejeição da hipótese  $H_0$  para um determinado valor de  $\alpha$  e à não rejeição para um valor menor. Uma outra maneira de proceder é apresentar a probabilidade de significância, ou nível descritivo, ou ainda valor-p do teste. Os passos são semelhantes aos apresentados na Seção 2.2 e a diferença está em não construir a região crítica e sim indicar a probabilidade de ocorrerem valores da estatística mais extremos que o observado e sob a hipótese de  $H_0$  ser verdadeira (BUSSAB; MORETTIN, 2010).

Normalmente, para testar-se uma hipótese usando-se o valor-p, utiliza-se a escala de evidência sugerida por Fisher. Desse modo, a escala de Fisher contra  $H_0$  é mostrada na Tabela 1:

Tabela 1 – Escala de Significância de Fisher.

Valor-P	0,10	0,05	0,025	0,01	0,005	0,001
Natureza da Evidência	Marginal	Moderada	Substancial	Forte	Muito Forte	Fortíssima

Assim, por exemplo, um valor-p igual a 0,01 indica uma evidência forte contra a validade de  $H_0$ . Essa escala usa como referência o valor 0,05 ou 5%, e as considerações feitas por Fisher referiam-se a testes Qui-quadrado (BUSSAB; MORETTIN, 2010).

## 2.4 Teste de Aderência

Tem-se uma população  $P$  e deseja-se verificar se ela segue uma distribuição especificada  $P_0$ , isto é, realizar-se um teste de hipótese em que  $H_0: P=P_0$ .

Existem diversos testes estatísticos que podem ser usados para fazer esse tipo de verificação, dependendo da necessidade do pesquisador, a saber: Teste Binomial, Teste Qui-quadrado e Teste Kolmogorov-Smirnov (SIEGEL; CASTELLAN, 2006).

No **Teste Binomial** o pesquisador possui apenas duas classes nas quais deseja dividir as diversas populações observadas.

Já o **Teste Qui-Quadrado** é adequado para os casos em que o pesquisador possui diversas classes nas quais deseja dividir as diversas populações estudadas. Esse teste mede o grau de concordância entre a distribuição esperada e observada por meio da soma dos quadrados das diferenças entre cada frequência observada e esperada dividida pela frequência esperada correspondente sobre "k" categorias, conforme mostra a equação (2.2).

$$\chi^2 = \sum_{i=1}^S \frac{(O_i - E_i)^2}{E_i} \quad (2.2)$$

Se a concordância for boa, a estatística do teste ( $\chi^2$ ) será pequena, pois as diferenças mostradas na equação (2.2) serão pequenas. No entanto, essa estatística possui a limitação de não poder mensurar o "poder do teste" com facilidade quando se têm dados nominais ou categorizados, ou quando os dados consistem em frequências em categorias inerentemente discretas.

Já o **Teste Kolmogorov-Smirnov** é outro teste de aderência em que é especificada a distribuição de frequências acumuladas que ocorreria dada a distribuição de probabilidade teórica e compará-la com a distribuição de frequência acumulada observada. A distribuição teórica representa o que seria esperado sob  $H_0$ . O ponto no qual essas duas distribuições, teórica e observada, sofrem a maior divergência é determinado. Esse teste admite que a distribuição da variável subjacente que está sendo testada é contínua, como especificado pela distribuição de frequências acumuladas. Dessa forma, esse teste é apropriado para verificar a aderência para variáveis que são medidas pelo menos de forma ordinal. A estatística do teste pode ser definida pela equação (2.3).

$$D = \max_{1 \leq i \leq n} |F(x_i) - F_e(x_i)| \quad (2.3)$$

Para o caso estudado neste trabalho, como se tem um escala de valores ordinais, optou-se por usar **Teste de Kolmogorov-Smirnov** para testar a aderência das distribuições de probabilidade obtidas experimentalmente na forma de séries temporais de potências instantâneas.

Para testar se uma dada distribuição obtida experimentalmente segue uma distribuição normal, optou-se por usar um caso particular do **Teste Kolmogorov-Smirnov**, conhecido como **Teste Lilliefors**, conforme descrito na Seção 2.5.

## 2.5 Teste Lilliefors

O **Teste Lilliefors** é um teste de aderência para uma função de distribuição de probabilidade empírica para a hipótese composta de normalidade (RPROJECT, 2012).

Esse teste realiza a máxima diferença absoluta entre uma função de distribuição cumulativa empírica e a hipotética. Ela pode ser computada pela estatística 2.4, sendo  $D^+$  e  $D^-$  definidas pelas equações representadas em 2.5.

$$D = \max\{D^+, D^-\} \quad (2.4)$$

$$D^+ = \max_{i=1, \dots, n} \left\{ \frac{i}{n} - P(i) \right\}, D^- = \max_{i=1, \dots, n} \left\{ P(i) - \frac{(i-1)}{n} \right\} \quad (2.5)$$

Em que  $P(i) = \phi\left(\frac{[x_i - \bar{x}]}{s}\right)$ , sendo  $\phi$  a função de distribuição cumulativa da distribuição normal padrão, e  $\bar{x}$  e  $s$  média e desvio-padrão dos valores dos dados. O valor-p é computado da fórmula exposta em Cristea *et al.* (DALLAL; WILKINSON, 1986), na qual é alegado que a estatística só é confiável quando o valor-p é menor que 0,1. Caso ele seja maior que 0,1, ele passa a ser calculado a partir da distribuição da estatística modificada mostrada na equação (2.6) (RPROJECT, 2012).

$$Z = D(\sqrt{n} - 0,01 + \frac{0,85}{\sqrt{n}}) \quad (2.6)$$

De acordo com esse teste, em  $H_0$  a distribuição empírica segue uma distribuição normal e em  $H_1$  esse fato não ocorre.

Os valores críticos da estatística para aceitar ou rejeitar  $H_0$  em um determinado nível de significância podem ser consultados em uma tabela. Para amostras maiores que 35 e um nível de significância  $\alpha = 5\%$ , o valor da estatística do teste,  $D$  pode ser calculada conforme equação (2.7).

$$D^* = \frac{1,36}{\sqrt{N}} \quad (2.7)$$

Se o valor da estatística do teste for maior que o valor crítico,  $H_0$  é rejeitada e, portanto, a distribuição da qual a amostra foi retirada não pode ser aproximada para uma distribuição normal.

## 2.6 Teste de Wilcoxon-Man-Whitney

Quando são obtidas mensurações ordinais para as variáveis estudadas, o teste de Wilcoxon-Man-Whitney pode ser usado para testar se dois grupos independentes foram

extraídos de uma mesma população. Ele é um teste não paramétrico muito poderoso, constituindo-se uma ótima alternativa ao teste paramétrico T de Student.

Ele é um teste de hipóteses segundo o qual, supondo-se duas populações independentes,  $X$  e  $Y$ , testa-se em sua hipótese nula se as duas populações possuem a mesma distribuição. A hipótese alternativa testa se  $X$  é estatisticamente maior que  $Y$ , construindo dessa forma uma hipótese direcional. De acordo com esse teste, aceita-se  $H_1$  apenas se a probabilidade de um escore de  $X$  ser maior que um escore de  $Y$  for maior que 50%. Assim  $H_0$  postula que:  $P(X > Y) > \frac{1}{2}$  (SIEGEL; CASTELLAN, 2006).

O método para aplicar o teste de Wilcoxon-Mann-Whitney funciona da seguinte forma: seja “m” o número de casos na amostra do grupo  $X$  e “n” o número de casos na amostra do grupo  $Y$ , assume-se que as duas amostras são independentes, combinam-se os escores de ambos os grupos e dispõem-se esses escores em postos em ordem crescente de tamanho. A seguir, foca-se em um dos grupos, como, por exemplo, o grupo  $X$  com “m” casos. O valor de  $W_x$  (estatística usada no teste) é dado pela soma dos postos no primeiro grupo. Da mesma forma,  $W_y$  será a soma dos postos no segundo grupo.

Para determinar a significância de  $W_x$ , como se têm amostras com valores de “m” e “n” maior que 10, faz-se isso por meio de uma aproximação à distribuição normal pela equação definida em 2.8 e acessa-se a significância de  $Z$  por uma tabela de estatísticas de uma distribuição normal.

$$Z = \frac{W_x \pm 0,5 - \mu_{w_x}}{\sigma_{w_x}} = \frac{W_x \pm 0,5 - m \times \frac{(N+1)}{2}}{\sqrt{m \times n \times \frac{(N+1)}{12}}} \quad (2.8)$$

Se o valor observado de  $W_x$  tem uma probabilidade associada menor ou igual a  $\alpha$  (nível de significância), rejeita-se  $H_0$  em favor de  $H_1$  (SIEGEL; CASTELLAN, 2006).

## 2.7 Método de integração Numérica

Para que se pudesse calcular o consumo aproximado de uma amostra, fez-se necessário estabelecer um método que pudesse realizar uma aproximação para a integral mostrada na equação (2.9), que representa a energia total consumida por um dispositivo ao longo do tempo.

$$E = \int P(t)dt \quad (2.9)$$

Dessa forma, recorreu-se à integral de Riemann, que é definida a partir da soma de Riemann e do conceito de partição.

De acordo com Guidorizzi (GUIDORIZZI, 2001), uma partição  $P$  de um intervalo  $[a, b]$  é um conjunto finito  $P = x_0, x_1, x_2, \dots, x_n$  tal que  $a = x_0 < x_1 < x_2 < \dots < x_n = b$ .

Segundo esse autor, uma partição  $P$  de  $[a, b]$  divide  $[a, b]$  em 'n' intervalos  $[x_{i-1}, x_i]$  em que  $i = 1, 2, 3, \dots, n$ . A amplitude do intervalo  $[x_{i-1}, x_i]$  será definida por  $\Delta x_i = x_i - x_{i-1}$ . Essas diferenças não são necessariamente iguais e a maior delas denomina-se amplitude da partição  $P$  e é expressa por  $\max \Delta x_i$ .

O conceito de soma de Riemann pode ser definido da seguinte forma (GUIDORIZZI, 2001): seja  $f$  uma função definida em  $[a, b]$  e  $P : a = x_0 < x_1 < x_2 < \dots < x_n = b$  uma partição de  $[a, b]$ , para cada índice  $i$ , seja  $c_i$  um número em  $[x_{i-1}, x_i]$  um número escolhido arbitrariamente, o número  $\sum_{i=1}^n f(c_i) \cdot \Delta x_i = f(c_1) \cdot \Delta x_1 + f(c_2) \cdot \Delta x_2 + \dots + f(c_n) \cdot \Delta x_n$  denomina-se soma de Riemann de  $f$  relativa à partição  $P$  e ao número  $c_i$ .

Dessa forma, a integral de Riemann pode ser definida no intervalo  $[a, b]$  pela seguinte equação (2.10) (GUIDORIZZI, 2001).

$$\int_a^b f(x) dx = \lim_{\max \Delta x_i \rightarrow 0} \sum f(c_i) \cdot \Delta x_i \quad (2.10)$$

Para aplicar esse conceito da integral de Riemann, recorreu-se a um método capaz de realizar essas somas computacionalmente e, para suprir essa necessidade, utilizou-se o método de integração numérica dos trapézios conforme mostrado pela equação (2.11), que, ao se desenvolver, resulta na equação (2.12) que pode ser calculada através da função construída *software* estatístico R.

$$I = \int_a^b f(x) dx \approx \int_a^b f_1(x) dx \quad (2.11)$$

$$I = (b - a) \times \frac{f(b) + f(a)}{2} \quad (2.12)$$

Essa função é descrita no pseudocódigo abaixo.

Código Fonte 2.1 – Código para realizar integração numérica

```

1 function (samples , time , samplesPerTime)
2 {
3
4   maxSamples = length (samples) / time ;
5
6   if (maxSamples < samplesPerTime) {
7     samplesPerTime = maxSamples ;
8   }
9
10  newSamples = interval (samples , samplesPerTime) ;
11
12  return trapezoidal (newSamples) ;
13 }
```

Na linha **1** a função responsável pelo cálculo do consumo é declarada. Tomam-se como parâmetros as amostras, o tempo de amostragem e a quantidade de amostras por segundo que se quer. A estratégia de usar amostragens diferentes é usada para provar o conceito de estabilização do resultado do consumo. Outra estratégia utilizada para essa finalidade foi a adoção de outros métodos numéricos, como a *regra de simpson*, só para verificar a proximidade dos resultados.

Na linha **4** o número máximo de amostras por segundo é calculada apenas para efeito de verificação do limite máximo, verificação esta feita na linha **6**. Caso o limite superior seja menor que o número de amostras por segundo, automaticamente é tomado esse limite para ser usado pela função *interval*, chamada na linha **10**. Essa função é responsável por gerar um novo vetor de amostras com os valores mostrados de acordo com a frequência especificada pela variável *samplesPerTime*.

Após isso, na linha **10**, o novo vetor de amostras é gerado e usado para cálculo do consumo pela função *trapezoidal* na linha **11**, retornando assim o valor do consumo aproximado da amostra.

Após estabelecerem-se os métodos necessários para realizarem-se as análises estatísticas e numéricas das amostras coletadas a fim de demonstrarem-se melhorias que viessem a ser encontradas, precisava-se buscar os métodos de otimização energética utilizados atualmente dentro do sistema operacional Android pelos pesquisadores da área de otimização energética. Foram elencados, então, 10 artigos que serão discutidos na próxima seção.

## 2.8 Estudos sobre Otimização do Consumo

Sobre otimização do consumo elencaram-se 10 artigos para serem analisados por considerar-se que o principal foco deste trabalho é buscar entender e aplicar otimizações energéticas em dispositivos móveis que utilizam o sistema operacional Android. Durante a leitura deles, percebeu-se que uma das técnicas mais recorrentes para se realizarem otimizações é a de escalonamento de frequência (*DVFS*).

No estudo realizado por Liang *et al.* (LIANG; LAI; CHIOU, 2010), esses autores defendem que, a fim de minimizar-se o consumo energético do dispositivo móvel, não basta apenas configurar a frequência de operação dele para a menor possível. Eles sustentam que há uma frequência de operação crítica. Nesse estudo, foi utilizado um processador *ARM*, que possui alguns registradores especiais capazes de fornecer métricas de *performance* da CPU. Esses registradores fazem parte do chamado *PMU (Performance Management Unit)* e, por isso, são denominados de registradores *PMU*. O cálculo da frequência crítica de operação é feito a partir da taxa de acesso à memória que pode ser calculada pela equação (2.13).

$$MAR = \frac{N_{cache\_miss}}{N_{instrexec}} \quad (2.13)$$

Liang *et al.* (LIANG; LAI; CHIOU, 2010) puderam observar que os programas com baixa taxa de acesso à memória podem se beneficiar pouco da redução de frequência do processador, pois esse tipo de programa é bastante dependente da CPU, e uma redução de frequência dela acarretaria um aumento no tempo de execução do *software*, o que aumentaria o consumo energético. Já os programas que são limitados por muitos acessos à memória, ou seja, com altas taxas de acesso à memória, beneficiam-se muito mais da redução da frequência da CPU, pois essa diminuição não altera muito o tempo de execução do programa, uma vez que o mesmo é dominado por acesso à memória.

Esses autores desenvolveram uma equação chamada MAR-CSE (Memory Access Rate-Critical Speed Equation), que descreve formalmente a relação existente entre a taxa de acesso à memória e a frequência crítica do processador. Ela pode ser definida pela equação (2.14).

$$y = -30474.04X^3 + 1885.97X^2 - 37.82X + 0.797 \quad (2.14)$$

Usando-se o *governor* AD-DVFS desenvolvido por Liang *et al.* (LIANG; LAI; CHIOU, 2010), que utilizam a equação acima mencionada para calcular a frequência crítica do processador, obteve-se de 37,9% a 47,7% de economia, comparada à do *governor* padrão **Performance**, que coloca o processador sempre na frequência máxima.

O fato mostrado neste trabalho é corroborado por Mittal *et al.* (MITTAL; SINGHAL; SETHIA, 2013), trabalho no qual se destaca que, dentre muitos fatores que determinam a escala de frequência, o uso do processador é o mais proeminente. Esse estudo traz uma estimativa do poder de processamento total necessário ao sistema em um dado instante. No entanto, esses autores destacam que a aplicação que está em execução em primeiro plano ajuda a identificar a máxima frequência de operação necessária ao sistema durante toda a execução da aplicação. Configurando-se a frequência ideal para uma aplicação, saltos desnecessários em níveis de frequência desnecessárias podem ser eliminados, promovendo um uso mais eficiente do dispositivo do ponto de vista energético.

Nesse estudo, os autores propuseram, então, um novo modelo de escalonamento de frequência que considera o uso do processador e a aplicação que está em execução em primeiro plano para ajudar a identificar o nível máximo de frequência que será necessário durante toda a execução da aplicação.

A categorização é feita dinamicamente sempre que uma aplicação é instalada ou executada pela primeira vez, e essa informação pode ser armazenada para uso futuro. As categorias são mapeadas para o limite máximo de frequência que é usado para configurar

a frequência correspondente durante o tempo de execução. O *governor* desenvolvido executa, no espaço do usuário, na forma de uma aplicação chamada **Appspace**. Logo que a aplicação é executada, a frequência correspondente à categoria da aplicação é configurada. A cada intervalo de tempo predeterminado, o uso do processador é aferido e um dos seguintes casos pode acontecer:

**Uso da CPU > Limiar superior:** Nesse caso, a frequência é configurada para a frequência máxima da categoria à qual a aplicação pertence.

**Uso da CPU < Limite inferior:** Nesse caso, a frequência da CPU será configurada para a frequência imediatamente anterior à atual e, caso ela já seja a menor frequência, permanecerá inalterada.

**Limite Inferior <= Uso da CPU <= Limite Superior:** Nesse caso, a frequência da CPU ficará inalterada.

De acordo com experimentos realizados por Mittal *et al.* (MITTAL; SINGHAL; SETHIA, 2013), a queda de tensão da bateria é 52,63% menor, quando usado o *governor* desenvolvido por eles em relação ao **OnDemand**.

Em outro trabalho que corrobora a ideia de otimização energética, utilizando chaveamento de frequência, Kwon *et al.* (KWON; CHAE; WOO, 2013) propõem um algoritmo de escalonamento que diminui o processamento de uma aplicação tanto quanto possível. Por meio desse cálculo da quantidade mínima de ciclos requeridos para atender às necessidades da aplicação, o algoritmo não perde os "deadlines" do processamento enquanto ele está executando lentamente. Os resultados experimentais mostram que o algoritmo permite ao escalonador do sistema operacional manter a tensão do processador baixa e, assim, viabiliza que o consumo energético seja reduzido em mais de 32%, quando a carga do sistema está em 30% da capacidade.

Esses autores desenvolveram um algoritmo que é executado em nível de aplicação e auxilia os algoritmos de escalonamento de frequência tradicionais presentes nos sistemas operacionais. Eles propuseram tais algoritmos porque notaram que os tradicionais são ineficientes quando se têm aplicações com rajadas de processamento que ocupam um curto intervalo de tempo do processador.

Para realizar essa função, Kwon *et al.* (KWON; CHAE; WOO, 2013) inseriram um algoritmo de escalonamento nos códigos de processamento das aplicações chamando a API, responsável por colocar o processador em estado de baixo consumo energético (modo *sleep*) após algumas rajadas de processamento na unidade de tempo. Para realizar isso, foi necessário: (a) reorganizar o código de processamento de dados em interfaces que permitissem execução incremental; e (b) definir uma unidade de tempo para decidir com

que frequência o algoritmo de escalonamento chama a API que coloca o processador em estado de baixo consumo energético (**Modo Sleep**).

Dessa forma, durante a execução, o algoritmo de escalonamento proposto realiza uma certa quantidade de processamento de dados e "dorme" no resto do intervalo de tempo. Caso o tempo de execução da aplicação exceda a unidade de tempo (perca o *deadline*), o algoritmo não executa seu estado de dormência.

Para decidir quão lenta a execução deve ser, o algoritmo de escalonamento calcula a quantidade mínima de ciclos por unidade de tempo requerida a fim de atender a todos os *deadlines* de todos os códigos de processamento de fluxos de dados. Assim, obtém-se a taxa mínima de execução, que é o total esperado de ciclos para finalizar o trabalho dividido pelo *deadline* do trabalho, e a taxa de execução total requerida para finalizar múltiplos trabalhos é obtida somando-se a taxa de execução requerida para cada trabalho.

Dessa forma, conforme mostra a equação (2.15), a taxa total de execução multiplicada pela unidade de tempo passa a ser a quantidade mínima de ciclos necessários por unidade de tempo. Na equação (2.15),  $C_i$  é o número de ciclos esperados para processar cada conjunto de dados em um  $i$ -ésimo fluxo,  $t_u$  é a unidade de tempo e  $n$  é o número de fluxos.

$$\left( \sum_{i=1}^n \frac{C_i}{D_i} \right) \times t_u \quad (2.15)$$

Para demonstrar o efeito da unidade de tempo, foi mensurado o consumo com unidades de tempo variadas. Kwon *et al.* (KWON; CHAE; WOO, 2013) chegaram, então, à conclusão de que, quando se variam as unidades de tempo de 100ms a 2000ms, tem-se um consumo ótimo com uma unidade de tempo igual a 100ms, que é igual ao tempo do escalonador do sistema operacional. Quando o sistema está com uma carga bastante baixa (menos que 70% da capacidade), o processamento com o algoritmo proposto consome menos potência que os algoritmos tradicionais. Quando a carga de trabalho está em 30%, o sistema consome 32% menos, o que significa uma extensão do ciclo de vida da bateria em 47%. À medida que a carga de trabalho aumenta, a economia com o algoritmo proposto diminui até o ponto de o consumo tornar-se igual à energia consumida, usando-se a abordagem tradicional.

Assim, neste trabalho, busca-se uma melhoria em relação ao algoritmo proposto por Kwon *et al.* (KWON; CHAE; WOO, 2013), procurando-se analisar, entender e modelar o consumo de tal forma, que se faça o que esses autores fizeram, sem precisar-se reestruturar o código da aplicação.

Não se pode deixar de registrar também que outros estudos centram-se em comparar as abordagens de economia de energia já existentes, como é o caso da comparação

realizada por Datta *et al.* (DATTA; BONNET; NIKAEIN, 2012) que comparam as abordagens adotadas pelo aplicativos SetCPU, CPU Tuner e Juice Defender, e classificam essas aplicações em dois grupos:

**Escala de frequência do processador:** Nesta categoria, estão os aplicativos que ajustam os *governors* e configuram a frequência de operação do processador, de modo a tentar economizar energia, utilizando a prerrogativa de que a potência e a frequência se relacionam pela equação (2.16), em que  $C$  é a carga capacitiva,  $V$  é a tensão de alimentação e  $F$  é a frequência de operação do circuito.

$$P = CV^2F \quad (2.16)$$

**Controle dos recursos do *smartphone*:** Nesta categoria, estão as aplicações que realizam economia de energia desligando dispositivos do *smartphone*, caso eles não estejam sendo usados em um determinado momento ou local.

Os aplicativos SetCPU e CPU Tuner mudam o *governor* para permitir o controle de frequência do processador de forma manual. Por meio de um monitor, essas aplicações são capazes de monitorar as condições do dispositivo móvel e configurar as condições impostas nos perfis energéticos configurados nesses aplicativos.

O aplicativo Juice Defender detém-se em controlar os recursos do dispositivo móvel, como desabilitar conectividade e reduzir o brilho da tela, a fim de reduzir o consumo. Ele também possui um monitor capaz de analisar o contexto do telefone, tal como local e hora, e assim ajustar o dispositivo de acordo com as condições do perfil energético ativo.

Entretanto há limitações nesses aplicativos:

1. a definição estática de perfis energéticos;
2. a parte de controle desses aplicativos não é inteligente, fazendo que, se determinada condição for atingida, um determinado perfil é ativado, mas a volta a uma condição anterior não é feita de forma automática;
3. para os aplicativos SetCPU e CPU Tuner, o dispositivo tem que dar permissões especiais;
4. ao contrário do que se espera dessas aplicações, nem sempre o ajustamento de frequência do processador deve ser o foco principal;
5. a informação de contexto do Juice Defender não é usada para qualquer processo de geração de padrão;

6. esses aplicativos não enfocam o aprendizado do padrão de consumo energético do usuário.

Têm-se feito também análises de atividade das aplicações. Zhu *et al.* (ZHU; REDDI, 2013) e Thiagarajan *et al.* (THIAGARAJAN *et al.*, 2012) realizaram uma análise do consumo para renderização de páginas *web*, usando o aplicativo Browser, padrão para exibição de páginas. Thiagarajan *et al.* (THIAGARAJAN *et al.*, 2012) fizeram um estudo sobre o custo energético de renderização de *sites* populares, como Facebook, Amazon e outros.

Os autores acima citados destacam que elementos, como Javascripts complexos e CSS, podem ser tão caros, energeticamente falando, quanto a renderização de uma imagem. Além disso, requisições de Javascript dinâmicas podem aumentar consideravelmente o custo energético de renderizar uma página, pois não permitem que ela seja armazenada. E, por fim, esses autores mostram que o carregamento de uma imagem JPEG é muito menos custosa em termos energéticos que outros formatos, como GIF e PNG.

Esses autores detectaram ainda que o grande consumo de energia por parte de Javascript se dá devido ao fato de as páginas carregarem grandes arquivos Javascript para renderizarem-se de forma completa, mesmo não utilizando todo o *script* por página. Como exemplo, eles citam o fato de a Wikipedia gastar em torno de 10 *Joules* para renderizar o Javascript, o que representa cerca de 30% do total de energia gasto na página. Para provar a economia, os autores refizeram os *scripts*, substituindo as funções usadas por *scripts* genéricos, como o **Jquery**, e gastaram apenas 9,5 *Joules* para renderizar a página com os *scripts* simplificados contra 15 *Joules* para a renderização com os *scripts* genéricos.

De forma semelhante, esses autores apontaram que há um grande consumo de energia na renderização de grandes arquivos CSS com muitas regras não usadas. Um exemplo que eles citam é o consumo do CSS da página da Apple, que é em torno de 12 *Joules*, pois ela requer o carregamento de 5 arquivos CSS diferentes com regras que são usadas na página. Com uma versão simplificada do CSS dessa página, eles conseguiram gastar apenas 5 *Joules*, o que representa cerca de 40% do total de energia gasto pelo site original. Substituindo as imagens originais de um *site* por imagens JPEG, esses autores chegaram a uma economia de 20%, no caso do *site* da Amazon, e de 30%, no caso do Facebook.

Por sua vez, Zhu *et al.* (ZHU; REDDI, 2013) se preocuparam em caracterizar o custo energético de cada *tag HTML*, conforme mostrado na Tabela 2.

Tabela 2 – Consumo das Tags HTML

Category	Microbenchmark	Time(ms)		Energy(mJ)	
		Abs.	Norm.	Abs.	Norm.
Tag	h3	8.85	1.0	26.31	1.0
	li	14.25	1.6	44.97	1.7
	option	21.48	2.4	67.12	2.6
	table	20.96	2.4	69.82	2.7
	input	44.92	5.0	166.28	6.3
	img	177.03	20.0	611.53	23.0
Attribute	cellspacing	17.30	1.0	63.45	1.0
	border	23.46	1.4	81.87	1.3
	bgsolor	28.05	1.6	97.86	1.5
	background	337.18	19.0	1145.75	18.0
DOM tree node count	1k	325.65	1.0	1103.98	1.0
	2k	560.73	1.7	1897.46	1.7
	4k	1017.19	3.1	3451.76	3.1
	8k	2041.87	6.3	6903.21	6.3

A partir da Tabela 2 é possível perceber-se que uma *tag* **h3**, por exemplo, gasta  $8.85ms$  para ser renderizada e consome  $26.31mJ$  de energia para ser carregada. Como outro exemplo tem-se a *tag* **table** que gasta  $20.96ms$  para ser renderizada e consome  $69.82mJ$ .

Outras estratégias para redução do consumo energético envolvem a caracterização energética dos dispositivos a fim de verificar a contribuição energética deles no consumo total do *smartphone*. Mora *et al.* (MORA; LEU, 2012) primeiramente realizam um estudo aprofundado e desenvolvem uma metodologia para caracterizar o consumo energético dos diferentes módulos internos do *smartphone*. Em seguida, um exame sobre as tendências de uso dos usuários é mostrado e, finalmente, eles propõem uma estratégia para economia de energia. Esses autores dispõem-se a criar uma metodologia de medição energética que consiste em atingir os seguintes objetivos:

1. estabelecer um método de medição de potência não intrusivo e preciso;
2. implementar uma medição de potência baseada em *script* a fim de diminuir o ruído aleatório;
3. caracterizar os seguintes módulos do telefone: estado ocioso do telefone, telefone ativo (sem rádio ligado), tela (brilho mínimo e máximo), ligação telefônica, conexão 3G, conexão *WiFi*, *SDCard* e *Player Multimídia*.

Mora *et al.* (MORA; LEU, 2012) observaram que o que mais consome energia são: *WiFi*, ligação telefônica, rádio 3G e tela. Como, segundo esses autores, não há muito a

ser feito para melhorar o algoritmo de eficiência energética para tela (redução do brilho), eles decidiram analisar o comportamento do usuário, coletando as ligações telefônicas e os acessos à rede. De posse das informações de consumo e de uso, esses autores decidiram criar uma técnica de economia de energia que administra os dois módulos que mais consomem energia: 3G/rádio celular e *WiFi*, e conexão com a rede, e escolheram o módulo que menos consome energia de acordo com as possibilidades. Com a técnica desenvolvida, eles rejeitaram em média 29% das tentativas de conexão, pois, de acordo com a técnica deles, essas conexões em demasia são fonte de muito consumo. Eles conseguem atingir pelo menos 8h de uso.

Já Sklavos *et al.* (SKLAVOS; TOULIOU, 2010) se propõem a fazer uma análise do consumo energético dos dispositivos móveis, apresentando uma arquitetura para esses dispositivos e posteriormente analisando quanto de energia cada um dos componentes da referida arquitetura consome. Após isso, eles fizeram uma análise das técnicas de gerenciamento de energia utilizadas nos dispositivos móveis e discutiram avanços futuros.

De acordo com esses autores, a distribuição de consumo energético de um *smartphone* se dá da seguinte maneira: as aplicações são responsáveis por 39,5%; o módulo de RF, por 21,5%; o subsistema de memória, por 19,4% do total do consumo; o *display*, por 17,6% do total do consumo energético do celular; e os outros periféricos representam 2% do total de energia consumida nessa plataforma.

As técnicas de gerenciamento de energia mostradas por esses autores são as seguintes:

**Telas adaptativas energeticamente:** De acordo com essa técnica, a qualidade do *display*, a representação de cores, o brilho e o tamanho são ajustados cada vez, de acordo com as preferências da aplicação e as demandas do usuário.

**Memórias de baixo consumo:** Técnicas de economia de energia sugerem que a memória dos celulares seja particionada em pequenos bancos, permitindo que, a cada intervalo de tempo, apenas o banco em uso seja ativado, além de certos avanços na tecnologia de memórias dirigidas para uso em dispositivos movidos a bateria.

**Unidade de gerenciamento de energia:** Para reduzir o consumo global do dispositivo, uma abordagem comumente empregada sugere colocar o dispositivo em execução lenta e parar circuitos de *clock* ou módulos que não estão sendo usados. Está incluso nessa categoria também o uso de técnicas como *DVFS*.

Uma outra abordagem bastante utilizada para otimização do consumo é a diminuição de carga computacional. Khairy *et al.* (KHAIRY; AMMAR; BAHGAT, 2013) desenvolveram uma técnica para diminuição da carga computacional em *smartphones* com base em informações contextuais. Com essa técnica, esses autores pretenderam estender o

ciclo de vida da bateria e ainda assim permitir que o *smartphone* executasse tarefas com intensa carga computacional, mantendo a qualidade do serviço entregue ao usuário. Eles realizaram experimentos em diversos contextos com diferentes aplicações — rede, usuário e configurações de dispositivo — e propuseram-se, por meio desse estudo, a oferecer uma técnica que utiliza SVR(Support Vector Regression) e diversos recursos contextuais para prever o tempo de execução remota e a energia consumida para, então, tomar a decisão de diminuição de carga computacional. Essa decisão é tomada de forma a garantir a otimização do tempo de resposta e da energia consumida. As técnicas de diminuição da carga computacional foram avaliadas com três exemplos de aplicação real.

Para otimizar o tempo de resposta e a energia consumida pela aplicação, Khairy *et al.*(KHAIRY; AMMAR; BAHGAT, 2013) introduziram contexto na técnica de diminuição de carga computacional, considerando as seguintes informações contextuais: características de rede (*WiFi*, *2G*, *3G* etc.); características do serviço (código passível de ser processado fora do dispositivo); características do dispositivo (*clock* do processador, modelo e taxa de transferência da interface de rede); e características do usuário (nível de bateria, data, hora e informação de localização).

O algoritmo proposto vem, então, a ser uma técnica de diminuição da carga computacional baseada em aprendizado supervisionado, enfocando eficiência energética. Assim, o sistema é formado pelos seguintes componentes:

1. um cliente que controla a transferência de dados para os serviços responsáveis pela diminuição da carga computacional;
2. um gerador de perfil que coleta as medições de consumo e tempo de execução alinhados com diferentes informações contextuais;
3. um preditor de consumo local que prevê o consumo energético e o tempo gasto para executar o serviço localmente, baseado no histórico salvo pelo gerador de perfil.

Para realizar os testes com a técnica desenvolvida, esses autores utilizaram as seguintes aplicações: a) Eyedentify, uma aplicação de reconhecimento de objetos a qual possui um serviço de processamento computacional externo que obtém um vetor de recursos de uma imagem para reconhecer; b) um tradutor OCR (Optical Character Recognition), que realiza a tradução das imagens capturadas por meio de um serviço de processamento externo; e c) um jogo de realidade aumentada, em que dois jogadores lutam em um duelo virtual no mundo real. Esse jogo tem um serviço de processamento externo responsável por reconhecer a face do oponente e mapeá-la virtualmente.

Os resultados apontaram uma redução de 40% da energia consumida e 43% do tempo consumido pelas execuções locais no caso do Eyedentify. Para o tradutor OCR, o uso da técnica possibilitou uma economia de 52% da energia consumida e 54% do tempo

gasto pelas execuções locais. Já para o Photoshoot, o uso da técnica economizou 56% da energia consumida e 58% do tempo gasto pelas execuções locais.

Como foi dito, uma das técnicas mais recorrentes nos trabalhos realizados visando à otimização energética de *smartphones* que utilizam o sistema operacional Android é o ajuste dinâmico de tensão e frequência do processador (*DVFS*). Isso se deve ao fato de o Linux, sistema operacional no qual o Android se baseia, já possuir um módulo em *software* para realizar esse ajuste, tornando-se então uma técnica economicamente viável. A fim de entender melhor o funcionamento desse módulo, foi realizado um estudo que está registrado na Seção 2.9.

## 2.9 CPUFreq: Escalonamento de Frequência no *Kernel* Android

O mecanismo de *DVFS* possibilita o chaveamento de frequência que permite a mudança da velocidade de *clock* da CPU em tempo real. Por meio desse mecanismo, o projetista do sistema pode ajustar a velocidade do processador adequadamente, de forma que ele consuma menos potência e a *performance* não seja degradada.

O mecanismo *DVFS* foi implementado desde o *Kernel* Linux 2.6 por meio de um módulo chamado CPUFreq([KERNEL.ORG](http://kernel.org), 2013), e o Android herdou essa característica por ser baseado nessa versão do *Kernel*. Dessa forma, o módulo **CPUFreq** funciona da mesma maneira, tanto no Android quanto no Linux. A arquitetura desse mecanismo é mostrada na Figura 4.

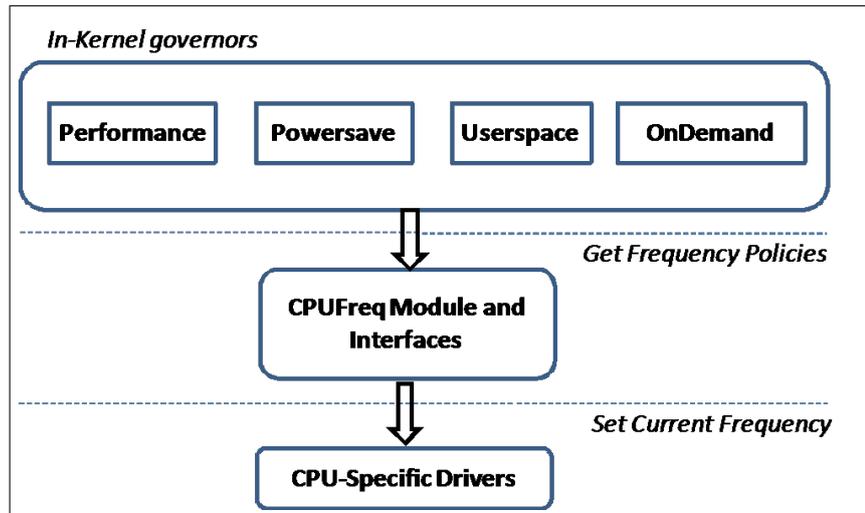


Figura 4 – Arquitetura do CpuFreq

De acordo com a Figura 4, o mecanismo DVFS pode ser dividido em três blocos:

**Governors:** responsáveis por configurar as políticas de ajuste de frequência;

**CPUFreq Module and Interfaces:** responsável por aplicar o mecanismo DVFS de fato;

**CPU Specific Drivers:** responsável pela comunicação entre o módulo CPUFreq e o *hardware*.

Há 5 *governors* padrão no *kernel* e cada um executa o escalonamento de frequência de uma forma que prioriza *performance* ou economia de energia, ou, se o usuário desejar, escala o *clock* do processador acima ou abaixo, de acordo com a demanda. São eles:

**Performance:** configura a frequência da CPU estaticamente para usar a maior frequência disponível dentro dos limites do *hardware*;

**Powersave:** configura a frequência da CPU estaticamente para usar a menor frequência disponível dentro dos limites do *hardware*;

**Userspace:** permite ao usuário ou a qualquer programa do espaço do usuário, executando com o identificador de administrador do sistema, configurar a CPU para uma frequência específica dentro do grupo de frequências disponíveis;

**OnDemand:** configura a frequência da CPU de acordo com o *workload* atual.

Além dos *governors* citados acima, no caso do *Kernel* modificado para Android, existem outros *governors* específicos para explorar as características atuais das CPUs existentes nos *smartphones* que estão sendo atualmente produzidos. No caso do Samsung

Galaxy SIII, existe 1 *governor*, além dos citados acima, chamado Pegasus-q. De acordo com o portal XDA Developers(DEVELPERS, 2012), esse é um *governor* multicore baseado no **OnDemand** e com o *governor* com ligação em tempo real. Os processos em execução na fila estão ativos no *array*, chamado *Run-Queue*, dentro do escalonador do *Kernel*, com seus valores de prioridade ordenados. Para garantir um compartilhamento justo do tempo de CPU entre processos, cada um deles executa por um tempo determinado. Desse modo, esse *governor* ajusta a frequência da CPU, de modo a não somente atender o processo que está em execução no momento, mas também os que estão na fila de processos ativos, esperando uma oportunidade para entrarem em execução.

O módulo CPUFreq, mostrado na Figura 4, representa a principal funcionalidade do mecanismo *DVFS* no *Kernel* Linux. Ele é responsável por mensurar a escala de frequência e prover uma padronização para os *drivers*, que são os elementos de *software* que interagem com o *hardware* para realizarem o chaveamento de frequência. Além disso, ele é responsável por notificar as outras partes do *Kernel* Linux a respeito do estado atual da frequência da CPU(KERNEL.ORG, 2013).

Os *drivers* específicos da CPU, a última parte da Figura 4, realizam a interface do módulo CPUFreq com o *hardware* para fazerem o chaveamento de frequência.

Outro mecanismo bastante discutido e utilizado pelas técnicas de otimização descritas na Seção 2.8 foram os serviços Android que são módulos de *software* especiais capazes de realizar uma tarefa no sistema operacional sem interação e conhecimento do usuário sobre a atuação dos mesmos. Eles se tornam bastante importantes por serem um mecanismo importante dentro do Android capaz de monitorar o funcionamento do sistema operacional e tomar as devidas ações, dependendo do estado interno do sistema. Esse mecanismo será discutido na seção a seguir.

## 2.10 Serviços Android

Para serem atingidos os objetivos descritos na Seção 1.1, precisava-se de um componente de *software* capaz de monitorar e informar a aplicação atualmente em uso, de modo a notificar o mecanismo desenvolvido durante a pesquisa para realizar o chaveamento da frequência adequadamente.

De acordo com a API do Android, o serviço é um componente de *software* que representa o desejo de uma aplicação de realizar uma operação por tempo prolongado, sem interagir com o usuário ou suprir uma necessidade para o funcionamento de outra aplicação(DEVELPERS, 2014).

Um serviço pode ser executado pela aplicação quando ela explicitamente faz a chamada do método *Context.startService()*. Então, o sistema irá obter o serviço, criando-o e chamando o método *onStartCommand(Intent, int, int)* dele. A partir de então, o serviço continuará a executar até que seja feita a chamada do método *Context.stopService()* ou *stopSelf()*(DEVELPERS, 2014).

Há dois modos de operação do serviço:

**START\_STICKY:** usado para serviços que são explicitamente iniciados e parados de acordo com a necessidade;

**START\_NOT\_STICKY** ou **START\_REDELIVER\_INTENT:** usado para serviços que devem ser executados apenas para processar comandos enviados a eles.

O ciclo de vida de um serviço é mostrado na Figura 5.

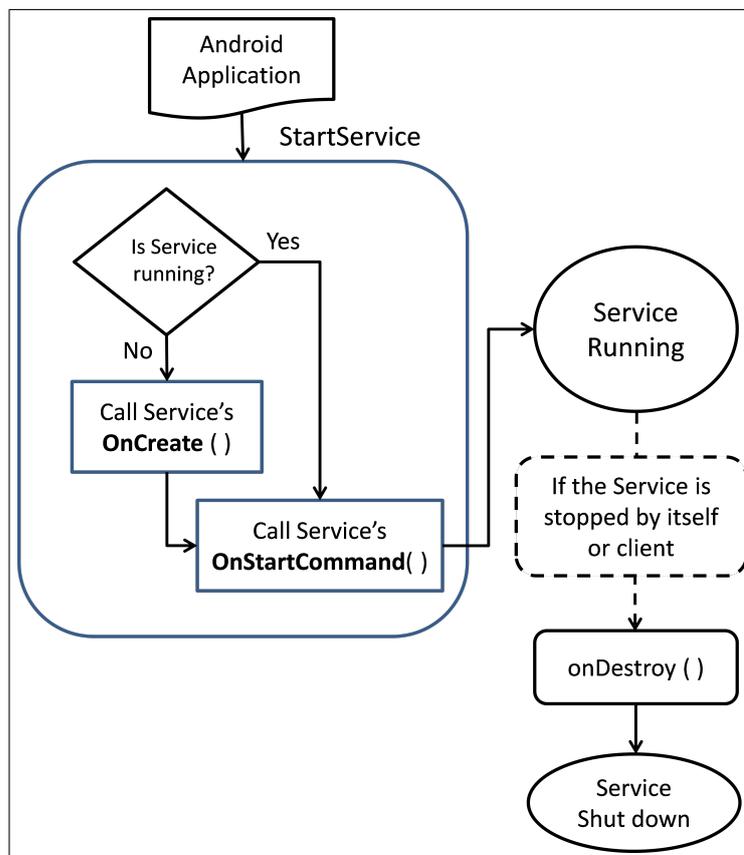


Figura 5 – Ciclo de vida do serviço Android

De acordo com a Figura 5, caso o serviço ainda não tenha sido criado, o método *onCreate()* é chamado para executar os procedimentos necessários para que o serviço entre em execução. Caso o serviço já esteja sendo executado, esse método não é chamado. Após essa chamada, o método *onStartCommand()* é chamado. Ele é que inicia o serviço Android de fato, fazendo-o executar em *background*. Uma vez que o serviço esteja executando, torna-se responsabilidade do desenvolvedor parar o respectivo serviço quando o trabalho do mesmo estiver finalizado. Para realizar essa parada, o desenvolvedor deverá chamar o método *stopService()* que chamará em seguida o método *onDestroy()*, responsável por desalocar todos os recursos do sistema usados durante a execução do serviço em questão e que por hora não são mais necessários.

Após todo o estudo realizado a fim de identificarem-se os mecanismos e métodos utilizados pelos pesquisadores da área de otimização energética para reduzir o consumo de energia de *smartphones* que utilizam o sistema operacional Android, foi possível reconhecer as lacunas existentes nas pesquisas sobre técnicas de medição, supri-las e identificar quais as metodologias mais utilizadas para se obter reduções significativas no consumo desses dispositivos.

De posse das teorias discutidas neste capítulo, foi possível estabelecer uma meto-

dologia de pesquisa para aproveitar as oportunidades de melhoria descritas na Seção 1.2, sendo esse um dos objetivos deste trabalho, conforme discutido na Seção 1.1. A construção dessa metodologia será delineada no próximo capítulo.

## 3 Metodologia

Após realizar-se o levantamento dos estudos mais recentes que têm sido feitos pelos pesquisadores da área de otimização energética utilizando *smartphones* com o sistema operacional Android, encontrar as lacunas nos estudos sobre medição de consumo, conforme foi descrito no Capítulo 2, e verificar as oportunidades de melhorias discutidas na Seção 1.2, resolveu-se empreender uma pesquisa experimental. De acordo com o que foi mostrado na Seção 1.2, esta pesquisa busca mostrar uma **abordagem de economia presumivelmente melhor**, quando comparada às abordagens citadas na literatura.

Dessa forma, percebeu-se que seria necessário investigar como essa abordagem poderia ser construída e recorreu-se à literatura sobre metodologia científica. De acordo com Wazlawick(WAZLAWICK, 2009), o estilo de pesquisa, "Apresentação de algo presumivelmente melhor", exige que qualquer nova abordagem seja comparada quantitativamente com outras da literatura e, na falta de *benchmarks* internacionalmente aceitos, fica a cargo do pesquisador realizar os experimentos para comprovar a melhora.

Por este trabalho se constituir como uma pesquisa experimental, necessitou-se também buscar na literatura referente à metodologia científica os procedimentos e os métodos que poderiam ser utilizados a fim de construir uma pesquisa desse tipo. De acordo com Gil(GIL, 2010), para se delinear uma pesquisa experimental, é necessário desenvolver uma série de passos que podem ser arrolados da seguinte forma:

- a) formulação do problema;
- b) construção das hipóteses;
- c) operacionalização das variáveis;
- d) definição do plano experimental;
- e) determinação dos sujeitos;
- f) determinação do ambiente;
- g) coleta de dados;
- h) análise e interpretação dos dados;
- i) apresentação das conclusões.

Conforme esse delineamento feito por Gil(GIL, 2010), os passos descritos por ele foram cumpridos da seguinte forma neste trabalho:

**Formulação do problema:** A formulação do problema havia sido feita no momento em que se iniciou o desenvolvimento deste trabalho, pois, nessa etapa, foi identificado o alto consumo energético por parte dos *smartphones*, conforme foi discutido no Capítulo 1.

**Construção das hipóteses:** As hipóteses foram construídas quando se verificou que uma menor quantidade de chaveamentos levava a uma menor potência dissipada, conforme foi discutido na Seção 1.2.

**Operacionalização das variáveis:** Para seguir esse passo, precisou-se entender os fatores de um maior consumo energético por parte do *smartphone*. Para isso, fez-se uma investigação e chegou-se à conclusão de que os elementos que mais contribuem para tal são: *display*, processador, interface de rede **Wi-Fi**, interface de rede **Bluetooth** e interface de comunicação de rádio. Neste trabalho, os elementos manipulados foram o processador, a interface de rede **Wi-Fi** e a interface de comunicação de rádio. Logo, as variáveis do estudo foram esses elementos.

**Definição do plano experimental:** Esse passo será explicitado neste capítulo.

**Determinação dos sujeitos:** Esse passo foi estabelecido quando se decidiu estudar o consumo energético de um *smartphone* com um sistema operacional Android.

**Determinação do ambiente:** Nesta etapa, foram selecionados os elementos dos quais se iria precisar a fim de conduzir os experimentos de forma controlada para que se pudesse determinar com maior acurácia a influência de cada uma das variáveis em estudo ao longo do trabalho. O ambiente experimental está descrito na Seção 4.1.

**Coleta de dados:** A coleta de dados foi feita conforme está descrito na Seção 3.1.1.

**Análise e interpretação dos resultados:** Conforme foi discutido no Capítulo 2, os estudos sobre medição de consumo apresentam uma lacuna quanto ao método utilizado. Para suprir essa lacuna, durante a realização deste trabalho, foram escolhidos métodos estatísticos e numéricos para realizar a análise e interpretar os resultados obtidos a partir delas. Esses procedimentos estão descritos na Seção 3.1.3 e na Seção 3.1.4.

**Apresentação das conclusões:** Após realizar os experimentos e análises, chegou-se aos resultados discutidos no Capítulo 4.

Neste capítulo, serão descritos o plano experimental e os procedimentos de coleta e análise de dados, como foi registrado anteriormente. Na próxima seção, será delineado o plano experimental.

## 3.1 Plano Experimental

O plano experimental desenvolvido para a realização deste trabalho está descrito pela Figura 6.

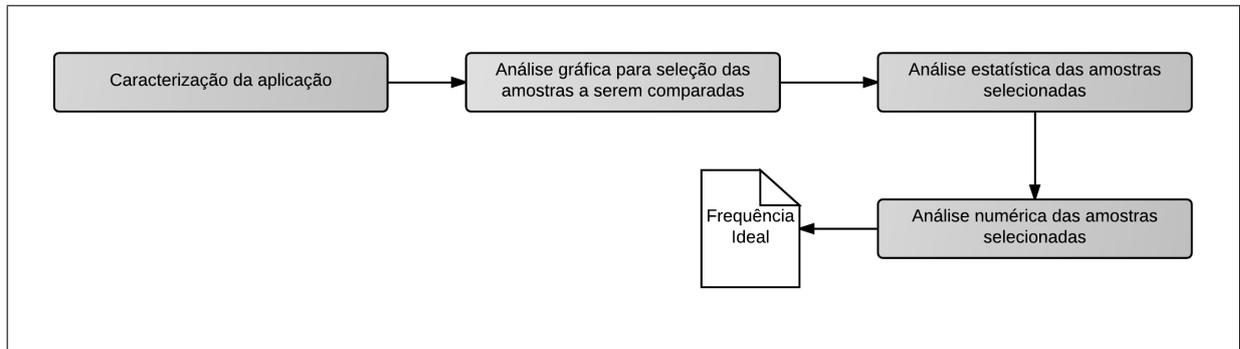


Figura 6 – Plano experimental desenvolvido para a realização da pesquisa

Na etapa de **Caracterização da aplicação**, foram realizadas as coletas de potência instantânea do *Smartphone*. Foram feitas, nesta etapa, 3 coletas de potência para cada uma das 13 frequências estudadas.

Na etapa seguinte (**Análise gráfica para seleção das amostras a serem comparadas**), as 3 coletas de potência para cada uma das 13 frequências estudadas foram comparadas para que fosse selecionada uma amostra de cada uma das frequências estudadas.

Na etapa de **Análise estatística das amostras selecionadas**, foi realizada uma análise estatística a fim de reunir evidências que apontassem que frequências de operação do processador levariam a um menor consumo em cada uma das situações estudadas durante a realização dos experimentos. Nessa etapa, as análises buscavam responder às seguintes perguntas:

1. se as amostras possuíam distribuição normal;
2. se havia diferença estatística da potência instantânea desses dispositivos nas diversas frequências de operação em que o processador dele pode operar.

Dessa forma, à medida que essas duas perguntas foram sendo respondidas, foram levantados indícios de possíveis frequências ideais de operação do dispositivo.

Na última etapa (**Análise numérica das amostras selecionadas**), calculou-se o consumo energético das frequências selecionadas na etapa anterior para se chegar a uma conclusão a respeito da frequência ótima de operação do dispositivo para a execução da tarefa que estava sendo estudada.

Seguindo-se esse plano, cumpria-se a parte do objetivo referente à definição de frequência ideal de operação de uma aplicação, no entanto precisava-se definir uma forma

de classificar essas aplicações que estavam sendo estudadas e assim não chavear a frequência de operação do dispositivo a cada instante em que houvesse uma troca de aplicações, conforme descrito na Seção 1.2. Para cumprir-se esse outro objetivo, adotou-se a ideia de **grupo** (Definição 1).

**Definição 1.** *Um grupo de aplicações é definido a partir da classe à qual a aplicação que está sendo estudada pertence. Dessa forma, foram elencados 3 grupos de aplicações: Stream de Vídeo, Navegação GPS e Navegação Web.*

O grupos de *Stream* de vídeo e Navegação GPS foram selecionados para a realização de experimentos porque, de acordo com Nielsen (NIELSEN, 2011), o aplicativo *Google Maps* responde por 70%-78% de uso, dependendo da faixa de idade do entrevistado, e o aplicativo *Youtube* responde por 51%-64%, dependendo da faixa de idade do entrevistado. Portanto, sentiu-se que esses aplicativos são bastante importantes e merecem investigação.

O grupo de Navegação Web foi selecionado para a realização de experimentos porque, de acordo com o Super Monitoring (SUPER!MONITORING, 2013), os usuários de *smartphone* gastam 18% de seu tempo com navegação Web. Apesar de esse mesmo instituto indicar que esses mesmos usuários gastam 32% de seu tempo com games, essa categoria não foi considerada por possuir requisitos bastantes distintos dependendo do jogo escolhido, o que a torna difícil para padronizar os experimentos.

Outro conceito que é importante ser definido é o de aplicativo, pois o plano experimental traçado neste trabalho é baseado na caracterização e definição de uma frequência ideal de operação para um grupo de aplicativos que foram classificados nos **grupos** acima definidos. Dessa forma, definiu-se um aplicativo conforme a Definição 2.

**Definição 2.** *Um aplicativo é uma atividade que o dispositivo móvel executa visando atender às necessidades do usuário em um determinado momento de operação do dispositivo. Como exemplo de aplicativos estudados, têm-se: Google Maps, Youtube e Dailymotion. Para simplificar, passa-se agora a se fazer referência a um aplicativo usando-se a sigla **App**.*

Passa-se, agora, a delinear os passos dados na elaboração do Plano Experimental, conforme se registrou na Figura 6.

### 3.1.1 Etapa 1: Caracterização da aplicação

Nesta etapa são coletadas três amostras da potência instantânea dissipada pelo *smartphone* na execução da **App** e na condição de execução que está sendo caracterizada em cada uma de  $N$  frequências de operação do dispositivo móvel.

Para realizar-se a coleta de dados, foi elaborado um circuito, como se pode ver na Figura 7.

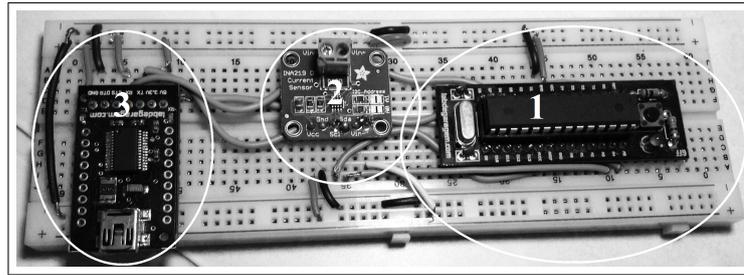


Figura 7 – Circuito de Medição

Esse circuito é composto principalmente de três componentes:

**Atmega328p:** O circuito indicado por **1** na figura acima é o microcontrolador da Atmel de 8 bits com 28 pinos, sendo 19 pinos de *E/S* (ATMEL, 2009). Ele é o responsável por interagir com o circuito de medição indicado por **2**, processar os dados e submetê-los ao computador através do circuito conversor USB->Serial, indicado na figura pelo número **3**.

**INA 219:** O circuito indicado na figura por **2** é o circuito de monitoramento de corrente e potência produzido pela *Texas Instruments* (TEXAS, 2011), responsável por medir a tensão e corrente instantâneas do celular e transmitir para o Atmega328p a fim de que o dado possa ser enviado ao computador e possa, assim, ser registrado.

**Conversor USB->Serial:** O circuito indicado na figura pelo número **3** é um circuito capaz de converter os sinais *Tx* e *Rx* para **USB** a fim de que possa haver a comunicação do microcontrolador com computadores mais modernos.

O procedimento de coleta foi realizado conforme mostrado na Figura 8. Ele foi dividido em 7 etapas, sendo as 2 últimas repetidas para cada nova coleta.

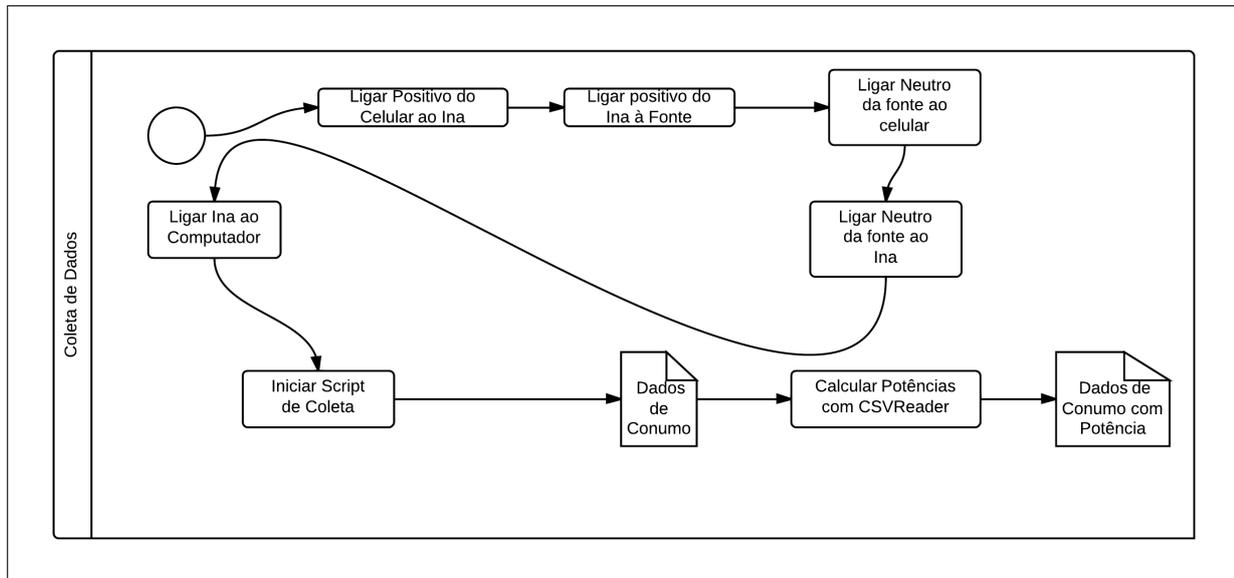


Figura 8 – Procedimento para coleta de dados

As 7 etapas da coleta são as seguintes:

**Ligar positivo do celular ao Ina:** Nessa etapa, liga-se o polo positivo da bateria do celular ao circuito de medição, de modo que este último fique ligado em série com a fonte que irá fornecer energia para o celular.

**Ligar positivo do Ina à fonte:** Nessa etapa, conclui-se a ligação em série do circuito de medição com o *smartphone* e a fonte.

**Ligar neutro da fonte ao celular:** Nessa etapa, conclui-se a ligação do polo neutro da bateria do celular com a fonte, de modo que o mesmo possa fechar o circuito e, assim, possa funcionar como se estivesse sendo alimentado pela bateria.

**Ligar neutro da fonte ao Ina:** Nessa etapa, unificam-se os neutros do circuito de medição e do *smartphone*, de modo que se possa eliminar qualquer ruído e o Ina possa realizar a medição da tensão à qual o celular está submetido.

**Ligar Ina ao computador:** Nessa etapa, o circuito de medição é ligado ao computador por meio do circuito conversor *Serial->USB*.

**Iniciar script de coleta:** Essa é a etapa que marca o início do processo de coleta de dados. Nela, executa-se um *Script Shell* que abre a conexão do computador com o circuito conversor *Serial->USB* e, assim, inicia o microcontrolador, que começa a enviar dados sobre a potência instantânea do celular em tempo real a uma taxa aproximada de 271 amostras por segundo, gerando um arquivo com os dados de consumo de tensão e corrente instantâneas do *smartphone*. O arquivo gerado pelo *script* de coleta tem o formato mostrado na Tabela 3.

Tabela 3 – Layout do arquivo de coleta

Bus Voltage(V)	Shunt Voltage(V)	Load Voltage(V)	Current(mA)
4.14	44.11	4.18	595.12
4.15	46.68	4.20	500.60
4.16	44.64	4.20	431.52
4.17	37.66	4.21	415.80

O circuito utilizado para a coleta de dados utiliza um resistor *shunt* para realizar a medição da corrente requerida pelo dispositivo através da queda de tensão nesse resistor. Dessa forma, no *layout* do arquivo de coleta, a coluna **Bus Voltage** informa a tensão no barramento após a queda sofrida devido ao resistor. Por sua vez, a coluna **Shunt Voltage** informa sobre a quantidade de queda de tensão devido ao resistor, enquanto a coluna **Load Voltage** informa sobre a tensão de entrada no circuito; logo, antes da queda devido ao resistor. Já a coluna **Current** informa sobre a corrente requerida pelo dispositivo em um determinado instante de amostragem.

**Calcular potências com CSVReader:** Nessa etapa, o arquivo gerado na etapa anterior é tratado para que se possa calcular a potência instantânea a partir dos dados de tensão, fornecidos pela coluna **Bus Voltage**, e a corrente, fornecida pela coluna **Current**, usando-se a equação (3.1) e gerando-se, assim, o arquivo que será utilizado nas análises estatística e numérica, conforme mostrado na Tabela 4.

$$P_i = U_i \times \frac{I_i}{1000} \quad (3.1)$$

Tabela 4 – Layout do arquivo de coleta com potência

Bus Voltage(V)	Shunt Voltage(V)	Load Voltage(V)	Current(mA)	Power(W)
4.14	44.11	4.18	595.12	2.464
4.15	46.68	4.20	500.60	2.077
4.16	44.64	4.20	431.52	1.795
4.17	37.66	4.21	415.80	1.734

### 3.1.2 Etapa 2: Análise gráfica para seleção das amostras a serem comparadas

Nesta etapa, após terem sido realizadas as coletas na etapa anterior, plotaram-se os gráficos de consumo para averiguar qual das três amostras apresentava maior estabilidade nas  $N$  frequências investigadas. O resultado obtido pode ser visto na Figura 9.

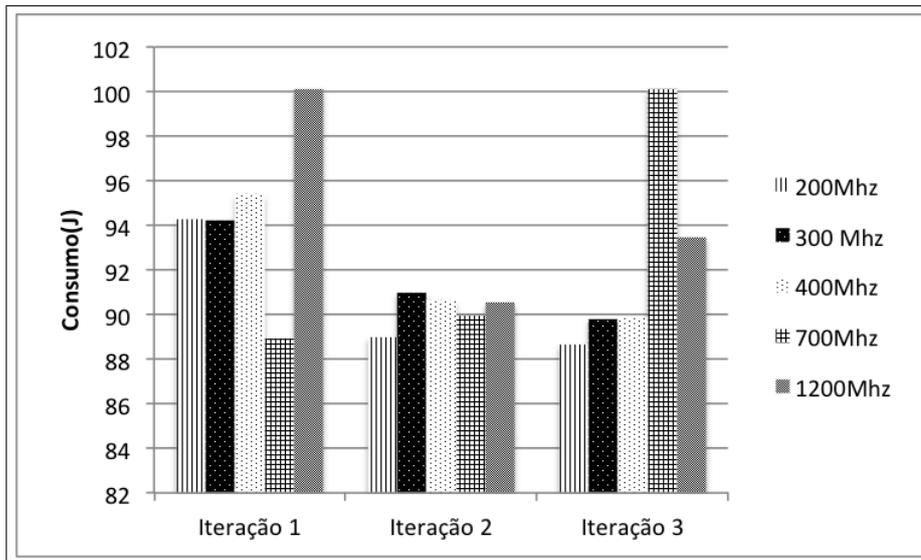


Figura 9 – Comparação das iterações durante análise gráfica

Como é possível perceber pela Figura 9, a iteração 2 de cada uma das frequências mostradas se mostra mais estável para ser comparada entre frequências distintas. Esse fato é comprovado pela proximidade dos valores de consumo, conforme mostrado bem claramente na Figura 9.

A estabilidade da iteração 2 se deve ao fato de que o processo de inicialização e finalização do *script* de coleta contém naturalmente muito ruído, pois, nesse momento, há a abertura e fechamento da conexão com o circuito, o que implica ligar e desligar o circuito de medição.

Ao ligar, o circuito de medição leva um tempo até estabilizar a tensão nos seus diversos componentes, inclusive no microcontrolador **Atmega**, responsável por realizar as leituras no **INA219**, convertê-las e enviá-las ao computador.

Ao desligar o circuito, também há muito ruído, pois há o encerramento das *threads* de monitoramento, o que leva fatalmente a coletas discrepantes e, assim, a um aumento substancial dos valores medidos.

Diante desse resultado, escolheu-se, então, a iteração de número 2, pois se entendeu que ela representa um consumo mais fidedigno do *smartphone* para a realização da atividade em estudo nas condições às quais o dispositivo está submetido.

### 3.1.3 Etapa 3:Análise estatística das amostras selecionadas

Após realizar-se a comparação gráfica, conforme descrito na Etapa 3.1.2, e selecionar-se a iteração 2 de cada uma das frequências estudadas(ver os motivos acima mencionados), essas amostras foram colocadas em um *roll* de amostras a serem comparadas estatisticamente, conforme será descrito nesta etapa.

A fim de empreender-se a análise estatística, adotou-se um procedimento composto de 7 etapas e foram usadas algumas teorias estatísticas para a realização de testes de aderência à normal e verificação se duas amostras provinham da mesma distribuição de probabilidade, ou não. As etapas são mostradas na Figura 10 e podem ser descritas da seguinte forma:

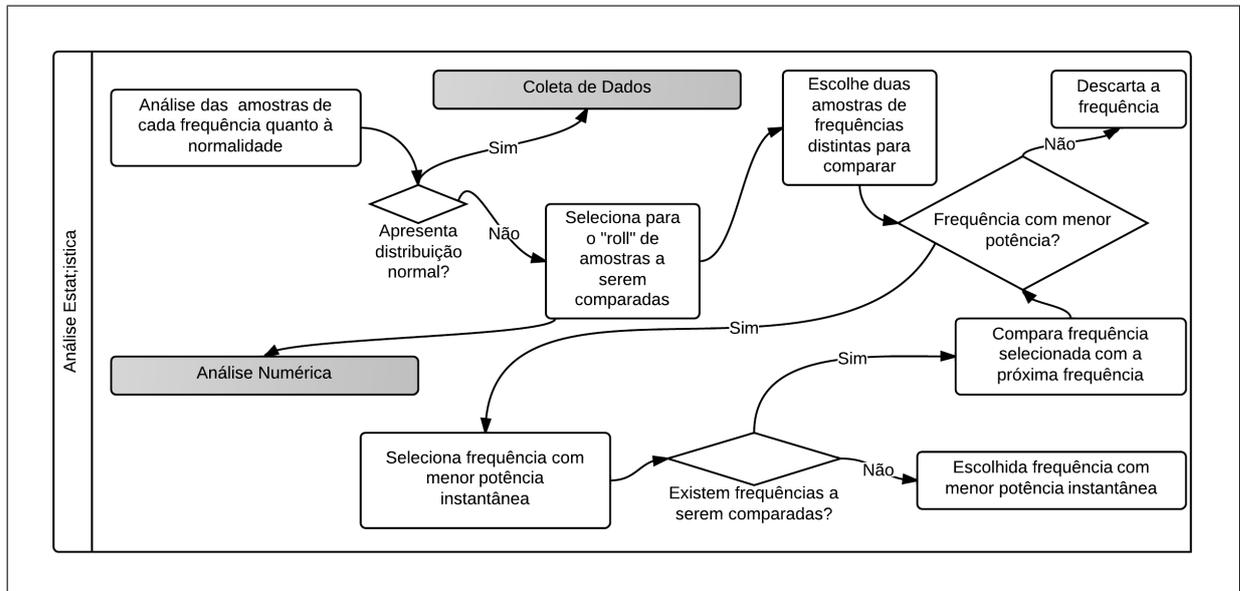


Figura 10 – Procedimento para realização da análise estatística

**Análise das amostras de cada frequência quanto à normalidade:** A fim de escolher-se o teste estatístico adequado para realizar a comparação entre as amostras selecionadas na etapa anterior, foi necessário verificar se elas aderiam a uma distribuição normal. Na literatura há vários tipos de teste de aderência, como foi discutido na Seção 2.4. De acordo com essa discussão, não se poderia utilizar o **Teste Binomial**, pois, no caso em estudo, há mais de duas classes nas quais podem ser enquadradas as diversas amostras. Já o **Teste Qui-Quadrado**, conforme foi salientado na Seção 2.4, possui a limitação de não poder mensurar o "poder do teste" com facilidade quando se têm dados nominais ou categorizados, ou quando os dados consistem em frequências em categorias inerentemente discretas. No caso em estudo, as séries de potência observadas se encaixam em categorias inerentemente discretas, uma vez que as faixas de frequências são discretas. Dessa forma, recorreu-se ao **Teste de Lilliefors** por este atender melhor às necessidades deste trabalho. Esse teste é um caso particular do **Teste de Aderência de Kolmogorov Smirnov** em que a distribuição hipotética é estabelecida como a normal. Na Seção 2.5 e na Seção 2.4, pode-se encontrar mais detalhes sobre a teoria desse teste e sobre testes de aderência de modo geral. A estatística do teste é definida pela equação (3.2) e pela equação (3.3).

$$D = \max\{D^+, D^-\} \quad (3.2)$$

$$D^+ = \max_{i=1,\dots,n} \left\{ \frac{i}{n} - P(i) \right\}, D^- = \max_{i=1,\dots,n} \left\{ P(i) - \frac{(i-1)}{n} \right\} \quad (3.3)$$

Um exemplo de resultado do **Teste de Lilliefors** pode ser visto na Tabela 5.

Tabela 5 – Exemplo do Resultado do Teste Lilliefors para as séries de potência do Youtube

Freq(MHz)	D
200	0.0538
300	0.0609
400	0.0520
500	0.0590
600	0.0555
700	0.0461
800	0.0505
900	0.0540
1000	0.0469
1100	0.0501
1200	0.0453
1300	0.0563
1400	0.0582

De acordo com esses resultados e a teoria vista na Seção 2.5, como todas as amostras coletadas durante a realização deste trabalho possuem mais de 35 exemplos, o valor crítico da estatística  $D$  pode ser calculado por:  $D^* = \frac{1,36}{\sqrt{N}}$ , que, substituído pela quantidade de amostras coletadas para esse exemplo, 15.377 em média, tem-se  $D^* = \frac{1,36}{\sqrt{15377}} = 0,01097$ . Como todos os valores da estatística mostrados na Tabela 5 possuem valores maiores que o valor limite, as distribuições das quais as amostras foram retiradas não podem ser aproximadas para uma distribuição normal, conforme foi discutido na Seção 2.5.

**Seleção do rol de amostras a serem comparadas:** Após serem realizados diversos experimentos, notou-se que nenhuma das amostras podia ser aproximada para uma distribuição normal, de modo que se pudesse usar um teste estatístico paramétrico para comparar se duas amostras de frequências distintas seriam iguais ou uma delas seria maior que a outra. Desse modo, após se constatar que a amostra não pode ser aproximada para uma distribuição normal, ela é selecionada para compor o rol de amostras a serem comparadas na escolha da frequência ideal. Caso o resultado do teste realizado na etapa anterior constatasse que essa aproximação poderia ser feita, inferia-se que havia ocorrido um erro na etapa de coleta de dados. Desse modo, os experimentos relativos à frequência em que a amostra apresentou distribuição normal eram repetidos.

**Escolha de duas amostras de frequências distintas para serem comparadas:** Nessa etapa, são escolhidas duas amostras de frequências distintas a serem comparadas a fim de encontrar-se a frequência que possui as menores potências instantâneas

e encontrarem-se as tendências de economia. Para realizar-se essa etapa, havia dois tipos de teste a serem feitos: o **Teste T de Student**, um teste paramétrico que utiliza média e desvio padrão para realizar a comparação entre duas amostras e afirmar se elas provêm, ou não, da mesma distribuição de probabilidade e, caso não provenham, qual das duas é maior que a outra; e o **Teste Wilcoxon-Mann-Whitney**, um teste estatístico que se utiliza de mensurações ordinais, conforme discutido na Seção 2.6, para testar se dois grupos independentes foram extraídos da mesma população. Esse teste utiliza estatística não paramétrica para identificar se as duas amostras provêm, ou não, da mesma distribuição de probabilidade e, caso isso não aconteça, qual das duas é maior ou menor que a outra. Escolheu-se utilizar um teste não paramétrico porque, conforme o resultado obtido no passo anterior, as amostras não podem ser aproximadas para uma distribuição normal e o **Teste T de Student** requer essa aproximação para ser usado. A estatística do teste é definida pela equação (3.4).

$$Z = \frac{W_x \pm 0,5 - \mu_{w_x}}{\sigma_{w_x}} = \frac{W_x \pm 0,5 - m \times \frac{(N+1)}{2}}{\sqrt{m \times n \times \frac{(N+1)}{12}}} \quad (3.4)$$

**Descarte da frequência:** Nessa etapa, a frequência que, porventura, for maior que a outra com a qual ela estiver sendo comparada é descartada para uso em futuras comparações com as frequências restantes, voltando-se, assim, ao rol de frequências escolhidas na etapa 2 do procedimento de análise para ser usado na fase de análise numérica.

**Seleção da frequência com menor potência instantânea:** Nessa etapa, após a comparação feita no passo 3 do processo, a frequência que apresentou menores valores de potência instantânea, de acordo com o *ranking* montado pelo **Teste de Wilcoxon-Mann-Whitney** descrito na Seção 2.6, é selecionada para ser comparada com as outras amostras de frequência ainda não testadas, caso existam.

**Comparação da frequência selecionada com a próxima frequência:** Nessa etapa, o procedimento realizado no passo 3 do processo é repetido para compararem-se as amostras ainda não submetidas ao **Teste de Wilcoxon-Mann-Whitney** com a amostra da frequência que apresenta menores valores de potência instantânea em relação às amostras já comparadas anteriormente. Caso uma nova amostra de frequência apresente menores valores de potências instantâneas, ela será selecionada para ser comparada com as outras amostras ainda não comparadas; caso contrário, a amostra tida como a menor permanece.

**Escolha da frequência com menor potência instantânea:** Nessa etapa, o procedimento adotado para a identificação da frequência que apresenta menores valores de frequência instantânea termina e, assim, já se tem indício da melhor frequência a

ser adotada para a realização da atividade em questão: exibição de vídeo, navegação GPS, ou navegação Web.

### 3.1.4 Etapa 4: Análise numérica das amostras selecionadas

Nessa etapa, cada uma das amostras selecionada na etapa anterior é submetida a uma análise numérica a fim de encontrar-se a frequência de operação do dispositivo que consome a menor quantidade de energia na execução da **App** em questão e, assim, estabelecer-se o que se chama de frequência ideal de operação para a aplicação em questão.

Para a realização da análise numérica, adotou-se um procedimento em que todas as amostras selecionadas no rol de amostras válidas da análise estatística, realizada na etapa anterior, são submetidas a uma nova análise para serem identificadas possíveis discrepâncias no tempo de execução, pois se sabe que o consumo de um determinado equipamento é dado pela equação (3.5) a qual define que a energia consumida é igual à integral da potência dissipada instantânea  $P_i$  no tempo  $t$ ; logo, faz-se necessário analisar também o tempo além da potência instantânea. Isso porque, dependendo do tempo, pode-se ter um maior consumo, mesmo com uma menor potência instantânea. Além disso, o processo de otimização proposto, conforme descrito na Seção 1.1, visa otimizar o consumo energético do *smartphone* sem diminuir o desempenho, de modo que a interação com usuário não fique prejudicada.

$$E = \int P_i(t) dt. \quad (3.5)$$

Para a realização do procedimento de análise numérica, foram percorridos **5** passos a fim de se analisar o valor numérico do consumo em cada uma das frequências analisadas e, dessa forma, terem-se subsídios para indicar a frequência que possui uma maior tendência de realizar economia.

Esses procedimentos são mostrados na Figura 11 e podem ser descritos da seguinte forma:

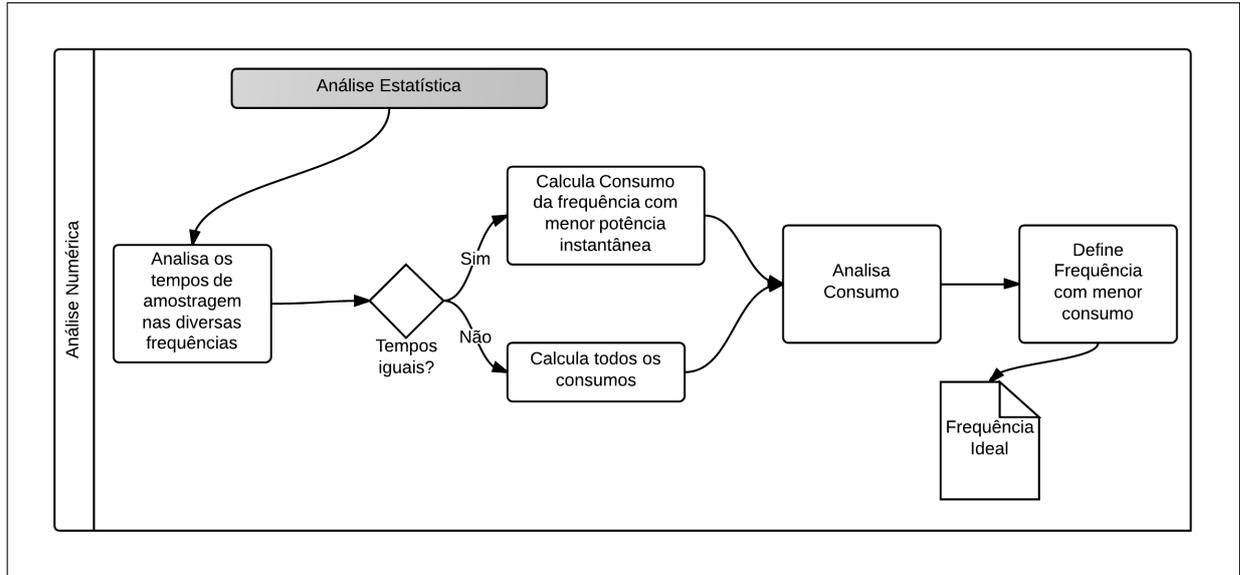


Figura 11 – Procedimentos para realização da análise numérica

**Analisa os tempos de amostragem nas diversas frequências:** Nesse momento, são analisados os tempos de amostragem para verificar-se se eles diferem e, assim, poder-se determinar se a frequência ideal encontrada durante a realização da análise estatística é de fato a frequência ótima de operação do dispositivo para a realização da atividade em questão.

**Calcula o consumo da frequência com menor potência instantânea:** Caso o resultado da análise feita na etapa anterior informe que os tempos de amostragem são iguais, o consumo da amostra com menor potência instantânea tenderá a ser o menor consumo, confirmando, assim, o resultado encontrado na análise estatística. O cálculo do consumo é feito utilizando-se a equação (3.6) que, ao desenvolver-se, tem-se a equação (3.7), conforme foi discutido na Seção 2.7.

$$I = \int_a^b f(x) dx \approx \int_a^b f_1(x) dx \quad (3.6)$$

$$I = (b - a) \times \frac{f(b) + f(a)}{2} \quad (3.7)$$

**Calcula todos os consumos:** Caso o resultado da análise feita na etapa 1 informe que os tempos de amostragem nas diferentes frequências de operação são diferentes, faz-se necessário realizar o cálculo do consumo das amostras de todas as frequências analisadas para se encontrar uma frequência que apresente maior tendência para consumir menos bateria durante a realização da atividade estudada. O cálculo do consumo é feito usando-se a equação (3.6) que, ao desenvolver-se, tem-se a equação (3.7), conforme discutido no passo anterior.

**Analisa consumo:** Após terem sido obtidos os resultados de consumo aproximado, faz-se uma análise do consumo energético em comparação com os outros *governors* do módulo de ajuste de frequência (CPUFreq) presente no *Kernel* Android a fim de apontarem-se as vantagens de adotar a frequência ótima de operação encontrada nas etapas anteriores.

**Define a frequência com menor consumo:** Esse é o último passo do processo de análise adotada. Nessa etapa, define-se a frequência ideal de operação do dispositivo na realização da atividade estudada, visando-se à redução do consumo energético do *smartphone*.

Nesse procedimento, foram utilizadas as ideias de integração propostas por Lima (LIMA, 1992), com a ressalva de que não se poderia aplicar completamente a teoria defendida por esse autor, pois não se tem nem a primeira, nem a segunda derivadas da função. Embora não se tenha como se obter o erro associado ao cálculo da área sob o gráfico da série, tem-se um cálculo com resultados bastante estáveis. Foram, então, utilizadas as ideias discutidas na Seção 2.7.

Observou-se que, ao aumentar a quantidade de amostras na unidade do tempo, o cálculo do consumo tendeu a uma estabilidade da ordem de casas decimais, indicando, assim, que havia uma tendência a encontrar-se uma forma de obter-se a área sob o gráfico, pois não se tinha o modelo da função e conseqüentemente não se podia utilizar os métodos comuns de integração numérica por não se ter como obter o erro associado. Dessa forma, optou-se por usar o método de integração numérica dos trapézios, conforme foi discutido na Seção 2.7.

Nesse momento, também se fez uma análise para definir quais das frequências selecionadas a partir da **Análise Estatística** para compor o *roll* de amostras a serem analisadas possuem consumo próximo ao da frequência ideal para uma futura comparação com as outras aplicações (**Apps**) do mesmo **Grupo**. Esses passos foram repetidos em todas as  $K$  aplicações analisadas durante a realização deste trabalho.

Após finalizar as análises dos **Apps** para cada um dos **Grupos** estudados, viu-se a necessidade de montar um resumo das frequências ideais e aceitáveis para cada **App** que compõe o **Grupo** em estudo. A estrutura desse resumo é mostrada e discutida na seção a seguir.

## 3.2 Resumo das aplicações analisadas para um grupo de aplicações

Depois de terem sido analisadas as aplicações selecionadas para compor um determinado **Grupo**, foi montado um resumo em que foram listadas as aplicações analisadas,

as frequências ideais e as frequências aceitáveis para cada uma delas. Um exemplo genérico de um resumo pode ser visto a seguir.

- App1(Wi-Fi)
  - Frequência ideal: 700 *Mhz*
  - Frequências aceitáveis: 200 *Mhz*, 300 *Mhz*, 400 *Mhz*, 1.2 *Ghz*
  
- App1(3G)
  - Frequência ideal: 300 *Mhz*
  - Frequências aceitáveis: 200 *Mhz*, 800 *Mhz*
  
- App2(Wi-Fi)
  - Frequência ideal: 200 *Mhz*
  - Frequências aceitáveis: 300 *Mhz*, 600 *Mhz*, 700 *Mhz*, 1 *Ghz*
  
- App2(3G)
  - Frequência ideal: 500 *Mhz*
  - Frequências aceitáveis: 400 *Mhz*, 600 *Mhz*, 700 *Mhz*, 800 *Mhz*, 1.0 *Ghz*

A partir de um resumo como o mostrado acima, foi possível decidir a frequência ideal para execução das aplicações do **Grupo** em questão na **condição de execução** apropriada; para o caso acima, a condição de execução se diferencia pelo tipo de conexão utilizado. A frequência ideal, de acordo com as análises, seria de 800 *Mhz* para esse **grupo** genérico na 3G e de 700 *Mhz* na Wi-Fi.

Está portanto, descrita a metodologia adotada por este trabalho que pretende apontar como se pode realizar a redução do consumo de energia pelas aplicações utilizando-se uma técnica mais inteligente para chaveamento da frequência, atingindo, dessa forma, o objetivo estabelecido: desenvolver um mecanismo de otimização do consumo de energia para dispositivos móveis baseado em **grupos** de aplicações, conforme mencionado na Seção 1.1.

Assim, na seção a seguir, será mostrada a interação entre os procedimentos de análise a fim de possibilitar um maior entendimento sobre como foi realizada a análise dos dados coletados para a realização deste trabalho, conforme mostrado na Figura 12.

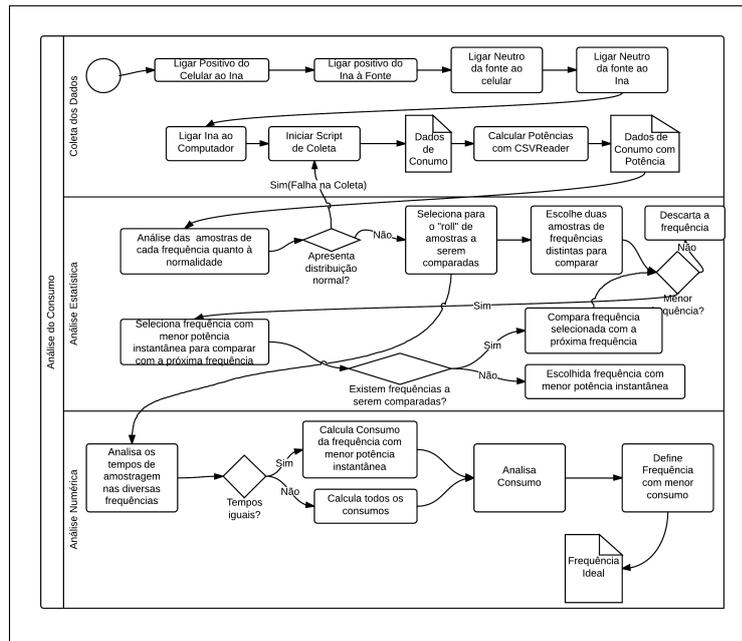


Figura 12 – Interação entre os procedimentos de análise

### 3.3 Interação entre os procedimentos de análise

Na Figura 12 pode-se perceber como o procedimento de coleta de dados interage com a análise estatística, fornecendo os dados coletados para serem analisados e mostrando-os novamente, caso uma amostra apresente aproximação a uma distribuição normal. Como se trata de uma série temporal com características que aparentam um movimento Browniano, pois a mesma não possui tendência nem sazonalidade, é realizada uma nova coleta de amostra na frequência em que foi constatado o problema. O fato de usar-se o **Teste de Wilcoxon-Mann-Whitney** se dá unicamente por conta de as amostras não poderem ser aproximadas a uma distribuição normal, o que impossibilita o uso de outro teste bastante utilizado para fazer esse tipo de comparação, o **Teste T de Student**.

Pode-se também perceber como o procedimento de análise estatística interage com o da análise numérica, pois a primeira é responsável por selecionar as amostras que serão posteriormente analisadas pela última. A análise estatística fica responsável, então, pela definição da frequência com menor potência instantânea que, dependendo do tempo para a execução do **App**, indicará a frequência que possui menor consumo energético. Já a análise numérica tem dois grandes papéis na definição deste trabalho: (a) corroborar a análise estatística, mostrando uma convergência de resultado por meio de métodos e técnicas diferentes e embasando esta pesquisa; e (b) apontar um mecanismo que seja capaz de calcular o consumo energético caso os tempos de execução do **App** variem, sinalizando dessa forma a frequência que possui maior tendência de economia.

Tendo sido montado todo o plano de experimentos, conforme foi descrito neste

capítulo, a próxima etapa a ser cumprida para alcançar os objetivos definidos na Seção 1.1 e aproveitar as oportunidades discutidas na Seção 1.2 foi a realização dos experimentos, da coleta dos resultados e a construção de um serviço Android para monitorar a **App** em execução em um determinado momento e chavear para a frequência adequada visando à redução do consumo energético do *smartphone*.

Para realizar essa tarefa, decidiu-se criar um serviço Android, porque, conforme foi discutido na Seção 2.10, esse componente é mais adequado para monitorar o estado do sistema operacional e adotar as ações devidas sem que haja qualquer tipo de interação com o usuário.

O ambiente experimental montado para realizar os testes, os resultados dos experimentos e a arquitetura do serviço construído para cumprir os objetivos descritos na Seção 1.1, de acordo com as oportunidades discutidas na Seção 1.2, serão descritos no próximo capítulo.



## 4 Resultados

Como foi mencionado na Seção 1.1, este trabalho tem por objetivo realizar a otimização energética do *smartphone* através da criação de **grupos de aplicações**, de modo a fazê-lo consumir menos na realização das seguintes tarefas: *Stream* de Vídeo, Navegação em mapas usando-se o *GPS* e Navegação na internet usando-se o aplicativo padrão do Android (Browser).

Para chegar-se a uma frequência ideal, levando-se em consideração a motivação mostrada na Seção 1.2, foram elaborados testes com os aplicativos mais comumente usados na realização das atividades acima mencionadas, pois, segundo Nielsen (NIELSEN, 2011), os aplicativos *Maps* e *Youtube* estão entre os mais comumente usados, como foi discutido no Capítulo 3. Além disso, o aplicativo *Waze* também vem despontando como uma alternativa ao *Maps*, pois ele ainda conta com a vantagem de ser um aplicativo do tipo *Social Network*, permitindo a atualização do trânsito quase que em tempo real.

A fim de encontrar-se uma frequência ideal, conforme mostrado na Seção 1.2, foi necessário estabelecer-se uma comparação com aplicativos que realizavam tarefas semelhantes às dos aplicativos apontados por Nielsen (NIELSEN, 2011) e, por isso, os aplicativos *Dailymotion* e *Vimeo* foram inseridos na categoria de aplicativos de *Stream* de Vídeo.

Assim, com a finalidade de realizar-se a análise do consumo energético dos aplicativos acima mencionados, configurou-se o módulo de escalonamento de frequência para usar-se o *governor userspace*, que possibilita a um aplicativo do espaço do usuário com permissões administrativas configurar a frequência desejada no processador, conforme explicado na Seção 2.9. Além disso, foram coletados dados de potência instantânea do dispositivo móvel na realização da mesma tarefa em cada uma das 13 frequências de operação disponíveis, conforme descrito no Capítulo 3.

É importante destacar, contudo, que, algumas vezes, a frequência ideal para a realização de uma tarefa resulta em um pior desempenho do funcionamento do dispositivo móvel como um todo. Nesses casos, adota-se a estratégia de trabalhar em uma frequência superior por um tempo e, só após determinado intervalo de tempo, adota-se a frequência considerada como ideal, de modo a não se prejudicar a experiência do usuário.

Dessa forma, para executar-se a etapa responsável por realizar o teste de aderência (Seção 2.4) das amostras coletadas quanto à distribuição normal, utilizou-se a função *lillie.test()*, disponível no pacote *nortest*(RPROJECT, 2012). Esse teste, como descrito na Seção 2.5, é um caso particular do teste de **Kolmogorov-Smirnov**, que testa a aderência de uma distribuição de probabilidade empírica com a distribuição normal.

Tendo-se, então, certificado de que as amostras coletadas não seguiam a distribuição de probabilidade normal, foi realizado o teste de **Wilcoxon-Mann-Whitney** para compararem-se as amostras e ter-se certeza de que elas não provinham da mesma distribuição de probabilidade, assim como analisar-se se havia subsídios suficientes para afirmar-se que as potências instantâneas de determinada frequência eram estatisticamente maiores, menores ou iguais às potências obtidas em uma outra frequência, como discutido no Capítulo 3.

Depois de comparadas as potências instantâneas, chegou-se a uma ideia de tendência de economia que precisaria ser confirmada pela etapa seguinte, a análise numérica, como descrito na Seção 3.1.4.

Contudo, para mostrar a aplicação do plano de experimentos descrito no Capítulo 3, será explicado como esse plano foi aplicado para o caso do **App Youtube** utilizando-se a tecnologia **Wi-Fi** de transmissão de dados, o que será feito na Seção 4.2. Antes, porém, na Seção 4.1, será descrito o ambiente experimental utilizado na realização deste trabalho, como foi registrado no Capítulo 3. Nas seções seguintes, serão apontados os resultados obtidos em cada uma das categorias estudadas durante a realização deste trabalho.

## 4.1 Ambiente Experimental

Como foi discutido no Capítulo 3, para a completa definição de um plano experimental, é preciso, além de definir os passos descritos nesse capítulo, determinar o ambiente onde serão realizados os experimentos necessários para a construção da pesquisa experimental.

Assim, nesta seção será explicado como o ambiente experimental foi definido e como foram realizadas as coletas para a realização das análises, conforme se explicou na Seção 3.1.3 e na Seção 3.1.4.

O ambiente experimental usado durante a realização deste trabalho está mostrado na Figura 13. Nela, estão os principais componentes utilizados, e suas ligações ocorrem como foi descrito na Etapa 3.1.1.

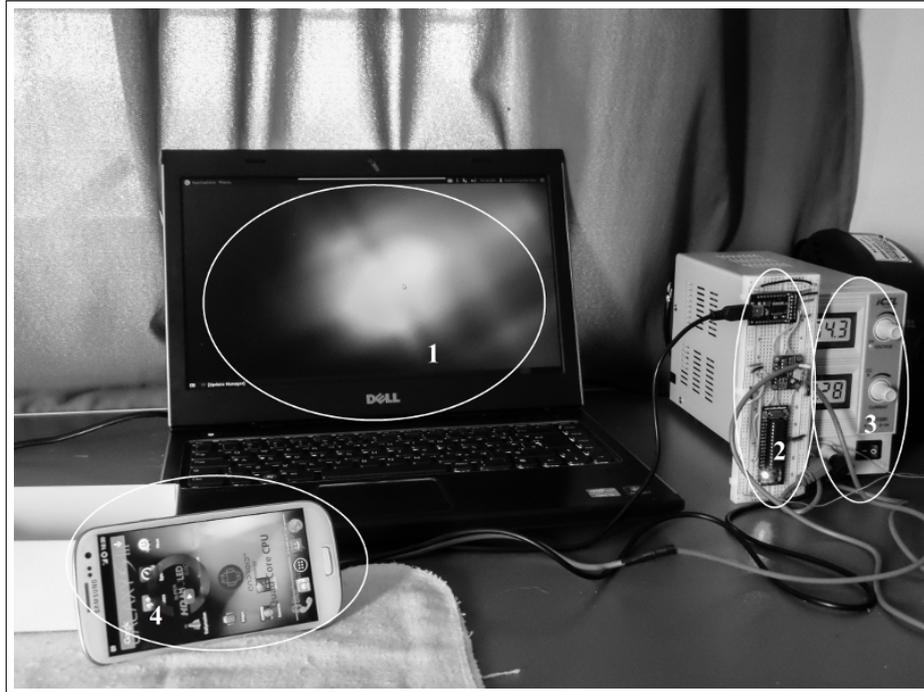


Figura 13 – Ambiente Experimental

**1.Computador:** O elemento 1 mostrado na Figura 13 representa o computador responsável por processar os dados coletados pelo circuito mostrado em 2 e gravá-los em um arquivo para ser futuramente analisado no procedimento de análise estatística. Ele é um Laptop Dell, com processador Intel Core i5, 12Gb de memória Ram, executando um sistema operacional Ubuntu Linux 12.04 64 bits em uma máquina virtual VMWare Workstation 10 com 2 processadores, 8Gb de memória Ram e disco rígido de 100Gb.

**2.Circuito de Medição:** O elemento 2 mostrado na Figura 13 é o circuito responsável por realizar as coletas de potências instantâneas do *smartphone* Galaxy SIII durante a realização das atividades estudadas nesta pesquisa. O circuito é formado pelos componentes mostrados na Figura 7 e estão descritos na Seção 3.1.1.

**3.Fonte de Alimentação:** O elemento 3 mostrado na Figura 13 é a fonte de alimentação responsável por fornecer uma tensão de  $4.4V$  constante ao *smartphone*. Ela é uma fonte Icel Manaus, modelo PS1500, com tensão de saída de 0-15V, corrente de 0-3A e exatidão de  $\pm 0,5\%$ . Mais detalhes podem ser obtidos em Icel Power Supply (ICEL, 2014).

**4.Smartphone:** O elemento 4 mostrado na Figura 13 é o *smartphone* escolhido para ser estudado durante a realização deste trabalho. Ele é um Galaxy SIII da Samsung, com 16Gb de memória interna, modelo GT-I9300, com tela super AMOLED de 16 milhões de cores, Sistema Operacional Cyanogen Mod 10.1, Processador Cor-

tex A9 Quad-Core e 1Gb de memória Ram. Mais informações podem ser obtidas em Galaxy SIII (ARENA, 2014).

Nesse ambiente, o monitor de corrente/potência INA219(TEXAS, 2011) do circuito de medição é ligado em série à fonte de alimentação e ao *smartphone* Samsung Galaxy SIII. Simultaneamente, os neutros da bateria do dispositivo móvel e do circuito de medição são ligados à fonte em paralelo, de modo a permitir que o circuito de medição seja capaz de mensurar a tensão que está sendo aplicada ao *smartphone*. A seguir, o circuito de medição é ligado ao computador por meio do componente conversor USB/Serial, e este último é ligado ao microcontrolador da Atmel Atmega168P(ATMEL, 2009). O microcontrolador se liga ao circuito de medição de tensão e potência INA 219 por meio de seus pinos SDA e SCL, fazendo dessa forma uma conexão I2C.

Para realizar a medição utilizando-se o monitor de corrente e tensão INA 219 (TEXAS, 2011), foi desenvolvido um programa usando-se a plataforma Arduino que se servia de um *driver* próprio para realizar a interface entre o microcontrolador Atmega168P(ATMEL, 2009) e o INA 219(TEXAS, 2011), desenvolvido pela Adafruit Industries(ADAFRUIT, 2014) e modificado para aumentar a taxa de amostragem. Esse programa realizava a coleta e disponibilizava o dado coletado por meio da interface serial virtual criada pelo conversor Serial/USB, utilizado no circuito criado para a realização deste trabalho e discutido na Seção 3.1.1.

Está, portanto, descrito o ambiente experimental e, assim, conclui-se a descrição do plano de experimentos iniciada no Capítulo 3. Agora, passa-se à construção de um exemplo, a ser descrito na Seção 4.2, para mostrar-se como esse plano foi posto em prática a fim de serem realizadas as coletas e análises necessárias a partir das quais se construirá o mecanismo **gDFS**, proposto neste trabalho. Depois de apresentar-se esse exemplo, serão elencados todos os resultados das análises feitas com os aplicativos e as condições de execução estabelecidas no plano experimental descrito na Seção 3.1.

## 4.2 Análise do aplicativo de *Stream* de Vídeo Youtube usando rede de dados Wi-Fi

Passa-se agora à apresentação do exemplo acima apontado. Assim, considerando-se o aplicativo de *stream* de vídeo Youtube, serão mostrados todos os procedimentos adotados e os respectivos resultados. Dessa forma, nesta seção, será descrito todo o procedimento de análise estatística e numérica para a definição da frequência ideal para exibição de vídeos, usando-se este aplicativo com uma conexão de dados Wi-Fi.

A fim de serem realizadas as análises estatística e numérica, primeiramente foram coletadas três amostras de corrente e tensão para cada uma das 13 frequências de operação

disponíveis (passo 6 do processo descrito na Etapa 3.1.1). As informações presentes no arquivo usado como base para calcular a potência instantânea de uma dada frequência constam na Tabela 3, na qual estão as informações de um arquivo de coleta, conforme explicado na Seção 3.1.1. Após ser coletado, insere-se no arquivo gerado uma coluna de potência, criando-se, assim, um novo *layout* de arquivo com as potências instantâneas que será usado nos procedimentos de análise numérica e estatística, como mostrado na Tabela 4 relativa às informações de um arquivo com potência, conforme discutido na Seção 3.1.1.

Realiza-se, então, o teste de normalidade, usando-se a função `lillie.test()` (RPROJECT, 2012) do aplicativo estatístico **R** para saber-se se a amostra em questão adere, ou não, a uma distribuição normal. O resultado é mostrado na Tabela 6.

Tabela 6 – Resultado do teste Lilliefors para as séries de potência do Youtube usando WiFi

Freq(MHz)	D
200	0.0538
300	0.0609
400	0.0520
500	0.0590
600	0.0555
700	0.0461
800	0.0505
900	0.0540
1000	0.0469
1100	0.0501
1200	0.0453
1300	0.0563
1400	0.0582

As frequências de operação variam de 200 MHz a 1400 MHz, aumentando a uma taxa de 100 MHz e todas apresentaram o mesmo valor-p ( $2.2 \times 10^{-16}$ ).

Como mencionado na Seção 2.5, o teste postula em sua hipótese nula que a distribuição empírica não segue a distribuição normal. Esse fato é comprovado pelo valor D da estatística. Entretanto, todos os experimentos realizados excederam 15000 amostras; logo, para  $D = \frac{1.36}{\sqrt{15000}} = 0.0111$ , todas as amostras excedem o valor limite, rejeitando, como mostrado na Tabela 6, a hipótese nula de que a distribuição em questão segue uma Normal e, de acordo com a escala de Fisher, um *valor - p*  $\leq 1 \times 10^{-3}$  constitui uma forte evidência para a afirmação do teste em questão, conforme descrito na Seção 2.3. Dessa forma, fica bastante visível o fato de que as séries de potência não seguem uma distribuição normal, tornando-se mais adequado o uso de um teste estatístico não paramétrico, como foi discutido na Seção 3.1.3.

Tendo-se em vista que a amostra não segue uma distribuição normal, ela é selecio-

nada para o rol de amostras, pelos motivos explicados na Seção 3.1.3, a serem comparadas com o teste não paramétrico de **Wilcoxon-Mann-Whitney** no procedimento seguinte.

Como são realizadas três iterações de amostragem para cada uma das frequências, primeiramente se deve testar se as amostras apresentam consumo semelhante, conforme mostrado na Tabela 7, a qual lista o resultado dos consumos nas iterações. Essa é uma forma de certificar-se acerca da não existência de falhas durante a fase de coleta.

Tabela 7 – Consumo em Joules de cada uma das iterações

Amostra	Consumo(J)
1	94.29871
2	88.99675
3	88.67001

Conforme se pode observar na Tabela 7, os consumos resultaram bem próximos, principalmente os da segunda e terceira iterações. Tendo-se constatado isso, escolheu-se uma iteração de cada frequência para compará-la com as outras frequências a fim de encontrar-se a frequência de operação que possui maior tendência de economia, aqui nomeada como frequência ideal. Para essa comparação, usou-se novamente o teste de **Wilcoxon-Mann-Whitney**, colocando-se a hipótese alternativa para "menor que" a fim de encontrar-se a frequência que possui as menores potências instantâneas, como mostrado na Tabela 8, referente ao resultado da comparação entre frequências. A comparação entre frequências ocorreu sempre usando-se a iteração 2, conforme foi explicado na Seção 3.1.2, de modo a não enviesar as comparações realizadas, pois não se desejava comparar sempre as iterações de amostragem das frequências com menor consumo, e sim determinar se em uma dada frequência o *smartphone* possui menores potências instantâneas.

Tabela 8 – Comparação entre as séries de potência instantânea produzidas pela execução do Youtube nas frequências disponíveis usando a rede de dados WiFi

Frequências(MHz)	P-Value
200 e 300	0.008145
200 e 400	$1.003 \times 10^{-12}$
200 e 500	$2.2 \times 10^{-16}$
200 e 600	$2.2 \times 10^{-16}$
200 e 700	0.02795
200 e 800	$2.2 \times 10^{-16}$
200 e 900	$2.2 \times 10^{-16}$
200 e 1000	$2.2 \times 10^{-16}$
200 e 1100	$2.2 \times 10^{-16}$
200 e 1200	0.2563
200 e 1300	$2.2 \times 10^{-16}$
200 e 1400	$2.2 \times 10^{-16}$

Infere-se, a partir dos resultados mostrados na Tabela 8, que a frequência de 200Mhz possui as menores potências instantâneas. Assim, essa frequência foi denominada de frequência ideal para a exibição de vídeo no aplicativo Youtube, usando-se a conexão de dados Wi-Fi, pois as hipóteses alternativas usadas em todos esses testes foi a de que a primeira amostra (200Mhz) era menor que a segunda (outros valores comparados a 200 Mhz) e boa parte dos valores-p possuem valores bem menores que 0,05 que, de acordo com a escala de Fisher, representa uma evidência fortíssima para a afirmação que está sendo feita, como pode ser visto na Seção 2.3.

Contudo, em alguns casos, o teste estatístico não foi capaz de afirmar isso com tanta certeza. Devido a isso, denominou-se essas frequências de "aceitáveis", que, no caso em estudo, são as seguintes: 300MHz, 400MHz, 700MHz e 1.2GHz. Essas frequências possuem, em sua maioria, valores-p superiores a 0,05 que, de acordo com a escala de Fisher, representa apenas uma evidência marginal do que se está afirmando, como pode ser verificado na Seção 2.3. O gráfico com os consumos dessas frequências aceitáveis e da frequência ideal está registrado na Figura 14.

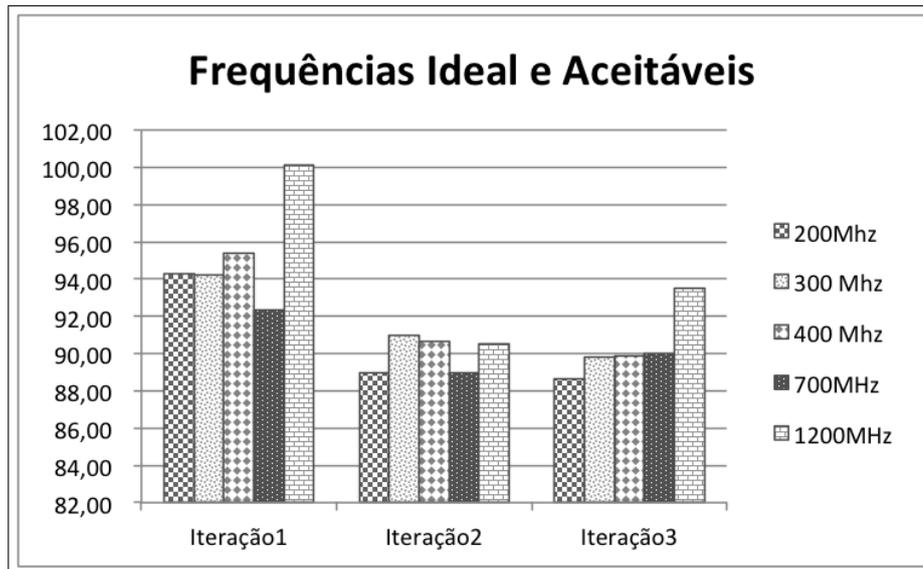
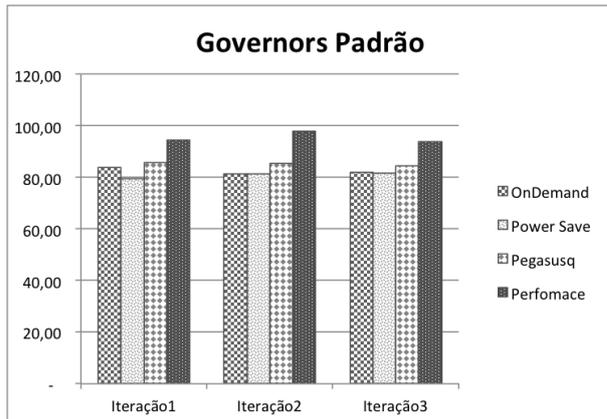


Figura 14 – Frequências ideal e aceitáveis

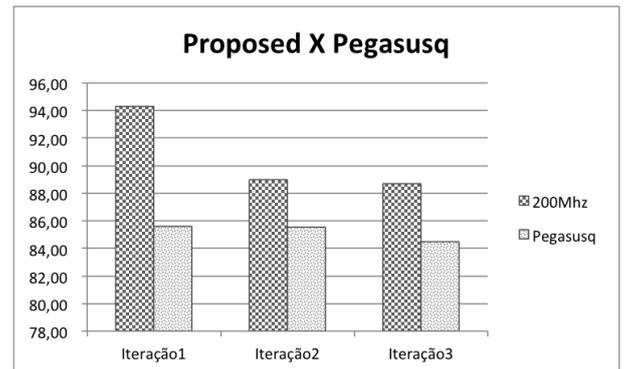
Dando-se continuidade a esse experimento, plotou-se o consumo das três iterações em cada uma das frequências para mostrar-se que elas possuem um perfil de consumo semelhante e, por isso, são consideradas como frequências aceitáveis. Ainda como pode ser visto na Figura 14, a iteração 1 é muito instável, ao contrário das iterações 2 e 3, motivo pelo qual também se escolheu realizar as comparações entre frequências sempre com iteração 2, conforme foi discutido na Seção 3.1.2.

Concluída essa etapa, fez-se necessário comparar os consumos dos *Governors* existentes no *Kernel* Android, usando-se o método de integração numérica proposto na Seção 2.7, e verificar qual deles apresentava o menor consumo. O resultado dessa comparação, registrado na Figura 15a, aponta que os consumos dos *governors* **Pegasusq** e **PowerSave** são os menores dentre todos os outros *governors*.

A partir desses resultados, chegou-se ao que está exposto na Figura 15, apresentada a seguir.



(a) Consumo dos Governors Padrão do Android



(b) Comparação método proposto x governador Pegasusq

Figura 15 – Comparação entre consumos da solução proposta e os governors do Kernel Android

A partir da observação dos gráficos registrados na Figura 15, percebem-se dois fatos bastante interessantes: o *governor* **PowerSave** apresenta um menor consumo em relação ao *governor* **Pegasusq**, como observado na Figura 15a, e a solução proposta apresenta um consumo maior que os apresentados pelas estratégias de chaveamento **PowerSave** e **Pegasusq**, como pode ser visto na Figura 15b.

Diante desses fatos, faz-se necessário considerar o que se defendeu na Seção 1.2: o fato de chavear a frequência a todo instante consome uma maior quantidade de energia e, assim, é preciso o estabelecimento de uma frequência fixa de operação do processador, que não necessariamente é a menor frequência.

Para comprovar esses fatos, realizou-se, então, um outro experimento, em que se exibiu um vídeo por 10 minutos contínuos, sem trocar de aplicação com todas as frequências de maneira fixa e com os *governors* padrão do Android. O resultado desse experimento está registrado na Figura 16.

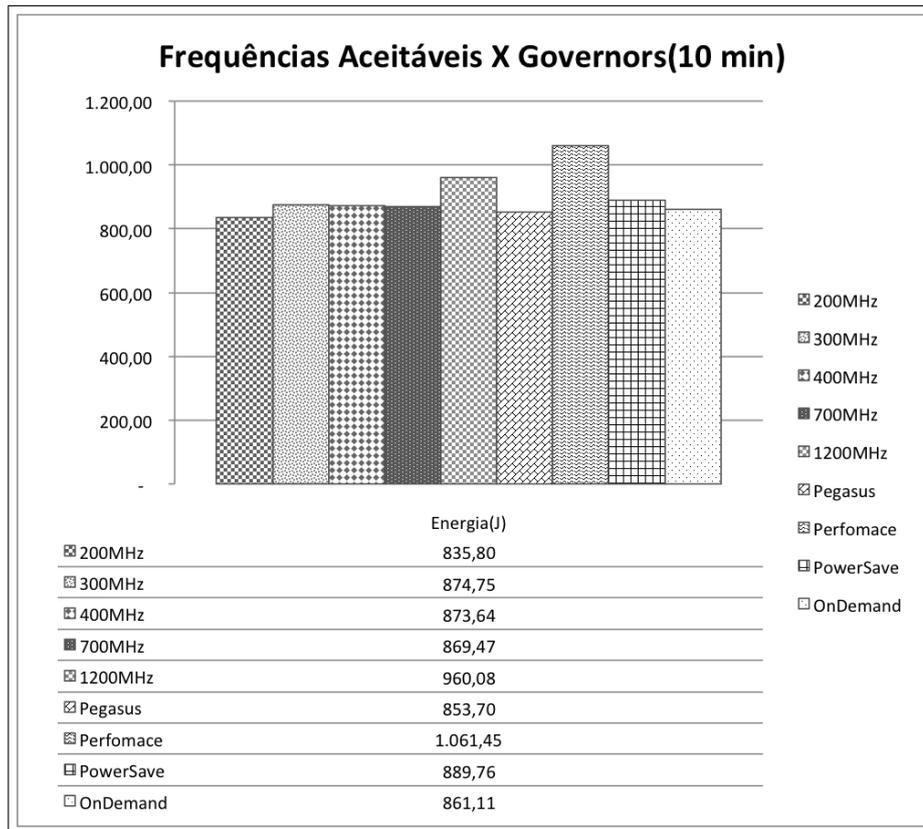


Figura 16 – Frequências ideal e aceitáveis x governors padrão do Android

Como é possível perceber na Figura 16, a abordagem proposta neste trabalho já apresenta um menor consumo a médio prazo. Isso comprova que é possível configurar o dispositivo móvel de maneira a já consumir menos energia a médio prazo.

Outro fato a ser destacado é que, durante a realização desta pesquisa, constatou-se que não se deve falar em uma frequência ótima que irá consumir menos energia; o mais correto é falar sobre frequências que possuem uma maior tendência a economizar energia. Isso se deve ao fato de existirem muitas variáveis, como: (a) condições de luminosidade do *display* do *smartphone*; (b) condições do sinal da antena de celular ou Wi-Fi no momento atual de uso do dispositivo; (c) condições de carga do processador e do sistema operacional no momento atual de uso; (d) forma como as aplicações que estão executando no momento funcionam; entre outras variáveis.

Constatados esses fatos, notou-se, contudo, uma limitação na abordagem construída (**gDFS**), verificando-se a impossibilidade de se encontrar uma frequência ótima de operação enquanto não se conseguisse construir um mecanismo mais inteligente que pudesse considerar as variáveis acima mencionadas e tomar decisões mais inteligentes. Essa discussão será feita no Capítulo 5.

Constata-se, assim, que se pode construir estratégias que tenham a tendência de economizar mais energia e faz-se necessária a utilização de técnicas mais inteligentes as quais permitam que as análises realizadas durante este trabalho sejam feitas de forma

online.

Para ter-se uma ideia a respeito do quanto a estratégia adotada pode tender a economizar a médio prazo, foi feita uma comparação entre a estratégia adotada e os *governors* padrão do Android na execução de um vídeo de 10 minutos. Chegou-se, então, aos resultados registrados na Figura 17.

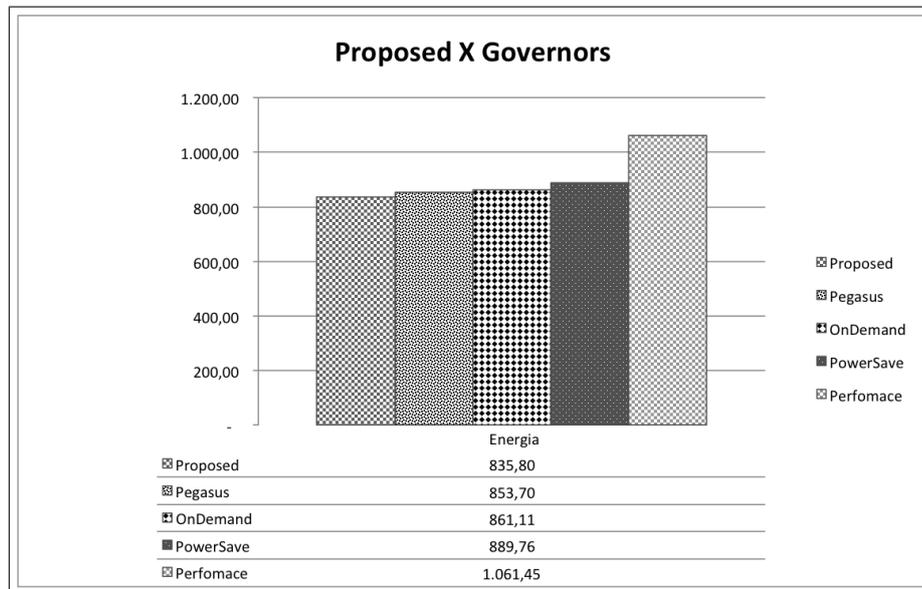


Figura 17 – Frequência ideal x governors padrão do Android

De acordo com o gráfico registrado na Figura 17, verifica-se que se chega a uma economia da ordem de *18 Joules*, se comparado com o *governor Pegasusq*, que configura a frequência de forma independente nos núcleos do processador do dispositivo móvel, conforme descrito na Seção 2.9; e uma economia da ordem de *226 Joules*, se comparado ao **Performance** que configura a frequência sempre para a maior frequência disponível no dispositivo, conforme descrito na Seção 2.9. Percentualmente, essas diferenças representam uma tendência de economia da ordem de **2,1%** e **21%**, respectivamente.

Tendo-se concluído esse experimento, fez-se outro, comparando-se apenas o consumo na frequência proposta neste trabalho com o ideal e o *governor Pegasusq* implementado no *Kernel* Android. Os resultados desse experimento são mostrados na Figura 18.

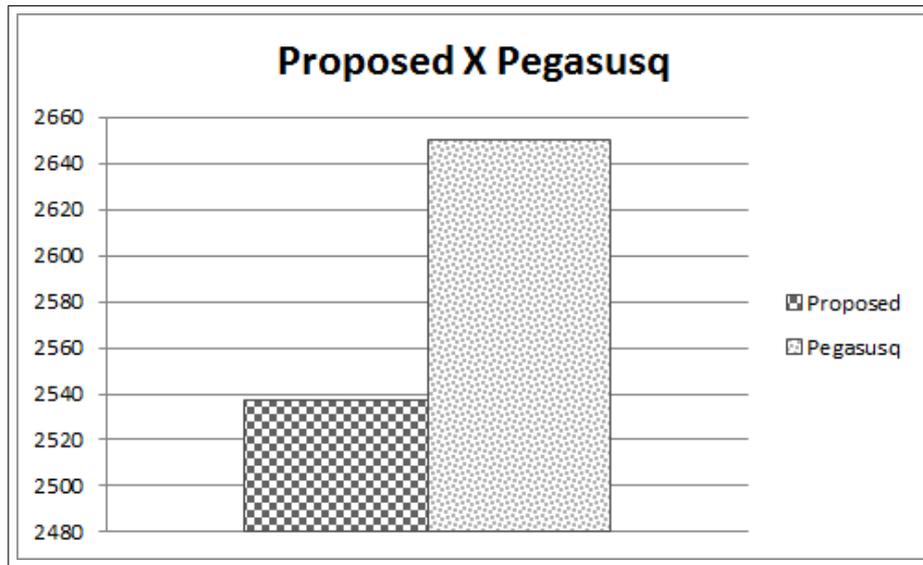


Figura 18 – Frequência ideal x governors padrão do Android

Observando-se o gráfico registrado na Figura 18, lê-se que o consumo do *governor Pegasusq* é da ordem de 2650 Joules, enquanto que o consumo na abordagem proposta é da ordem de 2537 Joules, ou seja, a abordagem proposta consome 113 Joules menos que a abordagem adotada pelo *governor Pegasusq*, o que resulta em uma economia percentual da ordem de 4,5%. Dessa forma, consegue-se mostrar que a economia é crescente de acordo com o tempo de execução da atividade pelo *smartphone*, pois o tempo cresceu na ordem de três vezes, e a economia percentual, na ordem de duas vezes.

Após apresentar-se a análise do **App Youtube** usando-se a rede de dados **Wi-Fi**, passa-se a mostrar os resultados obtidos a partir das análises dos outros aplicativos estudados e nas outras condições de execução estabelecidas durante a definição do plano experimental deste trabalho.

### 4.3 Análise de Aplicativos de *Stream* de Vídeo

Para a realização dos testes de aplicativos de *Stream* de Vídeo, foram coletados dados durante 1 minuto de exibição de vídeo nas 13 frequências de operação disponíveis no dispositivo Samsung Galaxy SIII. Os vídeos escolhidos para reprodução foram:

**Youtube** O vídeo "Mímica" do "Porta dos Fundos";

**Dailymotion** O vídeo "The Michael Jackson of Cotton Candy";

**Vimeo** O vídeo "Ivete Sangalo-Madison Square Garden" que é a abertura do show de Ivete Sangalo em Nova Iorque.

Essas escolhas foram feitas de forma aleatória, baseadas no tempo de exibição do vídeo. Todos esses vídeos tinham aproximadamente 1 minuto, ou seja, foram escolhidos

vídeos pequenos porque havia 13 frequências a serem estudadas em duas condições de exibição (**WI-Fi e 3G**).

### 4.3.1 Análise Youtube

Como a análise da exibição de vídeos usando-se o aplicativo Youtube com rede de dados Wi-Fi já foi realizada na Seção 4.2, nesta seção será feita a análise desse aplicativo apenas se usando a rede de dados 3G.

Feita a análise estatística e numérica descritas na Etapa 3.1.3 e na Etapa 3.1.4, chegou-se aos resultados registrados no gráfico da Figura 19.

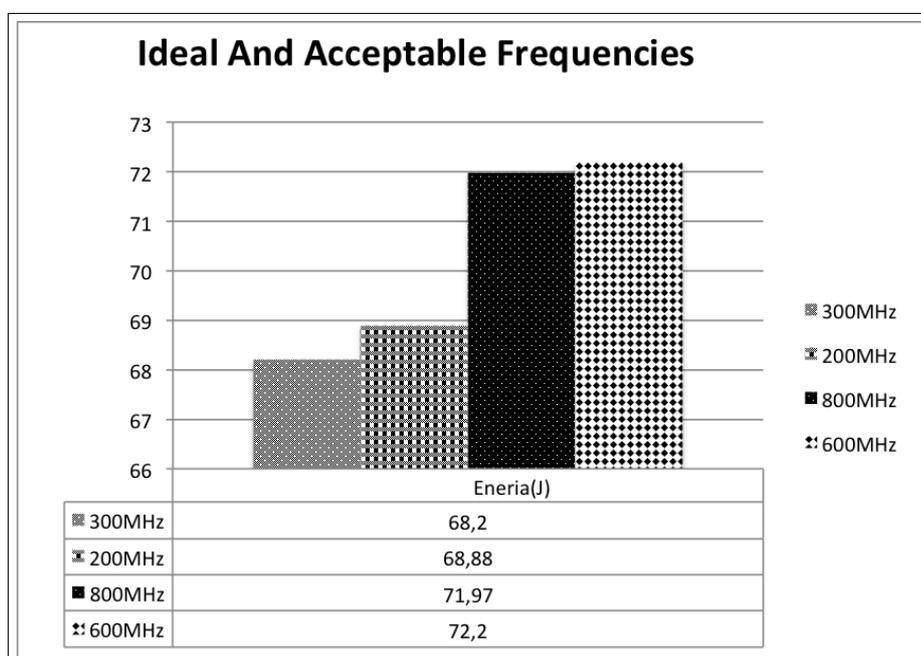


Figura 19 – Frequências ideal e aceitáveis para Youtube usando 3G

O gráfico registrado na Figura 19 mostra as frequências ideal e aceitáveis às quais se chegou após se realizarem as análises estatística e numérica. Observando-se esse gráfico, percebe-se que as frequências de 200 MHz, 300 MHz, 600 MHz e 800 MHz apresentam consumos muito próximos: apenas 4 Joules de diferença do menor para o maior.

Analisando-se os outros *governors* presentes no *Kernel* padrão do Android, chegou-se aos resultados mostrados no gráfico registrado na Figura 20.

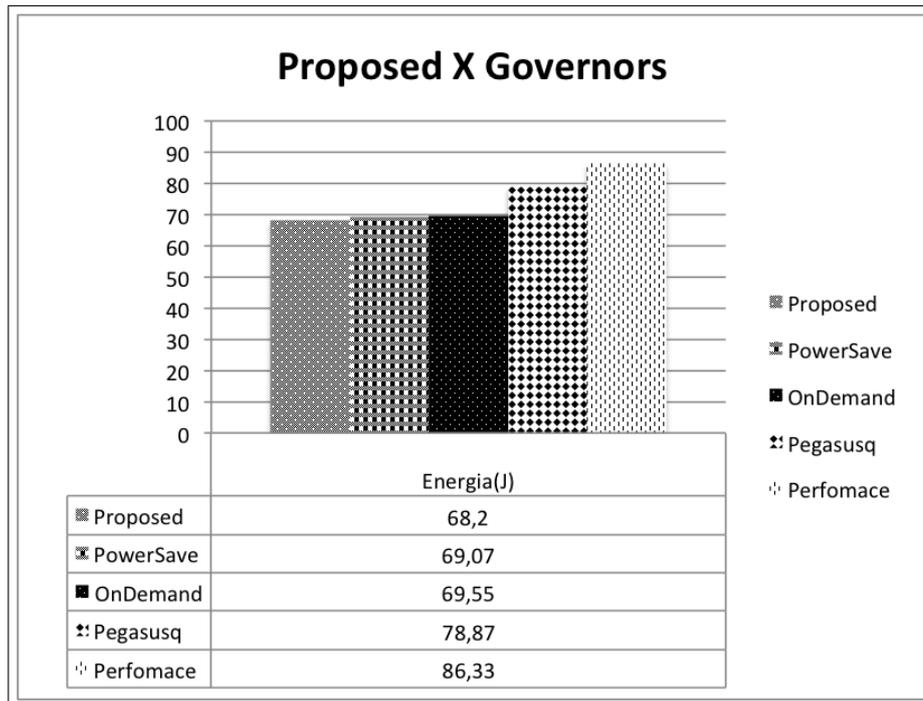


Figura 20 – Frequência ideal x governors padrão do Android para Youtube usando 3G

Analisando-se o gráfico registrado na Figura 20, percebe-se que a abordagem proposta apresentou consumo muito próximo às estratégias **PowerSave** e **OnDemand**. Isso aponta que, a curto prazo, pode-se usar qualquer uma das três estratégias para atingir-se o mesmo resultado de economia. Já em relação à abordagem **Perfomance**, obtém-se uma economia da ordem de 30 *Joules*, o que representa uma economia percentual da ordem de 35%.

#### 4.3.2 Análise Dailymotion

Ao realizarem-se as análises estatística e numérica, conforme descrito na Etapa 3.1.3 e na Etapa 3.1.4, do aplicativo de *Stream* de vídeo Dailymotion, utilizando-se uma conexão de dados Wi-Fi, verificou-se que a frequência ideal para exibição de vídeo usando-se esse aplicativo é de 200 *MHz*. O cálculo do consumo, conforme descrito na Etapa 3.1.4, aponta a frequência ideal e algumas frequências aceitáveis, conforme mostrado no gráfico registrado na Figura 21.

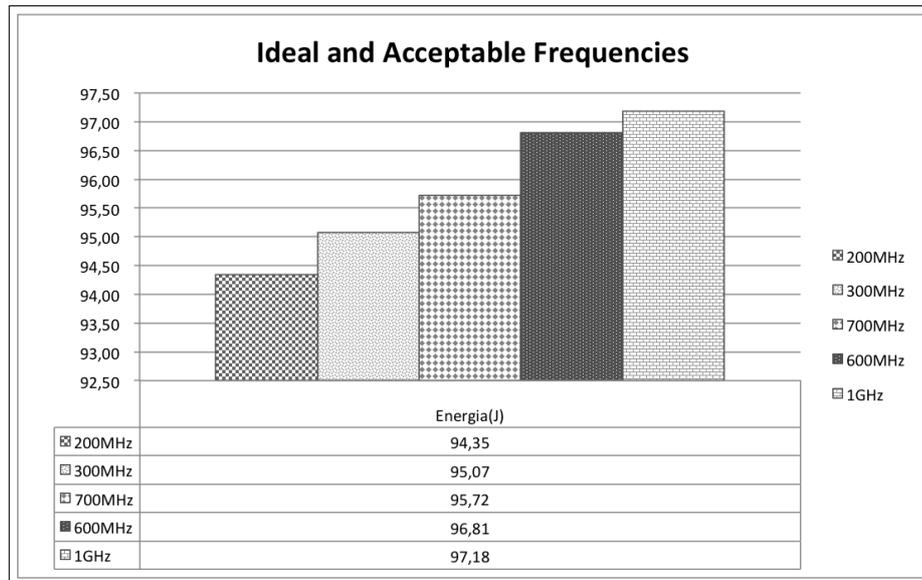


Figura 21 – Frequências ideal e aceitáveis para Dailymotion usando Wi-Fi

Pode-se perceber, a partir da análise do gráfico registrado na Figura 21, que as frequências de 200 MHz, 300 MHz, 600 MHz, 700 MHz e 1 GHz possuem consumos muito próximos: apenas 3 Joules de diferença do menor para o maior.

Analisando-se o consumo desse aplicativo, usando-se os outros *governors* presentes no *Kernel* padrão do Android, chega-se aos resultados mostrados na Figura 22.

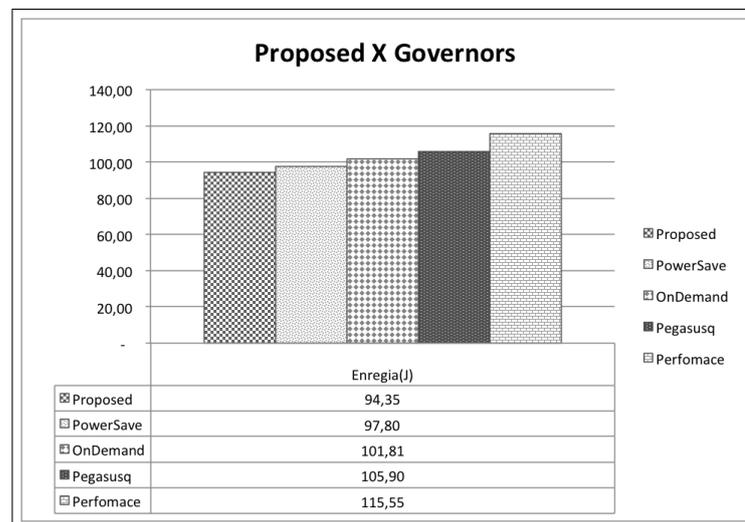


Figura 22 – Frequência ideal x governors padrão do Android para Dailymotion usando Wi-Fi

Observando-se o gráfico registrado na Figura 22, verifica-se que a abordagem proposta apresenta consumo muito próximo à estratégia **PowerSave**. Isso significa que, a curto prazo, pode-se usar qualquer uma das três estratégias para atingir-se o mesmo resultado de economia.

Em relação às outras abordagens, viu-se que o método proposto começou a apresentar uma diferença um pouco significativa a partir de 1 minuto de uso, revelando economias que variaram entre **8%** e **19%**, indicando, assim, uma tendência de economia muito maior a médio e longo prazos.

Já para a conexão de dados *3G*, o aplicativo de *Stream* de vídeo Dailymotion apresentou um comportamento um pouco diferente, assim como acontece com o aplicativo Youtube, devido às características dessa conexão, a qual exige uma maior comunicação com a Radio Resource Control(*RRC*), que é camada responsável por fazer a interface do sistema operacional Android com o dispositivo de rádio responsável por realizar a comunicação com a rede celular 2G/3G. Após terem sido feitas as análises estatística e numérica, conforme descrito na Etapa 3.1.3 e na Etapa 3.1.4, chegou-se à conclusão de que, nesse cenário, o aplicativo Dailymotion apresenta as frequências ideal e aceitáveis mostradas na Figura 23.

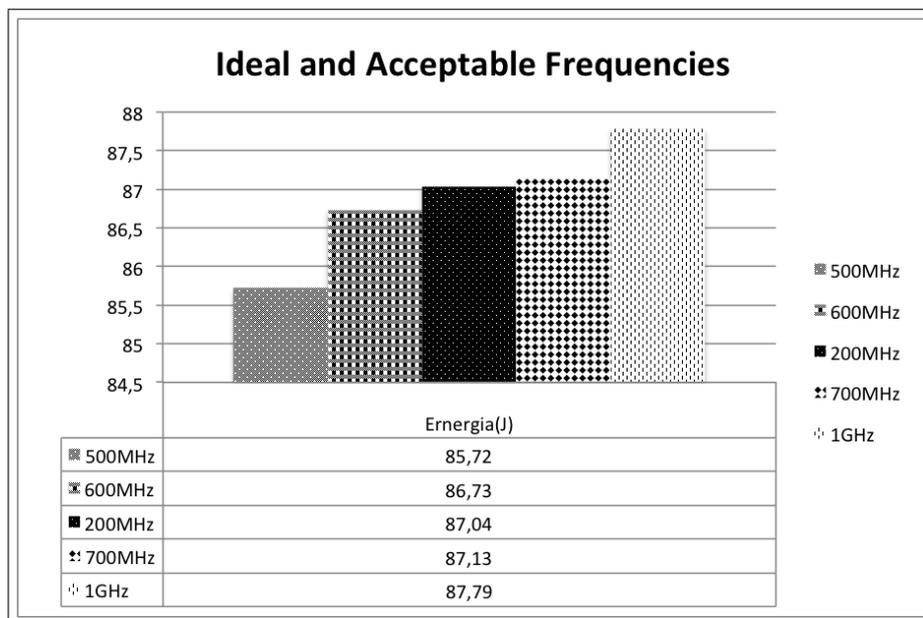


Figura 23 – Frequências ideal e aceitáveis para Dailymotion usando 3G

Examinando-se o gráfico registrado na Figura 23, vê-se que as frequências de 200 *MHz*(Ideal), 500 *MHz*, 600 *MHz*, 700 *MHz* e 1.0 *GHz* possuem consumos bastante próximos: apenas 2 *Joules* de diferença do menor para o maior. Isso indica que, a curto prazo, pode-se adotar qualquer dessas frequências sem grande aumento no consumo energético do dispositivo.

Comparando-se a abordagem proposta (**grupo** de *Stream* de vídeo) com outros *governors* presentes no *Kernel* do Android, chegou-se aos resultados mostrados na Figura 24.

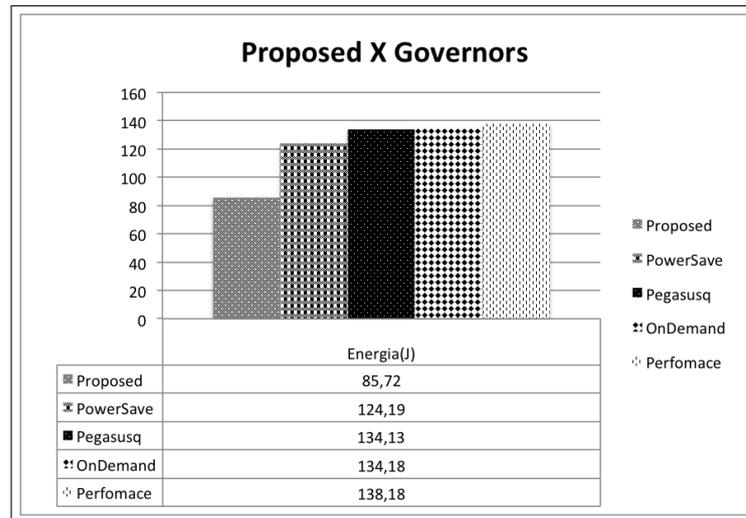


Figura 24 – Frequência ideal x governors padrão do Android para Dailymotion usando 3G

Tomando-se o gráfico registrado na Figura 24, infere-se que a abordagem proposta não apresenta consumo próximo a nenhuma das estratégias presentes por padrão no *Kernel* Android. O método proposto apresenta economias que variam entre **31%** e **38%**, indicando, dessa forma, uma tendência de economia muito maior a médio e longo prazos.

### 4.3.3 Análise Vimeo

Ao realizarem-se as análises estatística e numérica, conforme descrito na Etapa 3.1.3 e na Etapa 3.1.4, do aplicativo de *Stream* de vídeo Vimeo, utilizando-se uma conexão de dados Wi-Fi, verificou-se que a frequência ideal para exibição de vídeo usando-se esse aplicativo é de 500 *MHz*. O cálculo do consumo, conforme descrito na Seção 4.2, apontou a frequência ideal e algumas frequências aceitáveis, conforme mostrado no gráfico registrado na Figura 25.

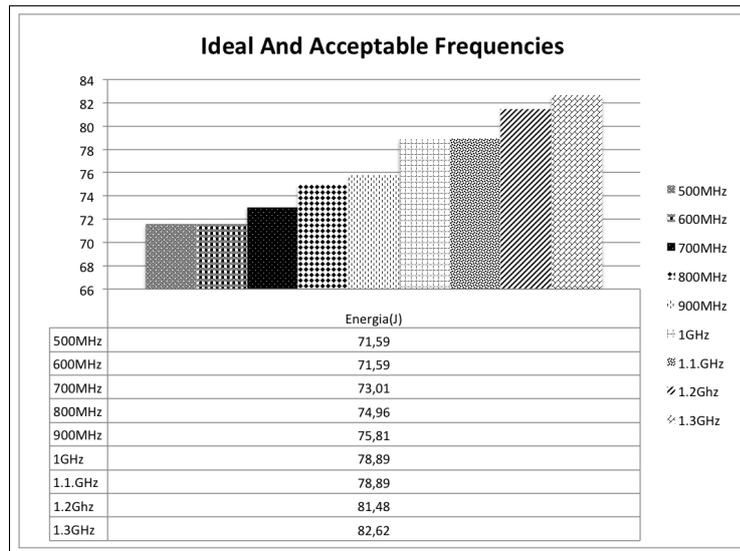


Figura 25 – Frequências ideal e aceitáveis para Vimeo usando Wi-Fi

Observando-se o gráfico registrado na Figura 25, percebe-se que as frequências de 500 MHz, 600 MHz, 700 MHz, 800 MHz, 900 MHz, 1.0 GHz, 1.1 GHz, 1.2 GHz, 1.3 GHz possuem consumos bastante próximos: apenas 9 Joules de diferença do menor para o maior consumo. Isso indica que, a curto prazo, pode-se adotar qualquer dessas frequências sem grande aumento no consumo energético do dispositivo.

Comparando-se a abordagem proposta com outros *governors* presentes no *Kernel* do Android, chegou-se aos resultados mostrados na Figura 26.

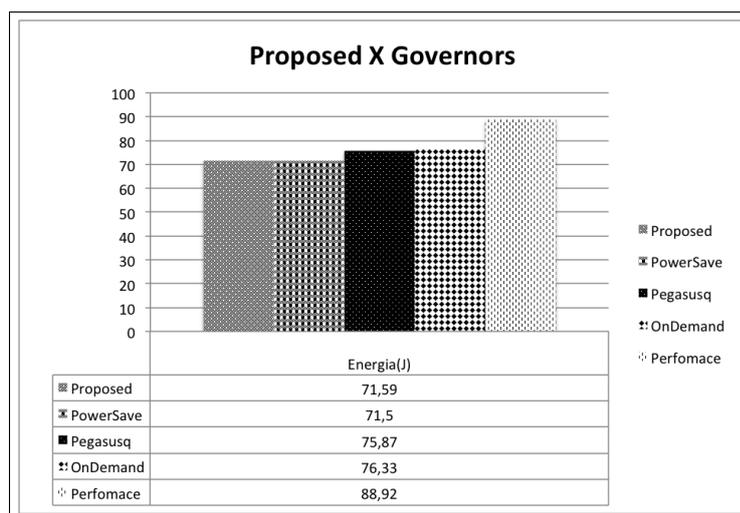


Figura 26 – Frequência ideal x governors padrão do Android para Vimeo usando Wi-Fi

Analisando-se os dados constantes no gráfico registrado na Figura 26, observa-se que a abordagem proposta apresenta consumo próximo à estratégia **PowerSave** presente por padrão no *Kernel* Android. O método proposto apresenta economias que variam entre 6% e 19%, o que indica uma tendência de economia muito maior a médio e longo prazos.

Já para a conexão de dados 3G, o aplicativo de *Stream* de vídeo Vimeo apresenta um comportamento um pouco diferente, assim como acontece com os outros aplicativos de exibição de vídeo, devido às características dessa conexão, que exige uma maior comunicação com a *RRC*. Após as análises estatística e numérica, conforme descrito na Etapa 3.1.3 e na Etapa 3.1.4, chegou-se à conclusão de que, nesse cenário, o aplicativo Vimeo apresenta as frequências ideal e aceitáveis mostradas na Figura 27.

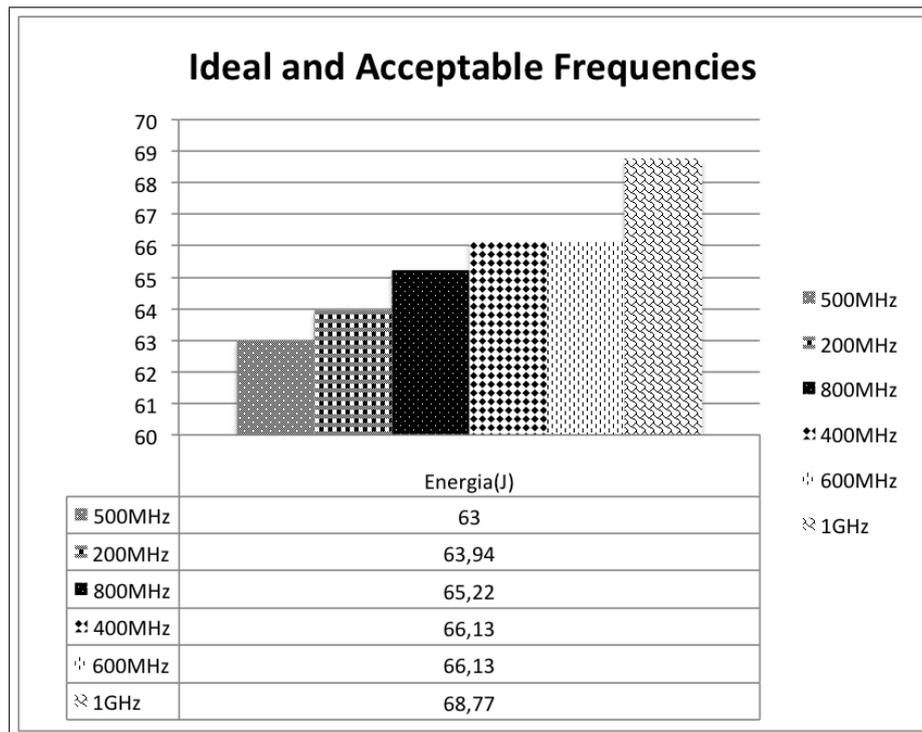


Figura 27 – Frequências ideal e aceitáveis para Vimeo usando 3G

Como se pode perceber no gráfico registrado na Figura 27, as frequências de 200 MHz, 400 MHz, 500 MHz (Ideal), 600 MHz, 800 MHz e 1.0 GHz possuem consumos bastante próximos: apenas 6 *Joules* de diferença do menor para o maior consumo. Isso indica que, a curto prazo, pode-se adotar qualquer uma dessas frequências sem grande aumento no consumo energético do dispositivo.

Comparando-se a abordagem proposta com outros *governors* presentes no *Kernel* do Android, chegou-se aos resultados mostrados na Figura 28.

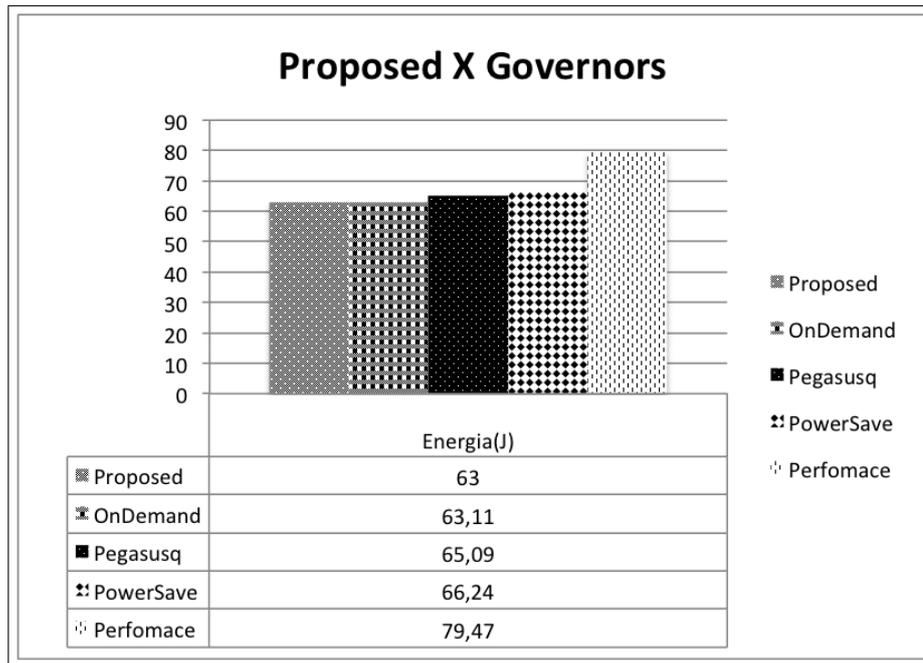


Figura 28 – Frequência ideal x governos padrão do Android para Vimeo usando Wi-Fi

De acordo com os dados do gráfico mostrado na Figura 28, vê-se que a abordagem proposta apresenta consumo próximo às estratégias **OnDemand**, **Pegasusq** e **PowerSave** presentes por padrão no *Kernel* Android. O método proposto apresenta economias que variam entre **0,17%** e **20%**, apontando, dessa forma, uma tendência de economia maior a médio e longo prazos.

#### 4.3.4 Conclusão das Análises de Aplicativos de *Stream* de Vídeo

A partir das análises feitas e mostradas nos gráficos e nas discussões realizadas nas subseções anteriores, é possível estabelecer uma faixa de frequência ideal para os aplicativos cujo propósito é exibir vídeos, ou seja, é possível estabelecer uma frequência de operação para o **grupo de aplicações** de *Stream* de vídeo. Essa faixa, de acordo com a abordagem construída por este trabalho, deverá ser usada no ajuste de frequência enquanto o usuário estiver realizando tarefas que envolvam a exibição de vídeo, a fim de atingir o objetivo de consumir menos energia durante a realização dessas atividades.

Em síntese verificou-se que, ao detectar que o usuário está executando uma aplicação de exibição de vídeo, o *smartphone* ajusta a frequência para a frequência ideal do aplicativo e, caso ele mude de aplicativo, o dispositivo móvel ajusta para a frequência ideal do **grupo**.

Os aplicativos possuem as seguintes frequências ideais e aceitáveis:

- Youtube(Wi-Fi)
  - Frequência ideal: 700 *Mhz*
  - Frequências aceitáveis: 200 *Mhz*,300 *Mhz*,400 *Mhz*, 1.2 *Ghz*
- Youtube(3G)
  - Frequência ideal: 300 *Mhz*
  - Frequências aceitáveis:200 *Mhz*,800 *Mhz*
- Dailymotion(Wi-Fi)
  - Frequência ideal: 200 *Mhz*
  - Frequência aceitáveis: 300 *Mhz*,600 *Mhz*,700 *Mhz*,1 *Ghz*
- Dailymotion(3G)
  - Frequência ideal: 500 *Mhz*
  - Frequências aceitáveis: 400 *Mhz*,600 *Mhz*,700 *Mhz*,800 *Mhz*,1.0 *Ghz*
- Vimeo (Wi-Fi)
  - Frequência ideal: 500 *Mhz*
  - Frequências aceitáveis: 200 *Mhz*,300 *Mhz*,400 *Mhz*,600 *Mhz*,700 *Mhz*,800 *Mhz*
- Vimeo(3G)
  - Frequência ideal: 500 *Mhz*
  - Frequências aceitáveis:200 *Mhz*,400 *Mhz*,600 *Mhz*,800 *Mhz*

A frequência ideal, de acordo com as análises, seria de 800 *Mhz* para esse grupo na 3G e de 700 *Mhz* na Wi-Fi.

## 4.4 Análise de Aplicativos de Navegação GPS

Para serem realizados experimentos com aplicativos de navegação *GPS*, não se utilizou a conexão de dados WI-Fi por entender-se que o principal uso desse tipo de aplicativo se dá em ambientes externos e em movimento. Assim, fez-se o acompanhamento das rotas previamente planejadas pelo usuário. Analisaram-se, então, dois principais aplicativos de navegação *GPS*: Google Maps e Waze. Essas análises foram feitas a partir da suposição de duas situações: cálculo de rota apenas e cálculo de rotas com iteração.

Quando se pensou no primeiro tipo de situação, cálculo de rota apenas, levou-se em consideração o momento em que o usuário está somente acompanhando o trajeto, sem fazer qualquer interação com o aplicativo em questão. Já no segundo momento, cálculo de rota com interação, pensou-se na ocasião em que o usuário está configurando uma rota a ser seguida no aplicativo. Desse modo, foram encontradas as frequências ideais e aceitáveis para esses dois momentos.

Os experimentos foram realizados, fazendo-se o cálculo da rota com ponto de origem na rua General Polidoro, localizada na Várzea, na altura do número 512, e com ponto de destino o C.E.S.A.R, localizado na rua Bione, no bairro do Recife. Foram feitos experimentos utilizando-se outras rotas como Recife-São Paulo e não se notou diferença significativa, de modo que foram mantidas apenas as coletas da rota General Polidoro - C.E.S.A.R.

#### 4.4.1 Análise Google Maps: Apenas Cálculo de Rota

A análise realizada com o aplicativo Google Maps para calcular rota e se manter atualizando-a indicou que ele apresenta menor potência instantânea quando o dispositivo móvel está configurado para trabalhar em uma frequência de 200 *MHz*. Porém, ao analisar-se o tempo de execução desse aplicativo para realizar a mesma tarefa, percebeu-se que houve diferença no tempo de execução. Assim, foi necessário fazer-se o cálculo da energia consumida, utilizando-se a metodologia descrita na Etapa 3.1.4, a fim de que se pudesse tomar uma decisão a respeito da frequência de operação ideal para esse aplicativo realizar essa atividade. Os tempos de execução estão descritos na Tabela 9.

Tabela 9 – Tempo de execução do Google Maps apenas para cálculo de rota

Freq(MHz)	200	300	400	500	600	700	800	900	1000	1100	1200	1300	1400
Tempo(S)	22	16	11	11	10	9	9	9	8	8	7	7	8

Feito o cálculo do consumo, conforme descrito na Etapa 3.1.4, chegou-se às frequências ideal e aceitáveis, que estão no gráfico registrado na Figura 29.

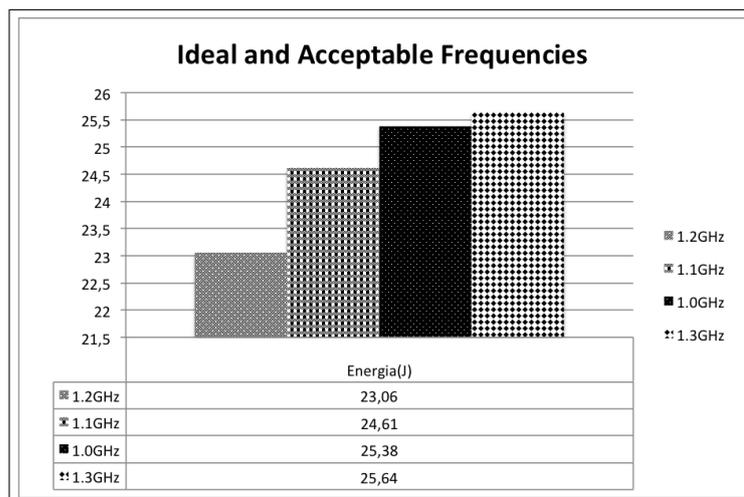


Figura 29 – Frequências ideal e aceitáveis para cálculo de rota usando Google Maps

Como se pode perceber no gráfico registrado na Figura 29, as frequências de 1.0 GHz, 1.1 GHz, 1.2 GHz e 1.3 GHz possuem consumos bastante próximos: apenas 3 Joules de diferença do menor para o maior consumo. Isso indica que, a curto prazo, pode-se adotar qualquer dessas frequências sem grande aumento no consumo energético do dispositivo.

Viu-se, portanto, a necessidade de fazer-se uma comparação estatística, utilizando-se o método descrito na Etapa 3.1.3 para poder-se afirmar qual das frequências acima mencionadas seria a frequência ideal de operação, pois o resultado do consumo é um número aproximado. Os resultados dessa análise estão registrados na Tabela 10.

Tabela 10 – Comparação entre frequências com relação à potência instantânea do Google Maps para o cálculo de rota

Frequências(GHz)	1.1 e 1.0	1.1 e 1.2	1.1 e 1.3
Resultados	p-value= 1.24e-05	p-value= 6.11e-11	p-value < 2.2e-16

Após essa análise, têm-se fortes indícios para indicar a frequência de 1.1 GHz como sendo ideal, pois, como pode ser visto na tabela acima, todos os testes feitos resultaram na afirmação da hipótese alternativa de que a amostra da frequência de 1.1 GHz é menor que as outras amostras das outras frequências que foram comparadas, e o grau de certeza é confirmado pelos valores baixos do valor-p nas respectivas comparações.

Comparando-se a abordagem proposta (**grupo** de aplicações para navegação GPS) com outros *governors* presentes no *Kernel* do Android, chega-se aos resultados mostrados na Figura 30.

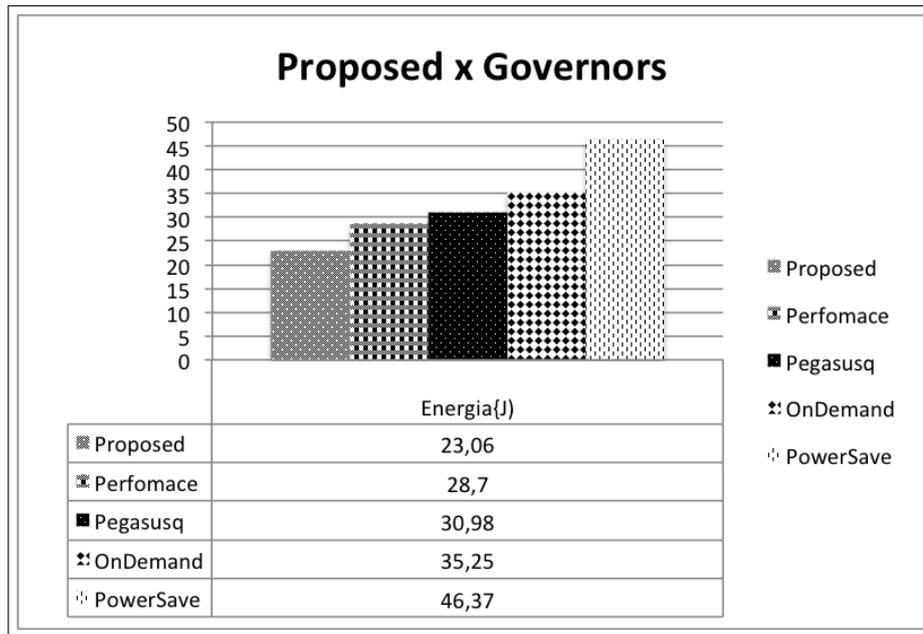


Figura 30 – Frequência ideal x governors padrão do Android para cálculo de rota usando Google Maps

De acordo com os dados apresentados no gráfico mostrado na Figura 30, vê-se que a abordagem proposta apresenta consumo próximo à estratégia **Performance** presente por padrão no *Kernel* Android. O método proposto apresenta economias que variam entre **20%** e **50%**. Isso indica uma tendência de economia muito maior a médio e longo prazos.

#### 4.4.2 Análise Google Maps: Cálculo de Rota e Iteração

Para fazer-se o cálculo de rota e ainda se atender às requisições do usuário, a frequência ideal de acordo com a menor potência também seria de  $200\text{ MHz}$ . Porém, novamente, o tempo de execução é bastante discrepante, tornando-se necessária uma análise do consumo do aplicativo para decidir-se qual a melhor frequência. As frequências e seus respectivos tempos de execução estão descritos na Tabela 11.

Tabela 11 – Tempo de execução do Google Maps para cálculo de rota e iteração

<b>Freq(MHz)</b>	200	300	400	500	600	700	800	900	1000	1100	1200	1300	1400
<b>Tempo(S)</b>	24	18	15	14	13	12	11	11	11	11	10	11	12

Dessa forma, fez-se necessária a realização de uma análise numérica, usando-se o método descrito na Etapa 3.1.4, com todas as frequências a fim de decidir-se qual seria a frequência ideal. Após a realização da análise numérica, chegou-se aos resultados mostrados no gráfico registrado na Figura 31.

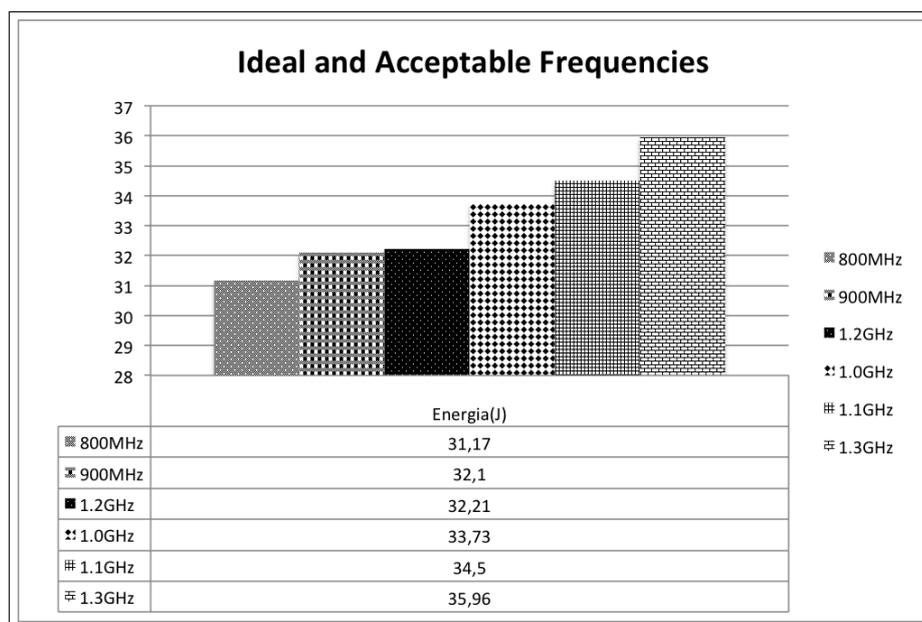


Figura 31 – Frequências ideal e aceitáveis para cálculo de rota com iteração usando Google Maps

Como se pode perceber no gráfico registrado na Figura 31, as frequências de 800 MHz, 900 MHz, 1.0 GHz, 1.1 GHz, 1.2 GHz e 1.3 GHz possuem consumos bastante próximos: apenas 5 Joules de diferença do menor para o maior consumo. Isso indica que, a curto prazo, pode-se adotar qualquer uma dessas frequências sem grande aumento no consumo energético do dispositivo.

Diante disso, viu-se a necessidade de realizar-se uma comparação estatística, utilizando-se o método descrito na Etapa 3.1.3 para poder-se afirmar qual das frequências acima mencionadas viria a ser a frequência ideal de operação, pois o resultado do consumo é um número aproximado. Os resultados dessa análise estão mostrados na Tabela 12.

Tabela 12 – Comparação entre frequências em relação à potência instantânea do Google Maps para cálculo de rota e iteração

Frequências(MHz)	800 e 900	800 e 1000	800 e 1100	100 e 1200	1100 e 1300
Resultados	p-value=5.3e-09	p-value < 2.2e-16	p-value < 2.2e-16	p-value < 2.2e-16	p-value < 2.2e-16

A partir dos resultados dessa análise, verifica-se que há fortes indícios para indicar a frequência de 800 MHz como a ideal, pois, como pode ser visto na tabela acima, todos os testes feitos resultaram na afirmação da hipótese alternativa de que a amostra da frequência de 800 MHz é menor que as outras amostras das outras frequências que foram comparadas, e o grau de certeza é confirmado pelos valores baixos do valor-p nas respectivas comparações.

Comparando-se a abordagem proposta com outros *governors* presentes no *Kernel* do Android, chegou-se aos resultados mostrados na Figura 32.

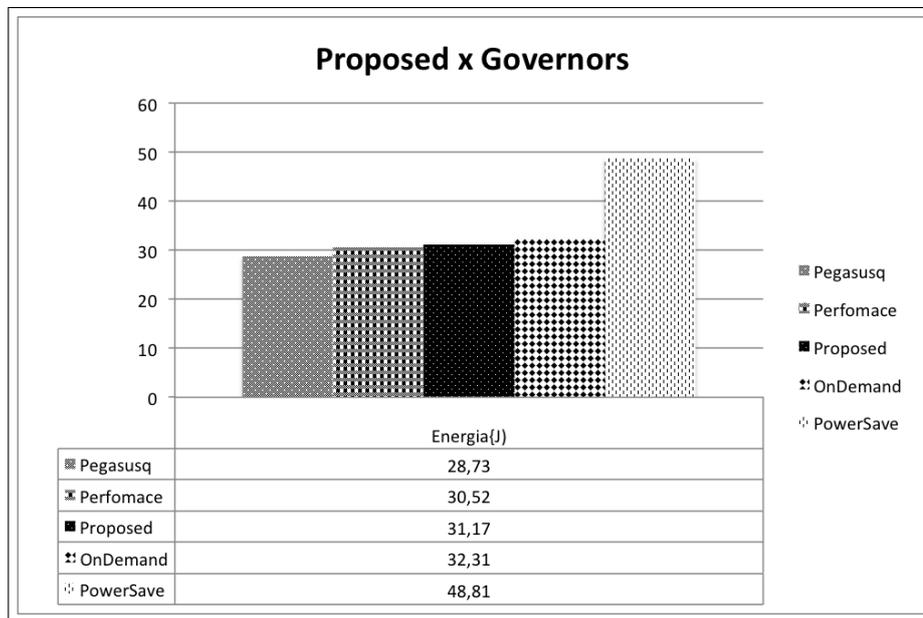


Figura 32 – Frequência ideal x governors padrão do Android para cálculo de rota com iteração usando Google Maps

Observando-se o gráfico registrado na Figura 32, percebe-se que a abordagem proposta apresenta consumo próximo às estratégias **Perfomace** e **Pegasusq** presentes por padrão no *Kernel* Android. O método proposto não apresenta economia a curto prazo, pois os tempos de experimento não foram suficientes para mostrar a economia gerada pela utilização do método. Tem-se, apenas, uma tendência de economia a médio e longo prazos, como ocorreu no caso mostrado na Seção 4.2.

#### 4.4.3 Análise Waze: apenas cálculo de rota

A análise realizada com o aplicativo Waze para calcular rota e se manter atualizando-a indicou que ele apresenta menor potência instantânea quando o dispositivo móvel está configurado para trabalhar em uma frequência de 200 *MHz*. Porém, ao analisar-se o tempo de execução desse aplicativo para realizar-se a mesma tarefa, percebeu-se que houve diferença no tempo de execução. Assim, foi necessário fazer-se o cálculo da energia consumida, utilizando-se a metodologia descrita na Etapa 3.1.4, a fim de que se pudesse tomar uma decisão a respeito da frequência de operação ideal para esse aplicativo realizar essa atividade. Os tempos de execução estão descritos na Tabela 13.

Após ser feito o cálculo do consumo, conforme descrito na Etapa 3.1.4, foram obtidas as frequências ideal e aceitáveis mostradas no gráfico registrado na Figura 33.

Tabela 13 – Tempo de execução do Waze para cálculo de rota

Freq(MHz)	200	300	400	500	600	700	800	900	1000	1100	1200	1300	1400
Tempo(S)	9	7	8	6	8	8	7	7	6	8	7	8	7

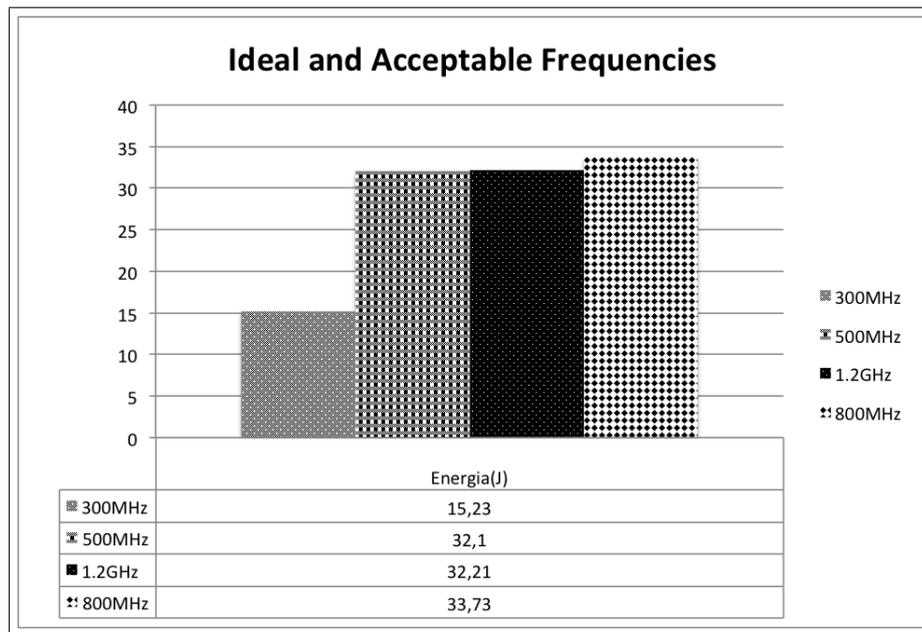


Figura 33 – Frequências ideal e aceitáveis para cálculo de rota usando Waze

Como se pode perceber no gráfico registrado na Figura 33, as frequências de 500 MHz, 800 MHz e 1.2 GHz possuem consumos próximos. Isso implica que essas frequências sejam consideradas como aceitáveis, pois, conforme explicado no Capítulo 3, precisava-se estabelecer, além de uma frequência ideal, frequências aceitáveis. Como frequência ideal considerou-se a de 300 MHz por possuir um consumo consideravelmente menor que todas as outras frequências: 15 Joules de diferença.

Dessa forma, viu-se a necessidade de fazer-se uma comparação estatística, utilizando-se o método descrito na Etapa 3.1.3 para poder-se afirmar qual das frequências acima mencionadas viria a ser a frequência ideal de operação, pois o resultado do consumo é um número aproximado. Os resultados dessa análise estão registrados na Tabela 14.

Tabela 14 – Comparação entre frequências em relação à potência instantânea do Waze para o cálculo de rota

Frequências(MHz)	300 e 500	300 e 800	500 e 1200
Resultados	p-value < 2.2e-16	p-value= 0.004361	p-value < 2.2e-16

A partir dessa análise, têm-se fortes indícios de que a frequência de 300 MHz é a ideal, pois, como pode ser visto na tabela acima, todos os testes feitos confirmam a

hipótese alternativa de que a amostra da frequência de 300 *MHz* é menor que as outras amostras das outras frequências que foram comparadas, e o grau de certeza é confirmado pelos valores baixos do valor-p nas respectivas comparações.

Comparando-se a abordagem proposta (criação de um **grupo** para representar a nevegação GPS) com outros *governors* presentes no *Kernel* do Android, chegou-se aos resultados registrados na Figura 34.

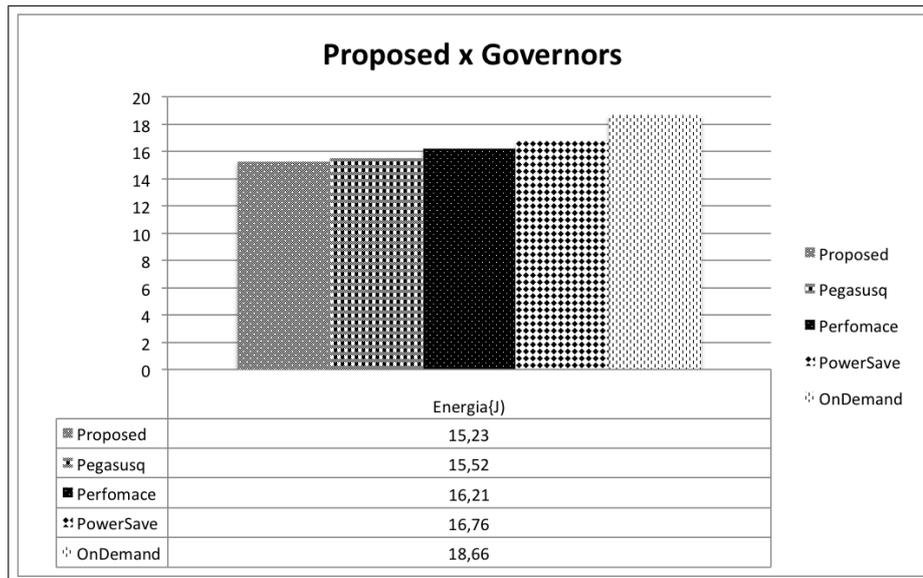


Figura 34 – Frequência ideal x governors padrão do Android para cálculo de rota usando Waze

Analisando-se o gráfico registrado na Figura 34, verifica-se que a abordagem proposta apresenta consumo próximo às estratégias **Pegasusq**, **Perfomance** e **PowerSave** presentes por padrão no *Kernel* Android. O método proposto apresenta economias que variam entre **2%** e **18%**. Isso indica uma tendência de economia maior a médio e longo prazos.

#### 4.4.4 Análise Waze: cálculo de rota e iteração

Para fazer-se o cálculo de rota e ainda se atender às requisições do usuário, a frequência ideal de acordo com a menor potência também seria de 200 *MHz*. Porém, novamente, o tempo de execução é bastante discrepante. Isso requer uma análise do consumo do aplicativo para decidir-se qual a melhor frequência. As frequências e seus respectivos tempos de execução estão descritos na Tabela 15.

Conforme se pôde verificar a partir da observação dos dados acima, fez-se necessária uma análise numérica, usando-se o método descrito na Etapa 3.1.4, com todas as frequências, a fim de decidir-se qual seria a frequência ideal. Após a realização da análise numérica, chegou-se aos resultados mostrados no gráfico registrado na Figura 35.

Tabela 15 – Tempo de execução do Waze para cálculo de rota e iteração

Freq(MHz)	200	300	400	500	600	700	800	900	1000	1100	1200	1300	1400
Tempo(S)	24	20	23	14	14	15	14	14	13	15	13	14	15

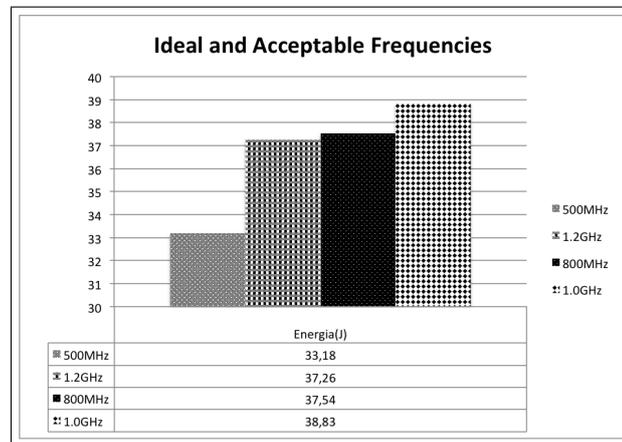


Figura 35 – Frequências ideal e aceitáveis para cálculo de rota com iteração usando Waze

Como se pode perceber no gráfico registrado na Figura 35, as frequências de 500 MHz, 800 MHz, 1.0 GHz e 1.2 GHz possuem consumos bastante próximos: apenas 6 Joules de diferença do menor para o maior consumo. Isso indica que, a curto prazo, pode-se adotar qualquer dessas frequências sem grande aumento no consumo energético do dispositivo.

Dessa forma, foi necessária uma comparação estatística, utilizando-se o método descrito na Etapa 3.1.3 para poder-se afirmar qual das frequências acima mencionadas viria a ser a frequência ideal de operação, pois o resultado do consumo é um número aproximado. Os resultados dessa análise estão mostrados na Tabela 16.

Tabela 16 – Comparação entre frequências em relação à potência instantânea do Waze para cálculo de rota e iteração

Frequências(MHz)	500 e 800	500 e 1000	500 e 1200
Resultados	p-value < 2.2e-16	p-value < 2.2e-16	p-value < 2.2e-16

A partir da observação dos resultados acima apresentados, vê-se que existem fortes indícios de que a frequência de 500 MHz pode ser considerada ideal, pois, como pode ser visto na tabela acima, todos os testes feitos confirmam a hipótese alternativa de que a amostra da frequência de 500 MHz é menor que as outras amostras das outras frequências comparadas, e o grau de certeza é confirmado pelos valores baixos do valor-p nas respectivas comparações.

Comparando-se a abordagem proposta com outros *governors* presentes no *Kernel*

do Android, chegou-se aos resultados mostrados na Figura 36.

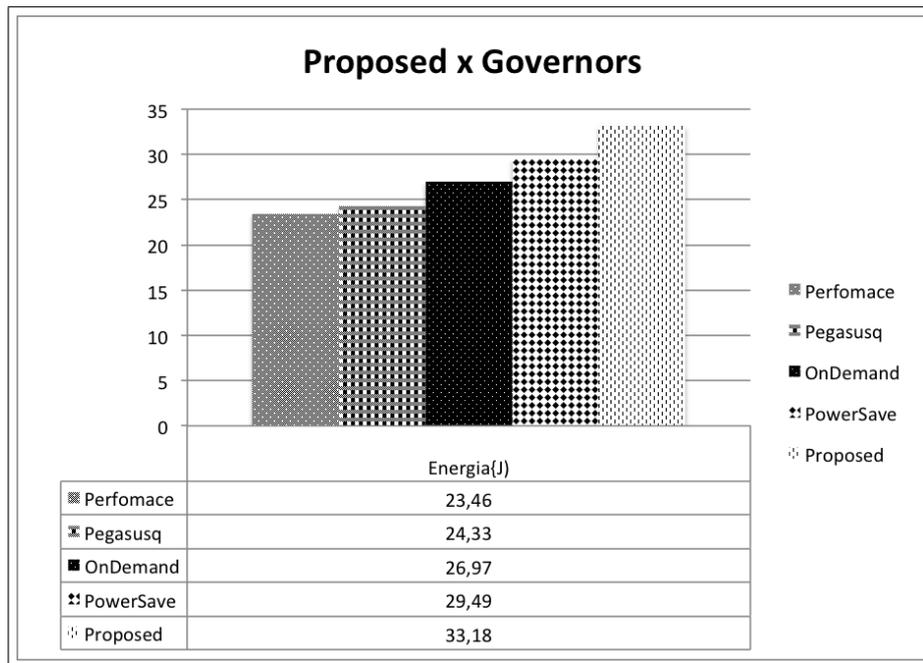


Figura 36 – Frequência ideal x governors padrão do Android para cálculo de rota com iteração usando Waze

Observando-se o gráfico registrado na Figura 36, percebe-se que a abordagem proposta apresenta consumo maior que as estratégias presentes por padrão no *Kernel* Android. O método proposto não apresenta economia a curto prazo, pois os tempos de experimento não foram suficientes para mostrar a economia gerada pela utilização do método. Chegou-se, apenas, a uma tendência de economia a médio e longo prazos, como foi o caso mostrado na Seção 4.2.

#### 4.4.5 Conclusão das Análises de Aplicativos de Navegação GPS

A partir das análises feitas e mostradas nos gráficos e das discussões realizadas nas subseções anteriores, é possível estabelecer uma faixa de frequência ideal para os aplicativos que possuem o propósito de realizar navegação *GPS*. Essa faixa, de acordo com a abordagem construída por este trabalho, deverá ser usada no ajuste de frequência enquanto o usuário estiver realizando tarefas que envolvam a navegação em mapas, a fim de atingir o objetivo de consumir menos energia durante a realização dessas atividades.

Viu-se que, ao detectar que o usuário está executando uma aplicação de navegação de mapas, o *smartphone* ajusta a frequência para a frequência ideal do aplicativo e, caso ele mude de aplicativo, o dispositivo móvel ajusta para a frequência ideal do grupo.

Os aplicativos apresentam as seguintes frequências ideais e aceitáveis:

- GMaps(apenas rota)
  - Frequência ideal: 1.1 *Ghz*
  - Frequências aceitáveis: 700 *Mhz*, 800 *Mhz*, 1.0 *Ghz*, 1.2 *Ghz*, 1.3 *Ghz*
- GMaps(rota e iteração)
  - Frequência ideal: 800 *Mhz*
  - Frequências aceitáveis: 700 *Mhz*, 900 *Mhz*, 1.0 *Ghz*, 1.2 *Ghz*
- Waze(apenas rota)
  - Frequência ideal: 300 *Mhz*
  - Frequências aceitáveis: 500 *Mhz*, 800 *Mhz*, 900 *Mhz*, 1.0 *Ghz*
- Waze(rota e iteração)
  - Frequência ideal: 500 *Mhz*
  - Frequências aceitáveis: 600 *Mhz*, 700 *Mhz*, 1.0 *Ghz*, 1.2 *Ghz*

A frequência ideal, de acordo com as análises, seria de 1.0 *Ghz* para essa categoria.

## 4.5 Análise de aplicativos de navegação de páginas

Para analisar o comportamento energético do aplicativo de exibição de páginas de internet padrão do Android, conhecido como Browser, recorreu-se às ideias discutidas no texto de Zhu *et al.* (ZHU; REDDI, 2013), que realizou uma avaliação energética das páginas Web de acordo com a complexidade de renderização delas, defendendo que cada *tag HTML* possui um custo energético de renderização.

Dessa forma, foram realizados experimentos para exibição de uma página complexa, composta por 1082 *tags HTML*, e uma página simples, composta por 125 *tags HTML*. Como meio de acesso à internet, foram utilizadas tanto redes de dados 3G como Wi-Fi. Os testes foram todos realizados colocando-se a página sob investigação como a inicial e mensurando-se o consumo do dispositivo móvel até o carregamento completo da página.

As páginas analisadas foram:

**Adobe** Página da Adobe, fabricante dos famosos *softwares* de animação gráfica para internet, como o **Flash**, e de edição fotográfica, como o **Photoshop** e o **Lightroom**. Essa página foi escolhida por possuir uma maior quantidade de tags e animações como conteúdo, tornando-se, assim, uma página com maior conteúdo para ser baixado e renderizado.

**Baidu** Sistema de buscas semelhante ao Google com apenas 125 *tags HTML* e *design* bastante limpo.

A contribuição energética de elementos invisíveis como *cookies* não foi contabilizada neste trabalho, pois, para isso seria necessário construir serviços especiais que realizassem o *download* desses elementos de forma isolada e fossem feitas medições de consumo para essa atividade.

#### 4.5.1 Análise Adobe: Wi-Fi

A análise realizada com o aplicativo Browser para exibir a página da Adobe, uma página com muito conteúdo multimídia, utilizando-se a rede Wi-Fi, revelou que ele apresenta menor potência instantânea, quando o dispositivo móvel está configurado para trabalhar em uma frequência de 200 *MHz*. Porém, ao analisar-se o tempo de execução desse aplicativo para exibir-se por completo a página, percebeu-se que houve diferença no tempo de execução. Assim, foi necessário fazer-se o cálculo da energia consumida, utilizando-se a metodologia descrita na Etapa 3.1.4, a fim de que se pudesse tomar uma decisão a respeito da frequência de operação ideal para esse aplicativo realizar essa atividade. Os tempos de execução estão descritos na Tabela 17.

Tabela 17 – Tempo de execução do Browser para exibir a página da Adobe

<b>Freq(MHz)</b>	200	300	400	500	600	700	800	900	1000	1100	1200	1300	1400
<b>Tempo(S)</b>	39	31	26	23	21	20	27	22	20	17	17	18	16

Após ser feito o cálculo do consumo, conforme descrito na Etapa 3.1.4, foram encontradas as frequências ideal e aceitáveis mostradas no gráfico registrado na Figura 37.

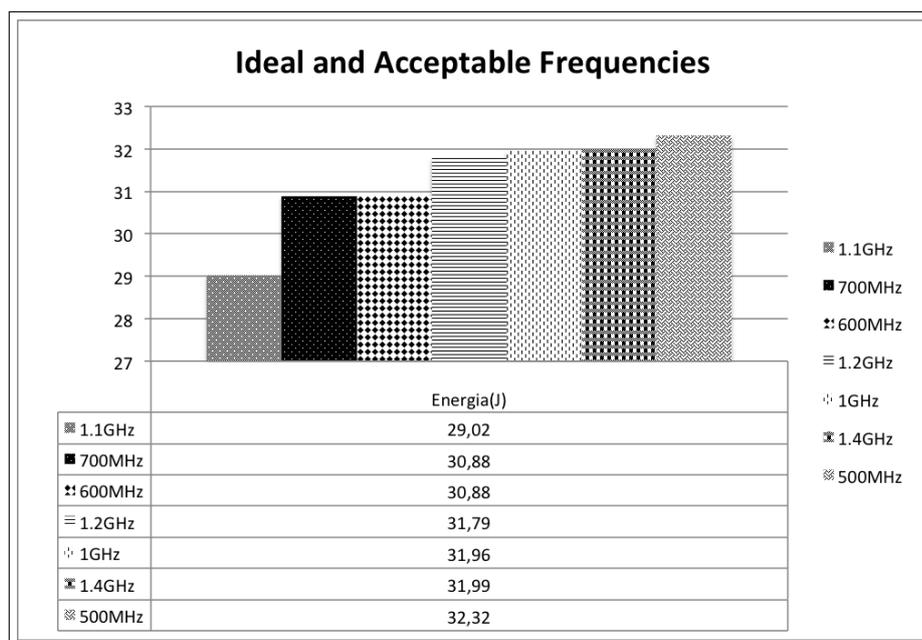


Figura 37 – Frequências ideal e aceitáveis para exibição da página da Adobe

Como se pode perceber no gráfico registrado na Figura 37, as frequências de 500 MHz, 600 MHz, 700 MHz, 1.0 GHz, 1.1 GHz, 1.2 GHz e 1.4 GHz possuem consumos bastante próximos: apenas 3 Joules de diferença do menor para o maior consumo. Isso indica que, a curto prazo, pode-se adotar qualquer uma dessas frequências sem grande aumento no consumo energético do dispositivo.

Dessa forma, foi necessário empreender uma comparação estatística, utilizando-se o método descrito na Etapa 3.1.3 para poder-se afirmar qual das frequências acima mencionadas viria a ser a frequência ideal de operação, pois o resultado do consumo foi um número aproximado. Os resultados dessa análise estão mostrados na Tabela 18.

Tabela 18 – Comparação entre frequências em relação à potência instantânea para exibição do site da Adobe na Wi-Fi

Frequências(MHz)	500 e 600	500 e 700	500 e 1000	500 e 1100	500 e 1200	500 e 1400
Resultados	p-value < 2.2e-16					

Considerando-se os dados acima, verifica-se que há fortes indícios de que a frequência de 500 MHz pode ser tida como a ideal, pois, como pode ser visto na tabela acima, todos os testes feitos corroboram a afirmação da hipótese alternativa de que a amostra da frequência de 500 MHz é menor que as outras amostras das outras frequências que foram comparadas, e o grau de certeza é confirmado pelos valores baixos do valor-p nas respectivas comparações.

Comparando-se a abordagem proposta (criação de um grupo para representar a

navegação Web) com outros *governors* presentes no *Kernel* do Android, chegou-se aos resultados registrados na Figura 38.

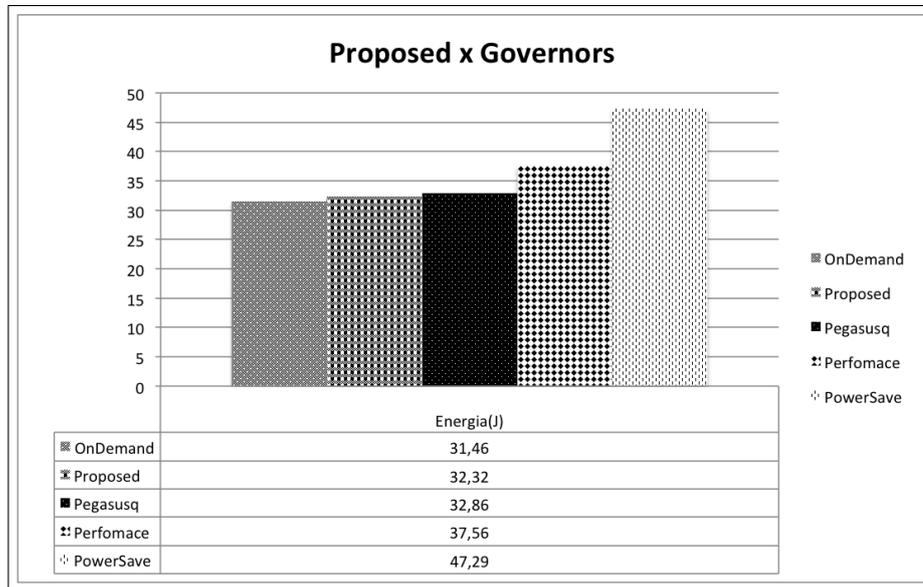


Figura 38 – Frequência ideal x governors padrão do Android para exibição do site Adobe usando WiFi

Percebe-se pelo gráfico registrado na Figura 38 que a frequência escolhida como ideal, embora tenha consumo próximo às abordagens **OnDemand** e **Pegasusq**, não possui consumo menor que a primeira. Assim, foi necessário recorrer-se ao fato discutido na Seção 1.2 e à análise estatística mostrada na Tabela 18 para verificarem-se economias apenas a médio e longo prazos, como mostrado na Seção 4.2, pois o tempo de análise não foi suficiente para verificar-se economia.

#### 4.5.2 Análise Adobe: 3G

Já para a conexão de dados *3G*, o aplicativo Browser apresenta um comportamento um pouco diferente para renderizar a página da Adobe, assim como acontece com os outros aplicativos, devido às características dessa conexão, que exige uma maior comunicação com a *RRC*. Após as análises estatística e numérica, conforme descrito na Etapa 3.1.3 e na Etapa 3.1.4, chegou-se à conclusão de que, nesse cenário, o aplicativo **Browser** apresenta as frequências ideal e aceitáveis para renderizar a página da Adobe mostradas na Figura 39.

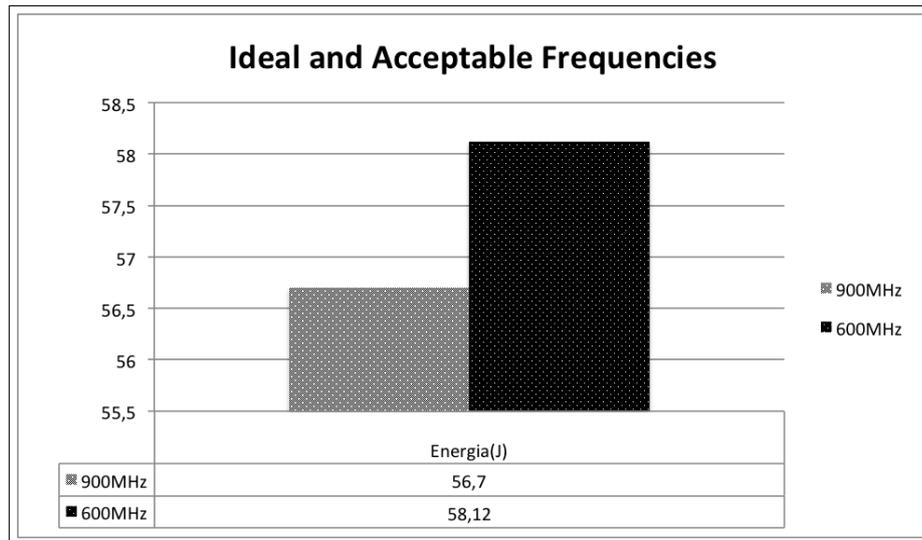


Figura 39 – Frequências ideal e aceitáveis para renderizar a página da Adobe usando 3G

Como se pode perceber no gráfico registrado na Figura 39, as frequências de 600 MHz e 900 MHz possuem consumos bastante próximos: apenas 2 Joules de diferença do menor para o maior consumo. Isso indica que, a curto prazo, pode-se adotar qualquer uma dessas frequências sem grande aumento no consumo energético do dispositivo.

Dessa forma, recorreu-se a uma comparação estatística, utilizando-se o método descrito na Etapa 3.1.3, para poder-se afirmar qual das frequências acima mencionadas viria a ser a frequência ideal de operação, pois o resultado do consumo foi um número aproximado. Os resultados dessa análise estão mostrados na Tabela 19.

Tabela 19 – Comparação entre frequências em relação à potência instantânea para exibição do site da Adobe na 3G

<b>Frequências(MHz)</b>	600 e 900
<b>Resultados</b>	p-value < 2.2e-16

Levando-se em conta os dados acima, viu-se que há fortes indícios de que a frequência de 600 MHz pode ser considerada como a ideal, pois, como pode ser visto na tabela acima, todos os testes feitos confirmam a hipótese alternativa de que a amostra da frequência de 600 MHz é menor que a amostra da frequência 900 MHz que foi comparada, e o grau de certeza é confirmado pelo valor baixo do valor-p na comparação realizada.

Comparando-se a abordagem proposta com outros *governors* presentes no *Kernel* do Android, chegou-se aos resultados mostrados na Figura 40.

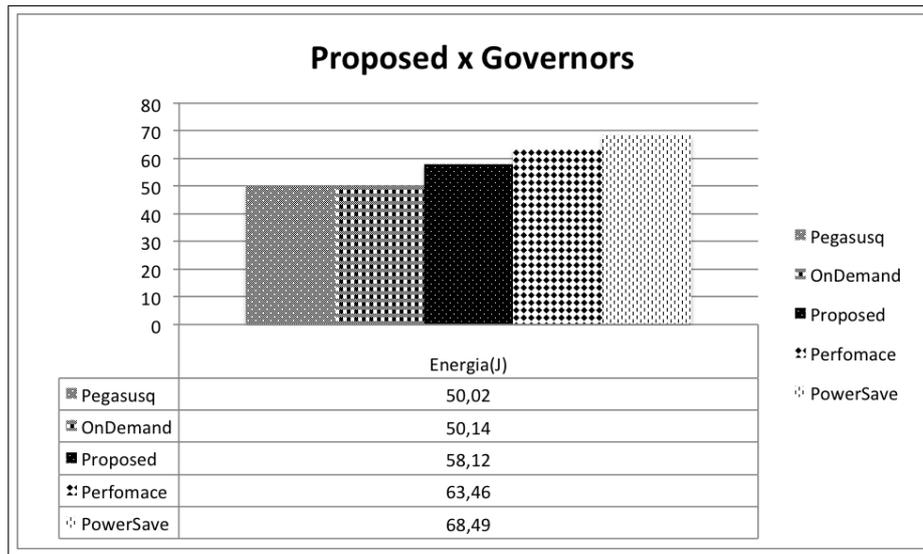


Figura 40 – Frequência ideal x governors padrão do Android para exibição do site Adobe usando 3G

Percebe-se, no gráfico registrado na Figura 40, que a frequência escolhida como ideal, embora tenha consumo próximo às abordagens **OnDemand** e **Pegasusq**, não possui consumo menor que as duas. Recorreu-se, então, às discussões apresentadas na Seção 1.2 e à análise estatística mostrada na Tabela 19 para verificarem-se economias apenas a médio e longo prazos, como mostrado na Seção 4.2, pois o tempo de análise não foi suficiente para se constatar economia.

#### 4.5.3 Análise Baidu: Wi-Fi e 3G

O Baidu é um mecanismo de busca que possui uma página semelhante à do Google. Dessa forma, a página dele é extremamente leve. Ela foi escolhida como contrapartida à página com mais recurso da Adobe por ser apontada no trabalho realizado por Zhu *et al.* (ZHU; REDDI, 2013) como uma das páginas de baixo consumo energético.

De acordo com a análise estatística descrita na Etapa 3.1.3, a frequência ideal é de 500 MHz. Porém, ao analisarem-se os tempos de execução, percebe-se que eles são muito pequenos e próximos, como mostrado na Tabela 20. Torna-se, assim, impossível determinar a melhor frequência com o método numérico usado neste trabalho, uma vez que se sabe haver um erro associado ao valor obtido do cálculo do consumo, embora não se saiba ainda majorar esse erro.

Tabela 20 – Tempo de execução do Browser para exibir página do Baidu: Wi-Fi

Freq(MHz)	200	300	400	500	600	700	800	900	1000	1100	1200	1300	1400
Tempo(S)	4	4	4	4	3	4	3	3	3	3	3	3	4

Dessa forma, chega-se à conclusão de que não existe frequência ideal para a exibição de páginas pequenas, usando-se a rede de dados Wi-Fi.

A partir da análise da energia gasta pelo aplicativo Browser para renderizar a página do Baidu, usando-se a rede de dados 3G, chegou-se à frequência de 300 *MHz* como sendo ideal. Porém, uma análise dos tempos de execução mostrou que novamente se têm tempos pequenos de execução, conforme mostrado na Tabela 21. E, como as potências instantâneas não são muito grandes, o consumo total do aplicativo Browser para renderizar a página do Baidu, usando-se a rede de dados 3G em cada uma das frequências, torna-se pequeno e muito próximo.

Tabela 21 – Tempo de execução do *Browser* para exibir página do Baidu: 3G

<b>Freq(MHz)</b>	200	300	400	500	600	700	800	900	1000	1100	1200	1300	1400
<b>Tempo(S)</b>	14	10	11	11	10	9	10	9	8	9	11	8	9

Tomando-se os dados acima, chega-se à conclusão de que não existe frequência ideal para renderização de páginas pequenas, como a do Baidu, usando-se a rede dados 3G.

#### 4.5.4 Conclusão das Análises para exibição de páginas no aplicativo Browser

A partir das análises feitas e mostradas nos respectivos gráficos e das discussões realizadas nas subseções anteriores, foi possível estabelecer uma faixa de frequência ideal para o aplicativo de navegação de páginas web Browser renderizar páginas complexas e simples.

O aplicativo possui as seguintes frequências ideais e aceitáveis para renderizar as páginas da Adobe e do Baidu:

- Adobe(Wi-Fi)
  - Frequência ideal: 500 *Mhz*.
  - Frequências aceitáveis: entre 600 *Mhz* e 700 *Mhz*, ou 1.0 *Ghz* e 1.4 *Ghz*.
- Adobe(3G)
  - Frequência ideal: 600 *Mhz*.
  - Frequência aceitável: 900 *Mhz*.
- Baidu(Wi-Fi)
  - Frequência ideal: não se tem uma frequência ideal.
- Baidu(3G)
  - Frequência ideal: não se tem uma frequência ideal.

A frequência ideal, de acordo com as análises, seria de 600 *MHz* para essa categoria na 3G e na Wi-Fi.

#### 4.5.5 Conclusão do mecanismo proposto

A partir das discussões feitas a respeito das frequências ideais e aceitáveis para cada um dos **grupos de aplicações** apresentados nas seções anteriores, chegou-se à conclusão de que os grupos de aplicações têm frequências ótimas de operação bastante próximas. Infere-se, então, que, ao se adotar esse mecanismo de **grupos de aplicações**, a oscilação na tensão de alimentação e na frequência terá uma baixa amplitude, fazendo que os picos de potência sejam menores e, conseqüentemente, resultem em um menor consumo energético por parte do *smartphone*. Essa conclusão está explicitada na Tabela 22.

Tabela 22 – Frequências ideais dos grupos de aplicações

Grupo	Apps	Freq Ideal(Mhz)	Freq Aceitáveis(Mhz)	Freq do Grupo
Video(Wi-Fi)	Youtube	700	200, 300,400, 1200	700 Mhz
	Dailymotion	200	300, 600,700, 1000	
	Vimeo	500	200,300, 400 ,600,700, 800	
Video(3G)	Youtube	300	200, 800	800Mhz
	Dailymotion	500	400 ,600,700, 800, 1000	
	Vimeo	500	200, 400 ,600, 800	
GPS(Rota)	Gmaps	1100	700, 800 ,1000, 1200,1300	1000Mhz
	Waze	300	500, 800 ,900, 1000	
GPS(Rota e Int.)	Gmaps	800	700, 900 ,1000, 1200	1000Mhz
	Waze	500	600,700, 1000, 1200	
Browser(Wi-Fi)	Adobe	500	600-700; 1000-1400	600Mhz
	Baidu	Não há	Não há	
Browser(3G)	Adobe	600	900	600Mhz
	Baidu	Não há	Não há	

Observando-se a Tabela 22, verifica-se que as frequências ótimas de operação dos grupos variam de 600 *Mhz* a 1.0 *Ghz*, resultando em uma amplitude de frequência ótima de operação de apenas 400 *Mhz*.

Levando em conta todo o estudo feito por meio dos experimentos e análises, foi possível chegar-se à conclusão de que poderia ser criado um mecanismo para economia baseado em grupos de aplicações, conforme o que se viu na Seção 1.2. Esse mecanismo foi implementado usando-se um serviço Android, como foi mencionado no Capítulo 3. A arquitetura desse mecanismo será discutida na próxima seção.

## 4.6 Arquitetura da Solução

A arquitetura da solução final está mostrada na Figura 41.

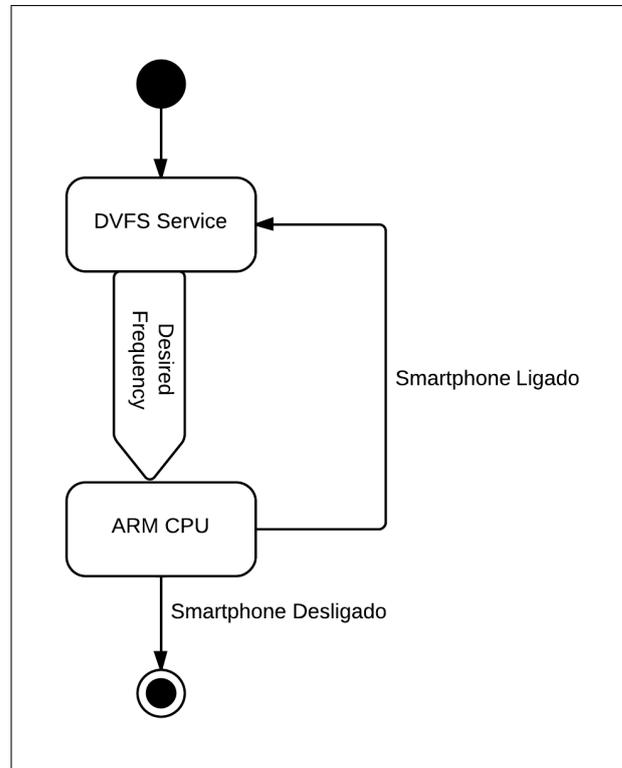


Figura 41 – Overview da solução final

Analisando-se a Figura 41, vê-se que o serviço de ajuste de frequência DVFS Service funciona integrado com o CPUFreq, que manda sinais para o processador continuamente até o desligamento do *Smartphone*, conforme descrito na Seção 2.9.

A arquitetura do serviço de ajuste de frequência, item **DVFS Service** da Figura 41, está registrada na Figura 42.

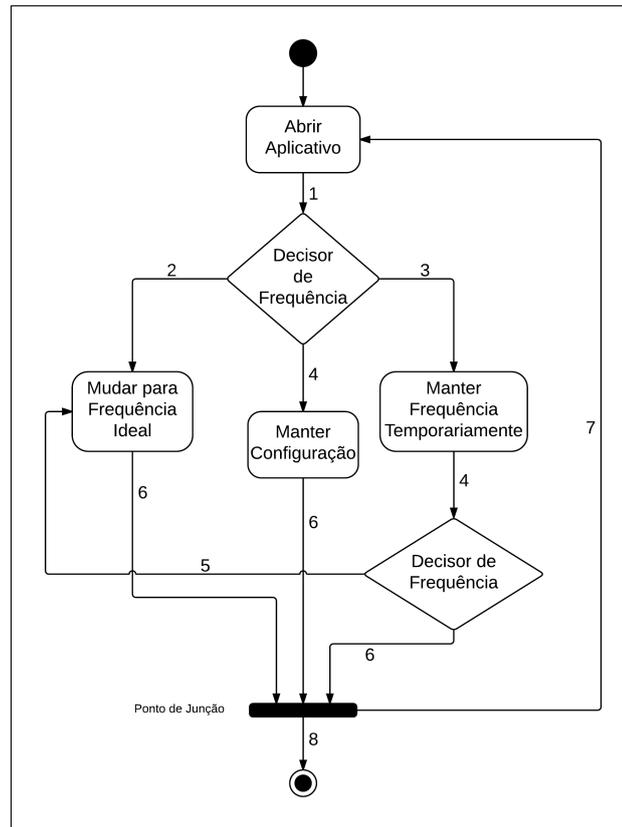


Figura 42 – Descrição da solução final

Observando-se a Figura 42, percebe-se que a arquitetura da solução final (serviço Android) é composta pelos seguintes estados:

**Abrir aplicativo:** O serviço começa a atuar quando um aplicativo é aberto e, conseqüentemente, posto em *foreground*, ou seja, colocado com aplicativo principal a ser utilizado pelo usuário do *smartphone*. Essa ação, mostrada na transição 1, faz o decisor de frequência atuar para seguir um dos fluxos mostrados pelas transições 2,3 ou 4.

**Decisor de frequência:** O papel do decisor de frequência é analisar a frequência em que o *smartphone* se encontra no momento para tomar a decisão de seguir um dos fluxos mostrados pelas transições 2,3 ou 4. Se a frequência de operação atual não for ideal nem aceitável para o aplicativo que acaba de ser aberto, o decisor segue o fluxo mostrado na transição 2. Caso contrário, ele segue os fluxos mostrados nas transições 3 ou 4.

**Mudar a frequência para ideal:** Ao tomar a transição 2, o **Decisor** decide mudar a frequência imediatamente, pois a mesma não é ideal nem aceitável para o aplicativo que acaba de ser aberto. Após realizar essa mudança, o **Decisor** toma a transição 6 e permanece inativo até a próxima abertura de aplicação.

**Manter configuração:** Ao tomar a transição **4**, o **Decisor** decide manter a frequência, pois a mesma já é ideal para o aplicativo que acaba de ser aberto. Após realizar essa mudança, o **Decisor** toma a transição **6** e permanece inativo até a próxima abertura de aplicação.

**Manter frequência temporariamente:** Ao tomar a transição **3**, o **Decisor** decide manter a frequência temporariamente, pois a mesma é uma frequência aceitável para o aplicativo em questão; isso faz que, a curto prazo, não haja um aumento de consumo considerado em relação ao realizado pela frequência ideal. Após um determinado tempo limite, o **Decisor** volta a atuar e, caso a aplicação continue aberta, o mesmo toma a transição **5** e muda a frequência de operação para a ideal imediatamente. Caso contrário, ele toma a transição **6** e permanece inativo até a próxima abertura de aplicação.

**Ponto de junção:** Este é um estado de inatividade no qual o serviço de ajuste de frequência permanece até a abertura da próxima aplicação ou o desligamento do *Smartphone*. Caso seja aberta uma nova aplicação, ele toma a transição **7**, retornando ao estado **Abrir aplicativo**. Caso o *Smartphone* seja desligado, ele toma a transição **8** e encerra.

Essa é a arquitetura da solução proposta. Percebe-se que, usando-se essa abordagem, atingem-se os objetivos descritos na Seção 1.1: redução da quantidade de chaveamentos de frequência e manutenção da *performance* do dispositivo móvel com uma tendência de redução do consumo energético do mesmo.

Após construir o serviço conforme mostrado na arquitetura acima, consegue-se atingir os objetivos discutidos na Seção 1.1, aproveitando as oportunidades de ganho de consumo mostradas na Seção 1.2 e, assim, conclui-se este trabalho de pesquisa. Após a construção deste serviço Android, foram observadas algumas limitações que serão discutidas no Capítulo 5.

## 5 Considerações Finais

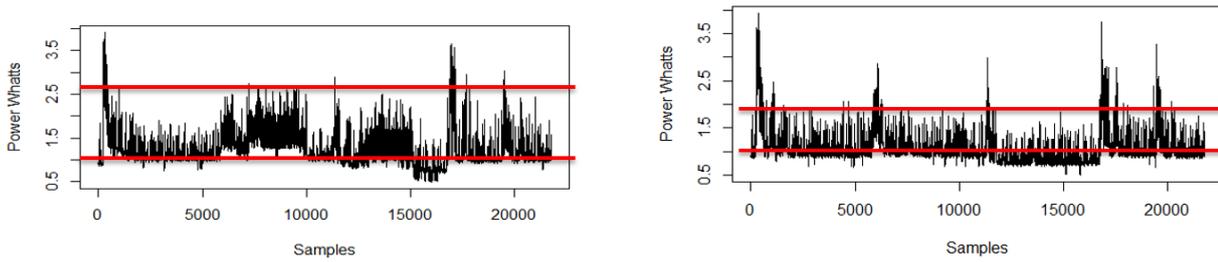
Como foi mencionado no Capítulo 3, este trabalho consiste em uma pesquisa experimental. Os experimentos apontaram o enorme consumo energético por parte não só das aplicações mais comumente usados no dia a dia da maioria dos usuários de *smartphones*, como também por parte do sistema operacional.

Levando-se, então, em consideração esse grande consumo energético, buscaram-se alternativas para minimizá-lo. A partir dessa busca, foram encontrados, por exemplo, os pacotes espúrios que são trocados frequentemente pelas aplicações que quase sempre fazem sincronização com a internet, como é o caso dos **Apps** Facebook, Gmail e WhatsApp, e a oscilação da frequência de operação do processador.

Verificou-se essa oscilação a partir do estudo do funcionamento dos circuitos modernos, **CMOS**. Descobriu-se, então, que o consumo desse tipo de circuito varia linearmente de acordo com a frequência de operação dele e é proporcional ao quadrado da tensão de alimentação, conforme discutido na Seção 1.2.

Após serem identificadas essas possibilidades de minimização do problema em enfoque, buscou-se entender o funcionamento do *kernel* Linux, no qual o Android, sistema operacional dos *smartphones* sob investigação, é baseado. Após a realização desse estudo, descobriu-se que havia um módulo dentro desse sistema operacional que possibilitava o controle da frequência de operação do processador, o **CPUFreq**, apresentado na Seção 2.9. Assim, resolveu-se investigar como se poderia usar esse módulo para reduzir o consumo energético por meio do chaveamento de tensão e frequência.

Durante essa investigação, após serem realizadas diversas coletas de potência usando-se o circuito descrito na Seção 3.1.1 e serem feitas diversas análises das séries de potência resultantes dessas coletas, identificou-se a oportunidade mostrada na Figura 43.



(a) Faixa de potência para um aplicativo com mudança de frequência

(b) Faixa de potência para um aplicativo sem mudança de frequência

Figura 43 – Comparação entre potências instantâneas com e sem mudança de frequência

Observando-se a Figura 43a, é possível perceber que as potências instantâneas de um aplicativo, quando há chaveamento frequência de operação do processador, ficam na faixa aproximada entre 1 *Watt* e 2,5 *Watts*. Já a partir da observação Figura 43b, é possível perceber que há uma diminuição considerável da faixa em que se encontram as potências instantâneas quando se mantém a frequência de operação do processador constante.

A partir dessa constatação, buscou-se estabelecer uma estratégia de economia que minimizasse a quantidade de chaveamentos. Estabeleceu-se, então, que grupos de aplicações seriam uma alternativa para alcançar-se esse objetivo. Para isso, viu-se que seria necessária uma análise a fim de decidir que **grupos** deveriam ser criados e que frequências seriam colocadas para cada um desses grupos.

Para realizar-se essa análise, recorreu-se a pesquisas referentes à redução do consumo energético a fim de identificar as técnicas que estavam sendo usadas para realizar a medição e análise das séries de potência dos dispositivos móveis de modo geral. Essa pesquisa foi feita para, com base nela, montar-se o plano experimental que guiaria todas as etapas de medição e análise de séries a serem usadas na realização deste trabalho.

Feita essa pesquisa, foram percebidas diversas lacunas, conforme discutido na Seção 2.1, que precisavam ser preenchidas para que se pudesse realizar a análise e, dessa forma, comparar as diversas abordagens possíveis a fim de identificar a melhor. Assim, para suprir essas lacunas, foi feito um estudo sobre técnicas estatísticas e numéricas que pudessem ser usadas, conforme foi mostrado nas seções 2.2, 2.3, 2.4, 2.5, 2.6 e 2.7.

Sabendo-se como mensurar e escolher a melhor abordagem, partiu-se para uma pesquisa a fim de conhecer os estudos mais relevantes sobre técnicas de otimização energética. Esse estudo foi apresentado na Seção 2.8, em que se pôde verificar que muitos estudiosos da área recorreram a técnicas de chaveamento de frequência e/ou ativação e desativação de dispositivos de acordo com a necessidade do usuário.

De posse desse conhecimento, pôde-se construir um plano experimental, feito con-

forme mostrado na Figura 44 e apresentado na Seção 3.1.

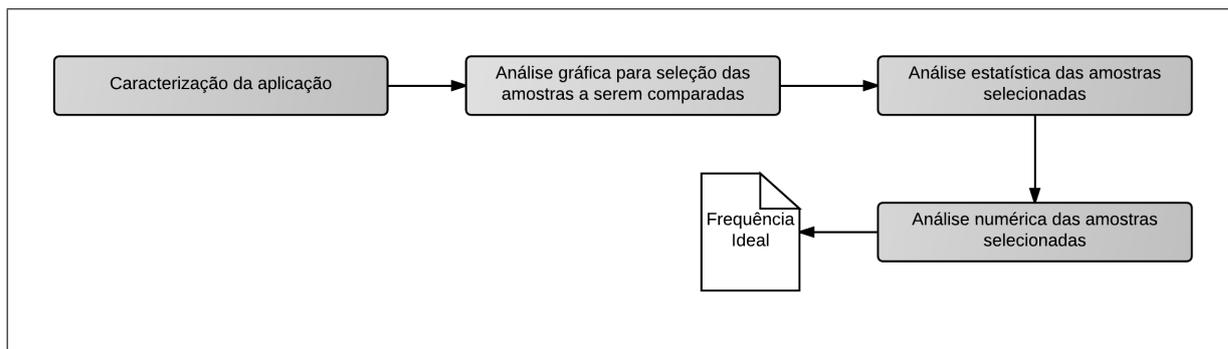


Figura 44 – Plano Experimental

A partir desse plano experimental, foram traçados os seguintes passos:

**Caracterização da aplicação:** Nesta etapa da pesquisa, foram criados os experimentos com três iterações em cada uma das frequências de operação do processador a ser analisado. Também foram realizadas as coletas e criados os arquivos com as potências instantâneas a serem analisadas, conforme foi descrito em 3.1.1.

**Análise gráfica para seleção das amostras a serem comparadas:** Nesta etapa, cada uma das três amostras coletadas na etapa anterior foi submetida a uma análise gráfica relativa aos seus respectivos consumos para serem identificadas possíveis discrepâncias entre essas iterações e para que se pudesse escolher as amostras a serem analisadas estatisticamente. Isso porque se pretendeu identificar a frequência de operação que levaria a um menor consumo para a realização da atividade em questão, conforme foi discutido em 3.1.2.

**Análise estatística das amostras selecionadas:** Nesta etapa, fez-se a análise estatística que permitiu indicar qual a melhor frequência de operação visando ao baixo consumo energético para a realização da atividade em questão pela **App** estudada, conforme foi exposto na Seção 3.1.3.

**Análise numérica das amostra selecionadas:** Após obter-se a informação sobre frequências que tenderiam a um menor consumo para a execução de uma determinada **App**, passou-se a analisar os tempos de execução da aplicação em estudo para verificar-se se o fato de adotar uma determinada frequência aumentaria de forma significativa o tempo de execução. De posse dessa informação, foi possível calcular a energia consumida pela **App** em uma determinada frequência e assim obter uma maior acurácia na escolha da frequência ideal. Nessa etapa, também foram elencadas frequências que possuem consumo próximo ao da frequência ideal, de acordo com o que foi discutido em 3.1.4, pois, conforme registrado, buscava-se construir um mecanismo de otimização baseado em **grupos** de aplicações.

Após a definição do plano experimental, construiu-se um ambiente, conforme descrito na Seção 4.1, onde esse plano deveria ser posto em prática. O ambiente escolhido para a realização dos experimentos foi, então, composto pelos seguintes elementos:

**Computador:** Responsável por processar os dados coletados pelo circuito de medição e gravá-los em um arquivo para ser futuramente analisado pelos procedimentos de análise estatística e numérica.

**Circuito de Medição:** Responsável por realizar as coletas de potências instantâneas do *smartphone* escolhido para a realização deste trabalho, Galaxy SIII, durante a realização das atividades estudadas.

**Fonte de Alimentação:** Responsável por fornecer uma tensão de 4,4V constante ao *smartphone*.

**Smartphone:** O escolhido para a realização dos experimentos realizados neste trabalho foi um Samsung Galaxy SIII, com 16Gb de memória interna, modelo GT-I9300, tela super AMOLED de 16 milhões de cores, Sistema Operacional Cyanogen Mod 10.1, processador Cortex A9 Quad-Core e 1Gb de memória Ram. Esse *smartphone* possui 13 frequências configuráveis.

Depois de definido o plano experimental e montado o ambiente em que os experimentos seriam feitos, foram estabelecidos os **grupos** de aplicações a serem analisados, a saber: a) *stream* de vídeo; b) navegação GPS; e c) navegação Web. Dentro dos grupos estabelecidos, selecionaram-se as seguintes aplicações:

**Stream de vídeo:** Youtube, Dailymotion, Vimeo.

**Navegação GPS:** Google Maps, Waze.

**Navegação Web:** Browser

Esses **grupos** e suas respectivas aplicações foram escolhidos pelos motivos explicados na Seção 3.1.

A realização dos experimentos mostrou que a utilização do mecanismo proposto nesta pesquisa pode levar a uma economia da ordem de 50% em certos casos, conforme foi discutido no Capítulo 4. Com a execução dos experimentos, chegou-se às frequências ideais mostradas na Tabela 22.

Diante desses resultados, foi possível, então, elaborar a arquitetura do mecanismo que faria a configuração da frequência ideal de acordo com a aplicação que estava em execução e com a tecnologia de comunicação que estava sendo usada (**Wi-Fi** ou **3G**), conforme foi mostrado na Figura 41 e na Figura 42.

Ao usar-se o serviço Android descrito na Seção 4.6, percebeu-se que nem sempre se chegava à economia encontrada durante a realização dos experimentos de forma isolada. Empreendeu-se, então, uma pesquisa para obter-se informações a respeito dos fatores que estariam impedindo que se chegasse aos resultados esperados. Essa busca requereu que se recorresse aos trabalhos sobre otimização energética que consideram o uso de **contexto** para atingir seus objetivos.

Shye *et al.* (SHYE; SCHOLBROCK; MEMIK, 2009) mostram que, para promover-se uma otimização energética efetiva, faz-se necessário não apenas estudar os níveis de circuito, arquitetura e sistema, mas também o comportamento dos usuários finais. Eles pontuam que o principal *workload* do sistema é o usuário final, pois as arquiteturas móveis executam aplicações que lidam diretamente com ele (o usuário final), uma vez que a execução de *jobs* em lote e serviços de execução longa são minimizados ou até mesmo não permitidos, como é o caso do **iPhone**. Desse modo, os arquitetos devem tratar os usuários finais como *workload* e estudar tendências, propriedades e padrões de uso desses usuários, pois, sem entender o comportamento deles, não é possível compreender o impacto de qualquer otimização energética realizada.

Já Bonetto *et al.* (BONETTO *et al.*, 2012) mostram que é impossível desconsiderar as condições do ambiente em que o *smartphone* está inserido e o estado interno dele (contexto), quando se fala de redução do consumo energético. Sabendo-se do contexto no qual dispositivo móvel está inserido, é possível economizar tanto mais energia quanto possível, otimizando, por exemplo, a qualidade da chamada de voz, a fim de consumir menos energia. Os dispositivos capazes de adaptar-se às diferentes situações em tempo de execução são chamados de autocientes ou cientes do contexto. Esses autores propõem, então, a criação de um componente responsável pela geração de um modelo energético capaz de avaliar o consumo energético do sistema em um determinado estado, sendo constantemente refinado, usando as informações obtidas dos hábitos do usuário e da experiência coletada no dispositivo. Os dados de consumo reais dos dispositivos são enviados para um servidor onde são analisados. Assim que um dispositivo móvel é registrado no serviço, ele é equipado com um **modelo de consumo genérico**, que é refinado usando os dados coletados do dispositivo específico. Esses modelos são construídos e refinados em um servidor externo que armazena os dados previamente obtidos. Com o modelo gerado pelo servidor no dispositivo, é possível saber-se o consumo energético atual do sistema e a expectativa de tempo de vida da bateria a partir das condições atuais.

As investigações sobre o uso de **contexo** permitem verificar que tipos de informações podem ser usadas para a tomada de decisão, dependendo do ambiente em que o dispositivo está em funcionamento, a saber:

- a) localização, co-localização e localizações relacionadas;
- b) identidade do usuário ou das pessoas co-localizadas;
- c) atividade;
- d) tempo;
- e) níveis de som;
- f) níveis de luz;
- g) movimento

Chalmers(CHALMERS, 2011) mostra que **contexto** são as circunstâncias relevantes para a interação entre um usuário e o ambiente computacional delas. A partir disso, veem-se possibilidades de melhorar a abordagem adotada neste trabalho, tornando-a mais inteligente, de modo que o serviço desenvolvido a partir dos resultados obtidos durante a realização desta pesquisa possa decidir sobre: a) o brilho de tela adequado; b) a conexão de dados a ser usada; c) a frequência de operação adequada para a aplicação em uso; e d) os dispositivos a serem ligados/desligados em cada uma das situações do dia a dia do usuário.

Percebeu-se, também, a partir desses estudos, que uma outra forma de otimizar o consumo do *smartphone* é por meio da construção de um modelo energético capaz de avaliar o consumo energético do sistema em um determinado estado e, para isso, têm-se usado redes radiais *Wavelet*, pois elas possibilitam regressão não paramétrica, como é mostrado em Holmes *et al.*(HOLMES; MALLICK, 2000), que analisam o desempenho desse tipo de redes dentro de uma estrutura *baesiana*. Assim, são obtidas distribuições de probabilidade tanto sobre a dimensão das redes quanto sobre os coeficientes da rede, colocando-se uma prévia sobre os graus de liberdade do modelo.

Um outra abordagem é discutida por Cristea *et al.*(CRISTEA; TUDUCE; CRISTEA, 2000), em que são apresentados métodos de aproximação de funções não lineares para estimação do modelo livre de sistemas por meio de Redes Neurais Artificiais, pois as características dessas redes - como o paralelismo maciço, a distribuição de computação entre elementos de processamento e, especialmente, a capacidade de aprender a partir de exemplos e de autoadaptação - são muito adequados para a abordagem de multirresolução.

Diante disso, considera-se necessário dar continuidade às pesquisas iniciadas neste trabalho a fim de enriquecer a arquitetura de solução proposta na Seção 4.6, colocando-se elementos, como um *framework* contextual no **Decisor**, mostrado na Figura 42, para proporcionar maior inteligência e poder de adaptação ao *smatrphone* com elementos contextuais, como localização, condições de iluminação do ambiente em que que o usuário se

encontra e atividade que o usuário está executando a fim de, assim, construir identidades e poder configurar os dispositivos do *smartphone* de modo a equilibrar da melhor forma o consumo energético e a *performance* requerida pelo usuário. Além disso, pode-se modelar as séries estudadas neste trabalho com o propósito não só de mensurar o erro associado no cálculo do consumo como também de realizar previsões de economias a médio e longo prazos, a fim de aplicar as melhores configurações para atingir-se esse objetivo.

Espera-se, portanto, que uma pesquisa doutoral, por exemplo, possa aprimorar a abordagem feita por este trabalho e, assim, contribuir para uma redução maior no consumo energético de um *smartphone* por prever configurações mais adequadas a cada uma das situações do dia a dia da maioria dos usuários e, dessa forma, aumentar o tempo de disponibilidade desses dispositivos que estão a cada dia sendo usados com maior frequência em uma quantidade maior de atividades.



## Referências

- ADAFRUIT INDUSTRIES. *Adafruit INA219 Current Sensor Breakout*. New York, 2014. 16 p. Disponível em: <<http://goo.gl/7pTeVb>>. Acesso em: 20.01.2014.
- ALAWNAH, S.; SAGAHYROON, A. Modeling smartphones power. In: *EUROCON, 2013 IEEE*. IEEE, 2013. p. 369–374. Disponível em: <<http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6625010>>. Acesso em: 19.1.2014.
- ARENA, G. *Samsung I9300 Galaxy SIII Specs*. 2014. Samsung I9300 Galaxy SIII Specs. Disponível em: <<http://goo.gl/oW5GhL>>. Acesso em: 01.02.2014.
- ASHWINI, H. S.; THAWANI, A.; SRIKANT, Y. N. Middleware for efficient power management in mobile devices. In: *Proceedings of the 3rd International Conference on Mobile Technology, Applications & Systems*. IEEE, 2006. Disponível em: <<http://dl.acm.org/citation.cfm?id=1292387>>. Acesso em: 19.1.2014.
- ATMEL CORPORATION. *ATmega48A/PA/88A/PA/168A/PA/328/P [DATASHEET]*. San Jose, 2009. 39 p. Revision 8271A-12/09. Disponível em: <<http://goo.gl/dvASr9>>. Acesso em: 20.01.2014.
- BALASUBRAMANIAN, N.; BALASUBRAMANIAN, A.; VENKATARAMANI, A. Energy consumption in mobile phones: a measurement study and implications for network applications. In: *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference*. ACM, 2009. p. 280–293. Disponível em: <<http://dl.acm.org/citation.cfm?id=1644927>>. Acesso em: 19.1.2014.
- BESSA, I. *Relatório Google sobre o uso de smartphones no Brasil*. 2012. Relatório Google sobre o uso de smartphones no Brasil. Disponível em: <<http://goo.gl/GO63P7>>. Acesso em: 10.01.2014.
- BONETTO, A. et al. Mpower: Towards an adaptive power management system for mobile devices. In: *Computational Science and Engineering (CSE), 2012 IEEE 15th International Conference on*. ACM, 2012. p. 318–325. Disponível em: <<http://doi.acm.org/10.1145/1669112.1669135>>. Acesso em: 19.1.2014.
- BUSSAB, W.; MORETTIN, P. *Estatística básica*. [S.l.]: Saraiva, 2010. ISBN 9788502081772.
- CHALMERS, D. *Sensing and Systems in Pervasive Computing: Engineering Context Aware Systems*. [S.l.]: Springer London, 2011. (Undergraduate Topics in Computer Science). ISBN 9780857298409.
- CONGRESS, G. C. *PRTM Analysis Finds Li-ion Battery Overcapacity Estimates Largely Unfounded, with Potential Shortfalls Looming; Total Market Demand in 2020 Will Require 4x Capacity Announced To Date*. 2010. Green Car Congress. Disponível em: <<http://goo.gl/z75ImT>>. Acesso em: 10.01.2014.
- CRISTEA, P.; TUDUCE, R.; CRISTEA, A. Time series prediction with wavelet neural networks. In: *Neural Network Applications in Electrical Engineering, 2000*.

- NEUREL 2000. *Proceedings of the 5th Seminar on*. ACM, 2000. p. 5–10. Disponível em: <<http://goo.gl/vc2u4Y>>. Acesso em: 19.1.2014.
- DALLAL, G. E.; WILKINSON, L. An analytic approximation to the distribution of lilliefors's test statistic for normality. *The American Statistician*, Taylor & Francis Group, v. 40, n. 4, p. 294–296, 1986.
- DATTA, S.; BONNET, C.; NIKAEIN, N. Android power management: Current and future trends. In: *Enabling Technologies for Smartphone and Internet of Things (ETSIoT), 2012 First IEEE Workshop on*. IEEE, 2012. p. 48–53. Disponível em: <<http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6311253>>. Acesso em: 19.1.2014.
- DEVELOPERS, A. *Service*. 2014. Service. Disponível em: <<http://goo.gl/GVPfVe>>. Acesso em: 05.02.2014.
- DEVELOPERS, X. *Android CPU governors explained*. 2012. Android CPU governors explained. Disponível em: <<http://goo.gl/Z0LWXB>>. Acesso em: 05.02.2014.
- GIL, A. *Como elaborar projetos de pesquisa*. [S.l.]: Atlas, 2010. ISBN 9788522458233.
- GUIDORIZZI, H. *Um curso de cálculo*. [S.l.]: LTC, 2001. (Um curso de cálculo, v. 1). ISBN 9788521612599.
- GUPTA, P. et al. Measurement of power consumption in multimedia mobiles for various network activities in 2g/3g networks. In: *Advanced Networks and Telecommunications Systems (ANTS), 2012 IEEE International Conference on*. IEEE, 2012. p. 141–144. Disponível em: <<http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6524245>>. Acesso em: 19.1.2014.
- HARJULA, E.; KASSINEN, O.; YLIANTTILA, M. Energy consumption model for mobile devices in 3g and wlan networks. In: *Consumer Communications and Networking Conference (CCNC), 2012 IEEE*. IEEE, 2012. p. 532–537. Disponível em: <<http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6181134>>. Acesso em: 19.1.2014.
- HENNESSY, J.; PATTERSON, D.; ASANOVIĆ, K. *Computer Architecture: A Quantitative Approach*. [S.l.]: Morgan Kaufmann/Elsevier, 2012. (Computer Architecture: A Quantitative Approach). ISBN 9780123838728.
- HOLMES, C.; MALLICK, B. Bayesian wavelet networks for nonparametric regression. *Neural Networks, IEEE Transactions on*, v. 11, n. 1, p. 27–35, Jan 2000. ISSN 1045-9227.
- HUANG, C.; YANG, H.-T. Evolving wavelet-based networks for short-term load forecasting. In: *Generation, Transmission and Distribution, IEE Proceedings-IET*, 2013. p. 222–228. Disponível em: <[http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=926421](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=926421)>. Acesso em: 19.1.2014.
- ICEL MANAUS. *MANUAL DE INSTRUÇÕES DA FONTE DIGITAL MODELO PS-1500*. Manaus, 2014. 11 p. Disponível em: <<http://goo.gl/GluztS>>. Acesso em: 10.02.2014.

KALIC, G.; BOJIC, I.; KUSEK, M. Energy consumption in android phones when using wireless communication technologies. In: LIN, F.; SATTLER, U. (Ed.). *MIPRO, 2012 Proceedings of the 35th International Convention*. IEEE, 2012. p. 754–759. Disponível em: <<http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6240745>>. Acesso em: 19.1.2014.

KERNEL.ORG. *Documentation about CPUFreq*. 2013. Documentation about CPUFreq. Disponível em: <<http://goo.gl/pEBPBZ>>. Acesso em: 05.02.2014.

KHAIRY, A.; AMMAR, H.; BAHGAT, R. Smartphone energizer: Extending smartphone's battery life with smart offloading. In: *Wireless Communications and Mobile Computing Conference (IWCMC), 2013 9th International*. IEEE, 2013. p. 329–336. Disponível em: <[http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=6583581](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6583581)>. Acesso em: 19.1.2014.

KWON, K.; CHAE, S.; WOO, K.-G. An application-level energy-efficient scheduling for dynamic voltage and frequency scaling. In: *Consumer Electronics (ICCE), 2013 IEEE International Conference on*. IEEE, 2013. p. 3–6. Disponível em: <[http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=6486770](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=6486770)>. Acesso em: 19.1.2014.

LIANG, Y.; LAI, P.; CHIOU, C. An energy conservation dvfs algorithm for the android operating system. *Journal of Convergence*, v. 1, n. 1, 2010.

LIMA, E. *Curso de análise*. [S.l.]: Instituto de Matemática Pura e Aplicada, CNPq, 1992. (Projeto Euclides, v. 1).

MITTAL, T.; SINGHAL, L.; SETHIA, D. Optimized cpu frequency scaling on android devices based on foreground running application. In: CHAKI, N.; MEGHANATHAN, N.; NAGAMALAI, D. (Ed.). *Computer Networks & Communications (NetCom)*. New York: Springer New York, 2013. v. 131, p. 827–834. Disponível em: <[http://dx.doi.org/10.1007/978-1-4614-6154-8\\_80](http://dx.doi.org/10.1007/978-1-4614-6154-8_80)>. Acesso em: 20.01.2014.

MORA, J.; LEU, J.-S. Intelligent power saving technique for mobile devices. In: *Communications (APCC), 2012 18th Asia-Pacific Conference on*. IEEE, 2012. p. 504–508. Disponível em: <<http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=6362319>>. Acesso em: 19.1.2014.

NAGATA, K.; YAMAGUCHI, S.; OGAWA, H. A power saving method with consideration of performance in android terminals. In: *Ubiquitous Intelligence & Computing and 9th International Conference on Autonomic & Trusted Computing (UIC/ATC), 2012*. IEEE, 2012. p. 578–585. Disponível em: <<http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=6332050>>. Acesso em: 19.1.2014.

NIELSEN. *APP-HAPPY WITH ANDROID THE MOST POPULAR ANDROID APPS BY AGE*. 2011. APP-HAPPY WITH ANDROID THE MOST POPULAR ANDROID APPS BY AGE. Disponível em: <<http://goo.gl/CzqsRG>>. Acesso em: 05.02.2014.

PARADISO, J. A.; STARNER, T. Energy scavenging for mobile and wireless electronics. *IEEE Pervasive Computing*, IEEE Educational Activities Department, Piscataway, NJ, USA, v. 4, n. 1, p. 18–27, jan. 2005. ISSN 1536-1268. Disponível em: <<http://dx.doi.org/10.1109/MPRV.2005.9>>.

- RPROJECT. *nortest: Tests for Normality*. Manaus, 2012. 10 p. Disponível em: <<http://goo.gl/d8FbgE>>. Acesso em: 10.02.2014.
- SHYE, A.; SCHOLBROCK, B.; MEMIK, G. Into the wild: Studying real user activity patterns to guide power optimizations for mobile architectures. In: *Proceedings of the 42Nd Annual IEEE/ACM International Symposium on Microarchitecture*. ACM, 2009. p. 168–178. Disponível em: <<http://doi.acm.org/10.1145/1669112.1669135>>. Acesso em: 19.1.2014.
- SIEGEL, S.; CASTELLAN, N. *Estatística não paramétrica para ciências do comportamento*. [S.l.]: Artmed, 2006. (Métodos de pesquisa). ISBN 9788536307299.
- SKLAVOS, N.; TOULIOU, K. A system-level analysis of power consumption & optimizations in 3g mobile devices. In: LABIOD, H.; BADRA, M. (Ed.). *New Technologies, Mobility and Security*. Netherlands: Springer Netherlands, 2010. p. 217–227. Disponível em: <[http://dx.doi.org/10.1007/978-1-4020-6270-4\\_19](http://dx.doi.org/10.1007/978-1-4020-6270-4_19)>. Acesso em: 20.01.2014.
- SUPER!MONITORING. *Infographic: State of mobile 2013*. 2013. Infographic: State of mobile 2013. Disponível em: <<http://goo.gl/lM5nmd>>. Acesso em: 05.09.2014.
- TEXAS INSTRUMENTS. *INA 219: Zero-Drift, Bi-Directional Current/Power Monitor with I2C Interface*. Dallas, 2011. 40 p. Disponível em: <<http://goo.gl/031Ega>>. Acesso em: 20.01.2014.
- THIAGARAJAN, N. et al. Who killed my battery?: analyzing mobile browser energy consumption. In: *Proceedings of the 21st international conference on World Wide Web*. ACM, 2012. p. 41–50. Disponível em: <<http://dl.acm.org/citation.cfm?id=2187843>>. Acesso em: 19.1.2014.
- WAZLAWICK, R. *Metodologia de Pesquisa para Ciência da Computação*. [S.l.]: Elsevier, 2009. ISBN 9788535266436.
- XIAO, Y.; KALYANARAMAN, R. S.; YLA-JAASKI, A. Energy consumption of mobile youtube: Quantitative measurement and analysis. In: *Next Generation Mobile Applications, Services and Technologies, 2008. NGMAST'08. The Second International Conference on*. IEEE, 2008. p. 61–69. Disponível em: <[http://ieeexplore.ieee.org/xpls/abs\\_all.jsp?arnumber=4756414](http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=4756414)>. Acesso em: 19.1.2014.
- ZHU, Y.; REDDI, V. J. High-performance and energy-efficient mobile web browsing on big/little systems. In: *Proceedings of the 2013 IEEE 19th International Symposium on High Performance Computer Architecture (HPCA)*. ACM, 2013. p. 13–24. Disponível em: <<http://dl.acm.org/citation.cfm?id=2495486>>. Acesso em: 19.1.2014.