



Universidade do Estado da Bahia  
Departamento de Ciências Exatas e da Terra  
Bacharelado em Sistemas de Informação

# COMPARANDO MODELOS PARA PREVISÃO EM SÉRIES TEMPORAIS APLICADOS NOS DISPOSITIVOS MÓVEIS

Raul Cezar Costa de Abreu

Salvador/BA

2013

**Raul Cezar Costa de Abreu**

**Comparando modelos para previsão em séries  
temporais aplicados nos dispositivos móveis**

Monografia apresentada à Banca Examinadora como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação pela Universidade do Estado da Bahia.

**Orientador: Murilo do Carmo Boratto**

**Salvador/BA**

**2013**

**Raul Cezar Costa de Abreu**

**Comparando modelos para previsão em séries  
temporais aplicados nos dispositivos móveis**

Monografia apresentada à Banca Examinadora como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação pela Universidade do Estado da Bahia.

Aprovada em: \_\_\_\_\_ / \_\_\_\_\_ / \_\_\_\_\_

**BANCA EXAMINADORA**

---

**Murilo do Carmo Boratto**

Doutorando em Matemática Aplicada as Ciências e as Engenharias  
Universidade do Estado da Bahia

---

**Trícia Souto Santos**

Mestre em Informática  
Universidade do Estado da Bahia

---

**Ivan Lessa**

Mestrando em Sistemas e Computação  
Unifacs

# Agradecimentos

Inicialmente agradeço a todos meus colegas e professores que me acompanharam neste caminho até a conclusão deste curso na Uneb.

Agradeço imensamente à minha família pela educação, moral, confiança e amor acima de tudo.

O maior agradecimento vai para meus melhores amigos, Bruno Vinícius, Felipe Piñeiro e Juliana Farjadini que desde o primeiro trabalho na Uneb formaram uma equipe para a vida toda, sem eles não teria passado em diversas matérias e principalmente no trabalho final, sem eles não teria experimentado minha futura profissão: Empreendedor. Nossa grande amizade será nossa força em nossas próximas jornadas!

*”Todas as ideologias são idiotas, seja religiosa ou política, pois é o pensamento conceitual, a palavra conceitual, que possui o infeliz e dividido homem.”*

*Jiddu Krishnamurti*

# Resumo

Atualmente nos dispositivos móveis percebe-se uma escassa oferta de energia, além de demandas cada vez maiores por parte dos diversos componentes de *hardware* e *software* embarcados. Estes componentes internos evoluíram vertiginosamente com exceção das baterias, que freiam o avanço tecnológico destes aparelhos. Devido a diversidade de ambientes e usuários, soluções que visam diminuir a demanda energética nestes dispositivos precisam ser adaptáveis através do reconhecimento e previsão do padrão de uso. Modelos de séries temporais são ferramentas clássicas para solucionar problemas que envolvem previsão, porém apesar de ser uma área bastante estudada, são poucos os trabalhos relacionados à dispositivos móveis. Devido a diversidade de métodos para previsão de séries temporais e a escassez de trabalhos semelhantes, este trabalho propõe a comparação de métodos clássicos para previsão em séries temporais executando em dispositivos móveis. Estes testes comparativos demonstraram que os métodos propostos são aplicáveis computacionalmente nos dispositivos móveis, porém devido a natureza não estacionária de algumas séries ou períodos destas, em termos de precisão somente 27,5% das previsões tiveram erro abaixo dos 120 minutos, mostrando que estes métodos de previsão mais simplórios podem não servir para a maioria das soluções relacionadas a otimização energética e padrão de uso em dispositivos móveis.

**Palavras chave:** Energia, dispositivos móveis, smartphones, android, séries temporais, previsão.

# Abstract

Today in mobile devices there is perceived a scarce supply of energy and increasing demands by the various embedded hardware and software. These internal components have evolved dramatically with the exception of batteries, which hold back the technological advancement of these devices. Because of the diversity of environments and users, solutions which seek a reduction on energy demand in mobile devices need to be adaptable through usage pattern recognition and prediction. Time series models are classic tools to solve forecast problems, but despite being a widely studied area, there are few studies related to mobile devices. Due to the diversity of time series forecast methods and shortage of similar work, this paper proposes a comparison of classic methods to time series forecast, running on mobile devices. These comparative tests have shown that the proposed method is computationally applicable to mobile devices, but due to non-stationary nature of some series or periods in these series, in terms of accuracy only 27.5 % of the tests forecast error were below 120 minutes, showing that these simplest methods for time series prediction can not serve for most related mobile solutions for energy optimization and usage pattern recognition.

**Keywords:** Energy, mobile devices, smartphones, android, time series, forecast.

# Lista de abreviaturas e siglas

3G *Terceira Geração*

AC *Alternate Current*

ARIMA *Auto Regressive Integrated Moving Average*

ARMA *Auto Regressive Moving Average*

CDMA *Code Division Multiple Access*

COC *Convention Over Configuration*

GPL *General Public License*

GPS *Global Positioning System*

HTML *HyperText Markup Language*

I/O *Input and Output*

ID *Identity*

IEEE *Institute of Electrical and Electronics Engineers*

IMEI *International Mobile Equipment Identity*

IMSI *International Mobile Subscriber Identity*

IP *Internet Protocol*

JVM *Java Virtual Machine*

MAE *Mean Absolute Error*

MVC *Model View Controller*

MIT *Massachusetts Institute of Technology*

LAN *Local Area Network*

PAFCL *Prediction Algorithm with Fixed Cycle Length*

PIB *Produto Interno Bruto*

PC *Personal Computer*

SDK *Software Development Kit*

SGBD *Sistema Gerenciador de Banco de Dados*

SMS *Short Message Service*

SO *Sistema Operacional*

USB *Universal Serial Bus*

UTC *Universal Time Coordinated*

UUID *Universally Unique Identifier*

WAN *Wide Area Network*

WEB *World Wide Web*

Wi-Fi *Wireless Fidelity*

# Lista de figuras

|     |   |    |
|-----|---|----|
| 2.1 | Exemplos gráficos de séries temporais. Fonte: Barros (2009). . . . .  | 26 |
| 2.2 | Sismógrafo registrando um terremoto no Haiti em 12/01/2010.<br>Fonte: Earthquake.it (2010). . . . .   | 26 |
| 2.3 | Duas séries estacionárias (a) e (b), e uma série não-estacionária<br>(c). Fonte: Box e Luceno (1997). . . . .   | 27 |
| 2.4 | As quatro componentes das séries temporais. Fonte: Almeida (2008).  | 29 |
| 2.5 | Uma série temporal decomposta em componente sazonal,<br>tendência-ciclo e remanescente. Fonte: Hyndman e Athanasopoulos<br>(2013). . . . .  | 31 |
| 2.6 | Produção quadrimestral de cerveja na Austrália: 1992/Q1 à<br>2008/Q3 com dois anos de previsão. Fonte: Hyndman e Athanasopoulos<br>(2013). . . . .  | 34 |
| 2.7 | Série temporal com a amplitude sazonal praticamente constante ao<br>longo do tempo. Fonte: Afonso (2010). . . . .   | 43 |
| 2.8 | Série temporal com a amplitude sazonal variando ao longo do<br>tempo. Fonte: Afonso (2010). . . . .   | 43 |
| 2.9 | Desenho esquemático mostrando a forma e as componentes das<br>configurações de baterias Íon-lítio. <b>a</b> , cilíndrica; <b>b</b> , moeda; <b>c</b> , pris-<br>mática; e <b>d</b> , fina e plana. Fonte: Tarascon e Armand (2001). . . . . | 47 |

|      |  |    |
|------|--|----|
| 2.10 | Modelo conceitual da HBI proposto em Rahmati, Qian e Zhong (2007). Fonte: Rahmati, Qian e Zhong (2007). . . . .  | 48 |
| 2.11 | Configurações das <b>redes para interconexão de sistemas (a)</b> e <b>LAN sem fios (b)</b> . Fonte: Tanenbaum (2003). . . . .                              | 50 |
| 2.12 | As camadas arquiteturais da plataforma Android. Fonte: Saha (2008). . . . .  | 53 |
| 4.1  | Arquitetura do monitor de bateria e conectividade. Fonte: Elaborada pelo autor. . . . .  | 67 |
| 4.2  | Dados de um participante da pesquisa de campo. Fonte: Elaborada pelo autor. . . . .  | 68 |
| 4.3  | Respectivamente: série original; e componentes tendência-ciclo, sazonal e remanescente, decompostas por Wessa (2013). Fonte Wessa (2013) . . . . .         | 73 |
| 4.4  | Respectivamente: série original; e componentes tendência-ciclo, sazonal e remanescente, decompostas nesta monografia. Fonte: Elaborada pelo Autor. . . . . | 74 |
| 4.5  | Previsão Holt-Winters aplicada pela ferramenta Wessa (2013). Em vermelho a previsão, em preto a série original. Fonte: Wessa (2013). 76                    |    |
| 4.6  | Previsão Holt-Winters aplicada pelo algoritmo desta monografia. Em vermelho a previsão, em azul a série original. Fonte: Elaborada pelo Autor. . . . .     | 76 |
| 4.7  | Ambiente dos testes comparativos para Android. Fonte: Elaborada pelo autor. . . . .  | 77 |
| 4.8  | Resultados de um teste no servidor. Em vermelho a previsão, em azul a série original. Fonte: Elaborada pelo Autor. . . . .                                 | 80 |

|      |   |    |
|------|---|----|
| 4.9  | A distribuição por hora dos eventos de bateria de um usuário que foi removido da população dos testes. Em azul eventos de conexão com a fonte de energia, em vermelho momentos de desconexão. Fonte: Elaborada pelo Autor. . . . .  | 82 |
| 4.10 | A distribuição por hora dos eventos de bateria de um dos usuários pertencentes a população selecionada. Em azul eventos de conexão com a fonte de energia, em vermelho momentos de desconexão. Fonte: Elaborada pelo Autor. . . . . | 83 |
| 5.1  | Previsão com pior MAE dos testes com dados de conectividade Wi-Fi. Em vermelho a previsão, em azul a série original. Fonte: Elaborada pelo Autor. . . . .   | 87 |
| 5.2  | Previsão com melhor MAE dos testes com dados de conectividade 3G. Em vermelho a previsão, em azul a série original. Fonte: Elaborada pelo Autor. . . . .  | 91 |

# Lista de tabelas

|     |  |     |
|-----|--|-----|
| 5.1 | Médias por tipo de dado e porcentagem dos testes com MAE abaixo dos 60 min . . . . . | 86  |
| 5.2 | Os 10 maiores MAEs de todos os testes . . . . .                                      | 87  |
| 5.3 | Comparação dos maiores MAEs do Holt-Winters com o Wessa por tipo de dado . . . . .   | 88  |
| 5.4 | Comparação dos menores MAEs do Holt-Winters com o Wessa por tipo de dado . . . . .   | 88  |
| 5.5 | Os 10 maiores MAEs dos testes de bateria . . . . .                                   | 89  |
| 5.6 | Os 10 maiores MAEs dos testes de conectividade 3G . . . . .                          | 89  |
| 5.7 | Os 10 maiores MAEs dos testes de conectividade Wi-Fi . . . . .                       | 90  |
| 5.8 | Os 10 menores MAEs de todos os testes . . . . .                                      | 90  |
| 5.9 | Média MAE por algoritmo e tipo de dado . . . . .                                     | 91  |
| D.1 | Os 10 menores MAEs dos testes de bateria . . . . .                                   | 101 |
| D.2 | Os 10 menores MAEs dos testes de conectividade 3G . . . . .                          | 102 |
| D.3 | Os 10 menores MAEs dos testes de conectividade Wi-Fi . . . . .                       | 102 |

# Lista de algoritmos

|     |  |    |
|-----|--|----|
| 2.1 | Registrando a intenção de captura de um evento do sistema . . . . .  | 54 |
| 2.2 | Código de um <i>BroadcastReceiver</i> recebendo mensagens das mudanças no estado da bateria . . . . .                          | 54 |
| 2.3 | Método chamado sempre que o serviço de monitoração é iniciado .  | 55 |
| 4.1 | Algoritmo do método naïve sazonal em Java . . . . .  | 70 |
| 4.2 | Algoritmo do método de média móvel ponderada em Java . . . . .   | 71 |
| 4.3 | Trecho final do método de previsão por decomposição (Soma das previsões de $S$ e $T$ ) escrito em Java . . . . .               | 72 |
| 4.4 | Algoritmo do método de previsão exponencial linear de Holt escrito em Java . . . . .   | 72 |
| 4.5 | Algoritmo do método de previsão exponencial tripla Holt-Winters escrito em Java, inspirado na implementação de Chandra (2012a) | 75 |
| B.1 | Método de decomposição clássica com média móvel escrita em um algoritmo Java . . . . .   | 97 |
| C.1 | Código Java de inicialização das componentes para o método de previsão Holt-Winters . . . . .                                  | 99 |

# Índice geral

|          |   |           |
|----------|---|-----------|
| <b>1</b> | <b>Introdução</b>   | <b>18</b> |
| 1.1      | Contextualização do problema estudado . . . . .                   | 19        |
| 1.2      | Objetivos . . . . .   | 20        |
| 1.3      | Justificativa . . . . .   | 21        |
| 1.4      | Metodologia . . . . .   | 22        |
| 1.5      | Estrutura do Trabalho . . . . .                                   | 23        |
| <b>2</b> | <b>Descrição teórica e ferramentas básicas de desenvolvimento</b> | <b>24</b> |
| 2.1      | Séries temporais . . . . .  | 25        |
| 2.1.1    | Classificação . . . . .   | 25        |
| 2.1.2    | Componentes . . . . .   | 28        |
| 2.2      | Previsões . . . . .   | 33        |
| 2.2.1    | Qualitativas e quantitativas . . . . .                            | 33        |
| 2.2.2    | De séries temporais e de dados transversais . . . . .             | 33        |
| 2.2.3    | Análise e preparação . . . . .                                    | 35        |

|          |   |           |
|----------|---|-----------|
| 2.2.4    | Avaliando a precisão . . . . .                            | 36        |
| 2.3      | Previsões em séries temporais . . . . .                   | 38        |
| 2.3.1    | Métodos <i>naïves</i> . . . . .                           | 38        |
| 2.3.2    | Por média móvel . . . . .                                 | 39        |
| 2.3.3    | Por decomposição . . . . .                                | 40        |
| 2.3.4    | Por suavização exponencial . . . . .                      | 41        |
| 2.4      | Dispositivos móveis . . . . .                             | 45        |
| 2.4.1    | Bateria . . . . .   | 46        |
| 2.4.2    | Conectividade . . . . .                                   | 50        |
| 2.5      | Ferramentas básicas de desenvolvimento . . . . .          | 52        |
| 2.5.1    | Android . . . . .   | 52        |
| 2.5.2    | RubyOnRails . . . . .                                     | 55        |
| <b>3</b> | <b>Estado da arte</b>                                     | <b>57</b> |
| 3.1      | Trabalhos relacionados . . . . .                          | 58        |
| 3.1.1    | Estudos de campo . . . . .                                | 58        |
| 3.1.2    | Comportamento dos usuários . . . . .                      | 59        |
| 3.1.3    | Previsão em séries temporais . . . . .                    | 59        |
| 3.2      | Conclusões parciais e oportunidades de pesquisa . . . . . | 60        |
| <b>4</b> | <b>Projeto experimental</b>                               | <b>62</b> |
| 4.1      | Visão geral do projeto . . . . .                          | 63        |

|          |   |           |
|----------|---|-----------|
| 4.2      | Monitor de bateria e conectividade . . . . .    | 64        |
| 4.2.1    | Dificuldades de um estudo de campo . . . . .    | 64        |
| 4.2.2    | Descrição do monitor . . . . .                  | 66        |
| 4.2.3    | Dados capturados pelo monitor . . . . .         | 68        |
| 4.3      | Implementação dos métodos de previsão . . . . . | 69        |
| 4.3.1    | <i>Naïve</i> sazonal . . . . .                  | 70        |
| 4.3.2    | Média móvel ponderada . . . . .                 | 70        |
| 4.3.3    | Por decomposição clássica . . . . .             | 71        |
| 4.3.4    | Holt-Winters . . . . .                          | 74        |
| 4.4      | Ambiente de testes . . . . .                    | 77        |
| 4.4.1    | Execução dos algoritmos . . . . .               | 77        |
| 4.4.2    | Métricas computacionais e de precisão . . . . . | 78        |
| 4.4.3    | Obtenção dos resultados . . . . .               | 80        |
| 4.4.4    | Características do ambiente de testes . . . . . | 81        |
| 4.4.5    | Dados utilizados nos testes . . . . .           | 81        |
| <b>5</b> | <b>Resultados experimentais</b>                 | <b>84</b> |
| 5.1      | Análise geral dos experimentos . . . . .        | 85        |
| 5.2      | Resultados . . . . .                            | 86        |
| <b>6</b> | <b>Conclusões e trabalhos futuros</b>           | <b>92</b> |

|            |   |     |
|------------|---|-----|
| Apêndice A | Termos de participação da pesquisa do monitor de dados de bateria e conectividade | 95  |
| Apêndice B | Algoritmo da decomposição clássica com média móvel                                | 97  |
| Apêndice C | Inicialização das componentes para o método de previsão Holt-Winters              | 99  |
| Apêndice D | Os 10 menores MAE para cada tipo de dado (Bateria, Conectividade 3G e Wi-Fi)      | 101 |

# Capítulo 1

## Introdução

*Este capítulo contextualiza o problema do uso de modelos de previsão em séries temporais para otimização energética em dispositivos móveis. Também apresenta os objetivos, as justificativas, a metodologia para resolução do problema e explica a estrutura geral desta monografia.*

## 1.1 Contextualização do problema estudado

Para os computadores de bolso, ou *smartphones*, o desempenho dos processadores cresceu 246 vezes durante os últimos 15 anos, no mesmo período a densidade energética das baterias apenas dobrou (PATHAK, 2012). Este poder de processamento atual em conjunto com os diversos componentes I/O embarcados (ex. GPS, câmeras, acelerômetros, Wi-Fi, 3G, etc.) aumentam progressivamente a disputa pelo recurso energético dos aparelhos. Essa escassez de energia é um gargalo na evolução dos dispositivos móveis e seu ecossistema.

Devido ao lento crescimento da capacidade das baterias e ao limite de peso e tamanho imposto pelo *design* dos aparelhos é necessário diminuir o consumo de energia dos outros componentes. Para isso diversas pesquisas científicas trazem soluções para otimização energética a nível de *hardware* e *software*, Vallina-Rodriguez e Crowcroft (2013) faz uma revisão bibliográfica ampla e atual das soluções neste cenário.

Atualmente nos *smartphones* os componentes de rede sem fio são os que mais consomem energia, como é visto na conclusão do estudo de Perrucci, Fitzek e Widmer (2011). Como agravante Shepard et al. (2011), Heikkinen e Nurminen (2010) e Falaki et al. (2010) demonstram que os usuários gastam grande parte do tempo de interação com os *smartphones* em atividades relacionadas a *internet*, ficando atrás somente dos telefonemas e SMS. Assim otimizar o gasto energético de programas que utilizam a *internet* tende a ser o principal foco dos esforços científicos para aumentar o tempo de uso do aparelho até uma próxima recarga.

Cada tecnologia de rede sem fio, nível do sinal de rádio e taxa de transmissão de dados possuem diferenças quanto ao desempenho e gasto energético nas transferências de dados. Os artigos Kalic, Bojic e Kusek (2012) e Nicholson e Noble (2008) abordam esta diversidade de desempenho e consumo de energia.

Esse cenário mostra que transferências de dados podem executar em diferentes condições. Uma forma de otimização energética portanto é adiar a execução de transferências tolerantes a atraso para condições mais favoráveis. Como em

carregamentos noturnos prolongados, carregamentos desnecessários após cargas completas, carga por conexão USB durante o uso de computadores e outros detalhados por Ferreira, Dey e Kostakos (2011).

Uma solução para deslocar a execução destas tarefas para momentos melhores é prever condições até uma determinada data e escolher entre elas a que trará menor custo energético. Por exemplo, o sistema apresentado em Nicholson e Noble (2008) prevê condições futuras de conectividade associando o movimento longitudinal e latitudinal do usuário com as conexões nas redes. Os autores deste trabalho sugerem que transferências tolerantes a atraso podem se beneficiar energeticamente de suas previsões.

Prever condições futuras utilizando um histórico de condições monitoradas é um problema que envolve séries temporais, ou seja, determinadas condições ao longo do tempo, como por exemplo os períodos de carregamento ou conectividade. Previsão em séries temporais é um problema bastante explorado e diversos métodos e modelos já foram amplamente testados e aplicados para prever cenários econômicos, climáticos, sociais, políticos, mercadológicos, de planejamento de infraestrutura, de gerenciamento desportivo, de saúde, e outros mais.

Dentro deste cenário de otimização energética em dispositivos móveis, da necessidade de prever melhores condições energéticas para transferência de dados tolerantes a atraso e devido a diversidade de métodos para previsão em séries temporais é apresentada nesta monografia a comparação (em termos de precisão, desempenho computacional e gasto energético) de métodos clássicos para previsão em séries temporais executando em dispositivos móveis.

## 1.2 Objetivos

O **principal objetivo** desta monografia é:

Apresentar uma análise comparativa de quatro métodos de previsão em séries temporais das condições de bateria e conectividade no ambiente de dispositivos

móveis, comparando-os quanto à precisão, desempenho computacional e eficiência energética.

## Objetivos Específicos

1. Desenvolver e descrever um monitor de dados de bateria e conectividade para *smartphones* Android.
2. Capturar dados de bateria e conectividade de usuários reais para serem utilizados nos testes.
3. Transcrever em linguagem Java quatro algoritmos de previsão das condições de bateria (períodos de carregamento) e conectividade (períodos de conexão ativa), correspondentes aos seguintes modelos de previsão em séries temporais:
  - ◇ A abordagem *naïve* (ingênua) sazonal;
  - ◇ Média móvel ponderada;
  - ◇ Decomposição clássica mais o método *naïve* e o método de Holt;
  - ◇ Método Holt-Winters de suavização exponencial;

## 1.3 Justificativa

Este estudo comparativo justifica-se pela falta de trabalhos que detalhem o desempenho computacional de métodos de previsão de séries temporais executando em ambientes de recursos escassos como em dispositivos móveis, ou especificamente em *smartphones*, juntamente com o aumento da importância dos métodos de previsão para soluções que buscam a otimização energética nestes aparelhos.

No melhor esforço de pesquisa utilizado não encontrou-se trabalhos deste tipo, apesar de alguns trabalhos utilizarem previsão em séries temporais nos dispositivos móveis, estes não apresentam resultados relacionados ao desempenho computacional.

Portanto este estudo comparativo será fonte de conhecimento para futuros trabalhos que necessitarão escolher métodos de previsão para serem utilizados em suas soluções. Conhecimento sobre a precisão, desempenho computacional e eficiência energética de alguns métodos de previsão em séries temporais consolidados, destacando os mais adequados a problemas que envolvem comportamento de carregamento e conectividade.

Na Seção 3.2 são descritas com maiores detalhes as oportunidades de pesquisa que tornam essa monografia relevante.

## 1.4 Metodologia

A metodologia desta monografia foi organizada nas seguintes etapas:

1. Desenvolver um monitor de dados de bateria e conectividade para *smartphones* Android de baixo impacto energético.
2. Elaborar uma pesquisa descritiva para recolher dados de bateria e conectividade de uma base de usuários anônimos e voluntários de um aplicativo Android que embarca o monitor de dados desenvolvido na etapa 1. Estes dados são utilizados nos testes dos algoritmos de previsão.
3. Realizar uma pesquisa bibliográfica sobre previsão, métodos de previsão de séries temporais e aplicações práticas destes métodos em dispositivos móveis.
4. Implementar para a plataforma Android, os quatro algoritmos de previsão (dos períodos de desconexão com fontes de recarga; dos períodos de desconectividade com as redes Wi-Fi e 3G) correspondentes aos quatro métodos de previsão de séries temporais definidos nos objetivos deste projeto.
5. Testar os algoritmos de previsão em ambiente controlado, utilizando os dados gravados na etapa 2 como entrada. Comparar os dados previstos nos testes com os dados reais monitorados correspondentes.

6. Comparar os resultados dos testes dos quatro algoritmos quanto a eficiência energética, desempenho computacional e precisão.

## 1.5 Estrutura do Trabalho

Esta monografia está estruturada em 6 capítulos. O primeiro introduz ao leitor o problema a ser abordado.

No **Capítulo 2** são descritas as teorias e ferramentas básicas de desenvolvimento necessárias neste trabalho.

No **Capítulo 3** são abordadas as pesquisas relacionadas ao problema deste projeto.

No **Capítulo 4** são apresentadas as ferramentas construídas, o ambiente e o plano de testes.

No **Capítulo 5** são discutidos os resultados dos testes realizados com os algoritmos de previsão.

Este trabalho é finalizado com conclusões e trabalhos futuros.

## Capítulo 2

# Descrição teórica e ferramentas básicas de desenvolvimento

*Neste capítulo são apresentados os conceitos teóricos de séries temporais, previsões, previsões em séries temporais, dispositivos móveis e das ferramentas básicas de desenvolvimento utilizadas nesta monografia.*

## 2.1 Séries temporais

Morettin e Tolo (1987) define sucintamente séries temporais como: um conjunto de observações ordenadas no tempo. E de forma mais detalhada Barros (2009) descreve as séries temporais como:

...um conjunto de observações ordenadas no tempo (não necessariamente igualmente espaçadas), e que apresentam dependência serial (isto é, dependência entre instantes de tempo).

Uma série temporal pode ser representada como uma sequência de valores  $y_1, y_2, y_3, \dots, y_T$  para uma série de tamanho  $T$ .  $y_T$  representa o último valor da série.

### 2.1.1 Classificação

#### Quanto à continuidade

Ehlers (2007) descreve as séries temporais **discretas** e **contínuas** como:

Uma série temporal é dita ser contínua quando as observações são feitas continuamente no tempo. Definindo o conjunto  $T = t : t_1 < t < t_2$  a série temporal será denotada por  $X(t) : t \in T$ . Uma série temporal é dita ser discreta quando as observações são feitas em tempos específicos, geralmente equidistantes. Definindo o conjunto  $T = t_1, \dots, t_n$  a série temporal será denotada por  $X_t : t \in T$ .

Para exemplificar a Figura 2.1 mostra duas séries temporais discretas e a Figura 2.2 mostra uma série temporal contínua obtida através de um sismógrafo.

Figura 2.1: Exemplos gráficos de séries temporais. Fonte: Barros (2009).

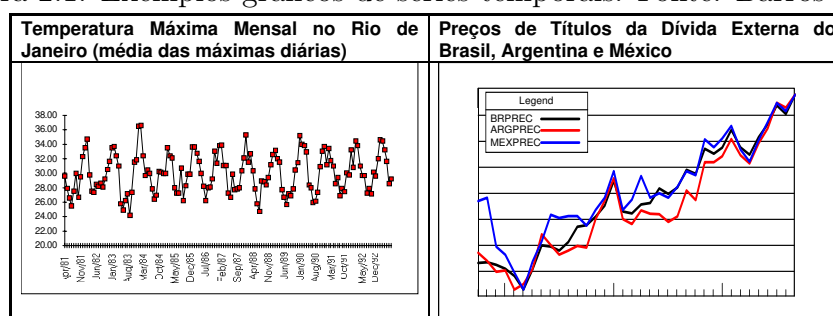
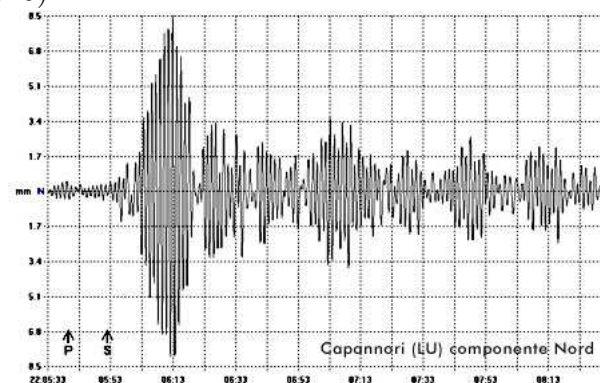


Figura 2.2: Sismógrafo registrando um terremoto no Haiti em 12/01/2010. Fonte: Earthquake.it (2010).



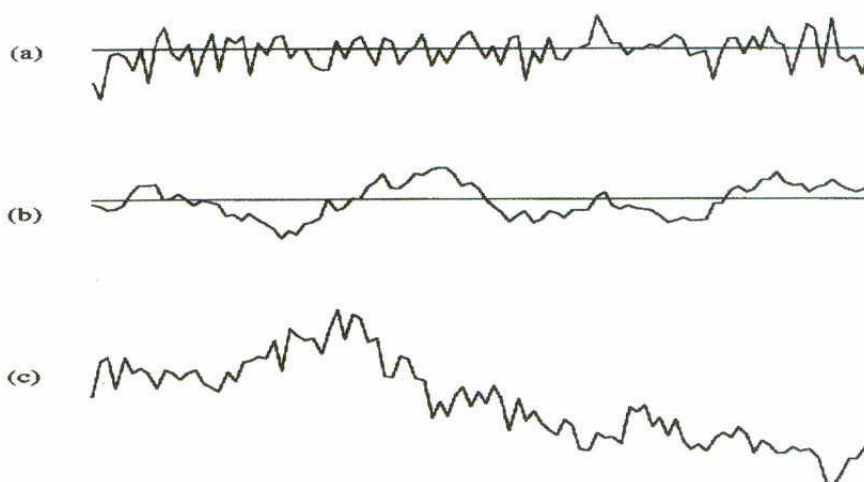
## Quanto ao determinismo

Uma série temporal pode ser dita **determinística** caso seja formada somente em função do tempo:  $y = f(\text{tempo})$ . Caso seja formada em função do tempo e mais uma componente aleatória  $\varepsilon$ , é dita **estocástica**:  $y = f(\text{tempo}, \varepsilon)$ . Um exemplo de série determinística é o PIB do Brasil dado pela soma dos PIBs de cada região ao longo do tempo e um de série temporal estocástica é o valor do dólar ao longo do dia.

### Quanto à estacionariedade

Para Azeredo (2010) uma série  $Y_t$  será estritamente estacionária, se a distribuição de probabilidade de seus valores não mudar com o tempo ou ainda para Morettin e Toloi (1987) “uma série que se desenvolve ao longo do tempo aleatoriamente ao redor de uma média constante, refletindo alguma forma de equilíbrio estável”. Dada uma mudança ou desequilíbrio nesta distribuição temos um nível de não-estacionariedade, que pode ser causado por alguma componente da série temporal. As componentes das séries temporais são explicadas na Seção 2.1.2. Na Figura 2.3 as séries temporais (a) e (b) possuem variação estacionária e a série (c) possui variação não-estacionária.

Figura 2.3: Duas séries estacionárias (a) e (b), e uma série não-estacionária (c).  
Fonte: Box e Luceno (1997).



Podemos ter também uma não-estacionariedade explosiva (mudança brusca e exponencial), ou uma não-estacionariedade cíclica durante períodos longos ou curtos (MORETTIN; TOLOI, 1987).

As séries temporais práticas do mundo real dificilmente serão estacionárias (AZEREDO, 2010). A maioria apresenta alguma não estacionariedade, como por exemplo uma tendência ascendente ou descendente de seus valores.

## Modelos univariados e multivariados

As séries temporais podem ser definidas por modelos de duas grandes classes, segundo Barros (2009):

- ◇ **Modelos Univariados:** a série temporal é explicada (prevista) apenas pelos seus valores passados;
- ◇ **Modelos Multivariados ou Causais:** a série temporal é explicada (prevista) pelos seus valores passados e também pelos valores passados de outras variáveis.

Nesta monografia serão abordados apenas modelos univariados, pois são os mais aplicados nos trabalhos que aplicam previsão em séries temporais na literatura relacionada à dispositivos móveis e otimização energética. Devido aos recursos de pesquisa limitados, principalmente o tempo, não foi possível abordar modelos multivariados.

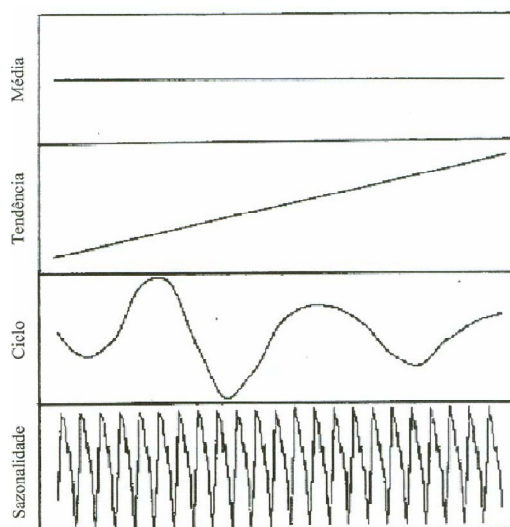
### 2.1.2 Componentes

As séries temporais demonstram diversos padrões e comportamentos que podem ser decompostos em componentes distintas para uma análise mais detalhada e um maior entendimento destas séries. Existem três tipos de componentes mais estudadas: a tendência, a sazonalidade e o ciclo de uma série temporal. Almeida (2008) ainda apresenta a característica de **média**, que existe quando os valores de uma série temporal flutuam em torno de um valor constante. A Figura 2.4 ilustra estas componentes. A média também pode ser chamada de nível da série temporal.

#### Tendência

Quando uma série temporal demonstra uma ascendência ou descendência de seus valores durante um longo período de tempo, caracterizamos este comportamento

Figura 2.4: As quatro componentes das séries temporais. Fonte: Almeida (2008).



como uma tendência (ALMEIDA, 2008). Hyndman e Athanasopoulos (2013) ainda acrescenta que a tendência não precisa ser necessariamente linear e que uma mudança de uma tendência ascendente para uma descendente, ou o contrário, denomina-se como uma mudança de direção da tendência.

### Sazonalidade

Segundo Hyndman e Athanasopoulos (2013) a sazonalidade em uma série temporal existe quando esta é influenciada por fatores sazonais (quadrimestral, semestral, semanal, diário, etc) que são sempre períodos fixos, bem conhecidos, previamente definidos e geralmente de calendário.

### Ciclo

Um comportamento cíclico de uma série temporal ocorre quando tal série demonstra ascendências e descendências recorrentes em períodos não fixos.

Para deixar bem clara a diferença entre sazonalidade e ciclo, Hyndman e Athanasopoulos (2013) explica:

Muitas pessoas confundem o comportamento cíclico com comportamento sazonal, mas eles são realmente muito diferentes. Se as flutuações não são de período fixo, então elas são cíclicas, se o período é imutável e associado a algum aspecto do calendário, então o padrão é sazonal. Em geral, o comprimento médio dos ciclos é maior que o comprimento do padrão sazonal, e a magnitude dos ciclos tende a ser mais variável do que a magnitude dos padrões sazonais. (Tradução livre)

## Decomposição

Qualquer variação em uma série temporal que não pode ser atribuída aos componentes sazonais, tendência, ciclo e média, é então atribuído ao ruído aleatório no processo gerador dos dados, tal ruído não é matematicamente modelável (ALMEIDA, 2008). Para entender cada um desses componentes, suas relações e participação no comportamento da série é necessário aplicar a decomposição destes componentes, ou seja, separá-los da série original.

Pode-se compreender uma série temporal  $y_t$  como composta de três componentes: uma sazonal, um agregado de tendência e ciclo (tendência-ciclo, as vezes chamado somente de tendência), e a remanescente (aleatoriedade da série). Hyndman e Athanasopoulos (2013) demonstra então de forma aditiva ou multiplicativa os modelos de constituição das séries temporais apresentados nas Equações 2.1 e 2.2 a seguir:

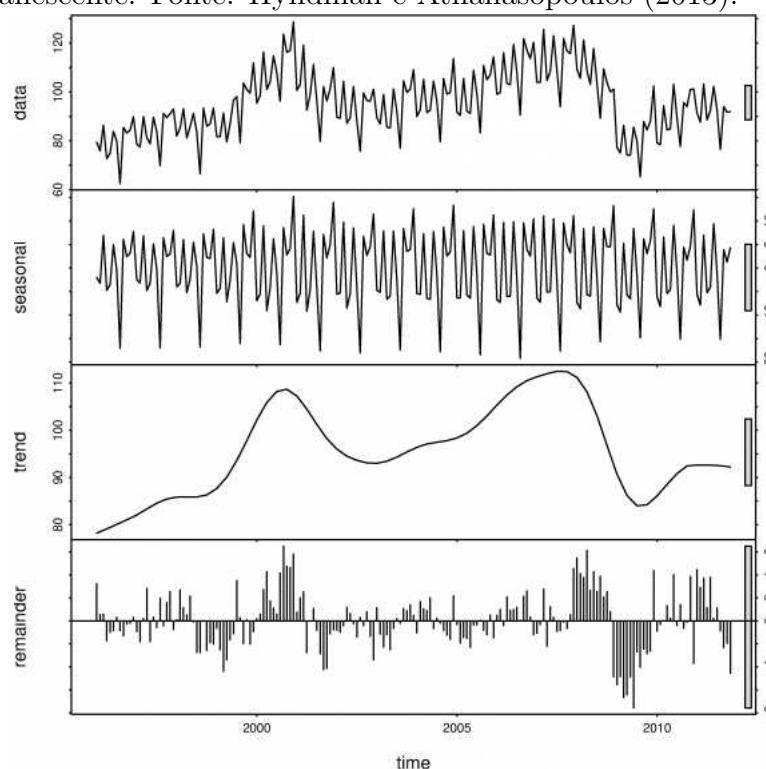
$$y_t = S_t + T_t + E_t \quad (2.1)$$

$$y_t = S_t \times T_t \times E_t \quad (2.2)$$

Onde  $y_t$  é o valor observado,  $S_t$  é o valor sazonal,  $T_t$  é o valor da tendência-ciclo e  $E_t$  é o valor remanescente (ou irregular, ou errático), todos no período  $t$ . A Figura 2.5 ilustra estas componentes.

Quando  $S_t$  não depende das outras componentes, como por exemplo  $T_t$ , é mais adequado utilizar-se o modelo aditivo. Já quando as amplitudes sazonais variam com a tendência-ciclo ( $T_t$ ) o modelo mais adequado é o multiplicativo (MORETTIN; TOLOI, 1987).

Figura 2.5: Uma série temporal decomposta em componente sazonal, tendência-ciclo e remanescente. Fonte: Hyndman e Athanasopoulos (2013).



Existem diversos métodos para se extrair cada uma destas principais componentes citadas anteriormente. Para exemplificação, a seguir será mostrado os passos do **método clássico de decomposição aditiva** de uma série temporal.

Hyndman e Athanasopoulos (2013) organiza a decomposição em 4 passos:

Inicialmente será discutido o **método de suavização usando média móvel**, utilizado no primeiro passo da decomposição clássica para estimar a componente tendência-ciclo, o modelo deste método é visto na Equação 2.3:

$$\hat{T}_t = \frac{1}{m} \sum_{j=-k}^k y_{t+j} \quad (2.3)$$

Na Equação 2.3,  $m = 2k + 1$  é chamado de ordem da média móvel.

A média móvel de ordem  $m$  pode ser chamada também de  $m$ -MA (HYNDMAN; ATHANASOPOULOS, 2013).

As médias móveis podem ser aplicadas sequencialmente diversas vezes em uma mesma série temporal (com valores diferentes ou iguais de  $m$ ). Para estes tipos de aplicação utiliza-se a notação  $k \times m$ -MA que significa aplicar uma média móvel  $m$ -MA e em seguida uma  $k$ -MA, onde  $k$  é qualquer número natural (HYNDMAN; ATHANASOPOULOS, 2013).

Inserindo em  $m$  o valor do período ou índice sazonal da série ( $m = 4$  para dados quadrimestrais,  $m = 12$  para dados mensais,  $m = 7$  para dados com padrão semanal) e usando  $y_t = S_t + T_t + E_t$  como modelo de composição temos os passos:

- ◇ **Passo 1:** Se  $m$  for um número par, calcular a componente tendência-ciclo usando uma  $2 \times m$ -MA (média móvel) para obter  $\hat{T}_t$ . Se  $m$  for um número ímpar, calcular a componente tendência-ciclo usando uma  $m$ -MA para obter  $\hat{T}_t$ .
- ◇ **Passo 2:** anular a componente tendência da série através da subtração:  $y_t - \hat{Y}_t$ .
- ◇ **Passo 3:** estimar a componente sazonal de cada período de  $m$  tirando-se a média dos valores com tendência anulada daquele período de  $m$ . Esses valores sazonais de cada período  $m$  são então ajustados para garantir que somem zero. A componente sazonal da série é obtida ao se juntar ordenada e repetidamente os valores sazonais de cada temporada (e.g. todos os meses do ano). Com isso temos  $\hat{S}_t$ .
- ◇ **Passo 4:** a componente remanescente é calculada subtraindo-se as componentes sazonal e tendência-ciclo estimados:  $\hat{E}_t = y_t - \hat{T}_t - \hat{S}_t$ .

## 2.2 Previsões

Uma previsão é a criação de informações futuras através de uma análise de informações do passado. Esse palpite tem chances de estar errado porém geralmente é um valor próximo do real e isso mantém a importância dos estudos das técnicas de previsão. Nota-se também que a palavra predição confunde-se como sinônimo de previsão, porém na verdade significa uma descoberta de algo que irá acontecer no futuro, sem incertezas quanto ao valor predito.

### 2.2.1 Qualitativas e quantitativas

Quando não existem dados ou tais dados não são relevantes para uma previsão então métodos **qualitativos** podem ser utilizados. Para Hyndman e Athanasopoulos (2013) tais métodos não são puramente palpites ou conjecturas, mas utilizam técnicas estruturadas e bem desenvolvidas para previsões mesmo sem o uso de dados históricos. Já Almeida (2008) deixa claro que como os métodos qualitativos baseiam-se em opiniões de especialistas, eles são vulneráveis à tendências que podem comprometer a confiabilidade de seus resultados com um baixo grau de precisão.

Os métodos **quantitativos** podem ser utilizados quando um histórico de dados do passado existe e quando alguns padrões destes dados do passado continuam a se repetir nos dados futuros (HYNDMAN; ATHANASOPOULOS, 2013). A maioria das previsões quantitativas utilizam os dados no formato de séries temporais (valores coletados ao longo do tempo como visto na Seção 2.1) ou dados transversais (valores coletados em um único ponto no tempo).

### 2.2.2 De séries temporais e de dados transversais

Os dados transversais são um conjunto de dados unidimensionais ou conjunto de dados em um único ponto no tempo. Estes dados são constituídos de características diversas de vários objetos de uma mesma classe todas observadas no

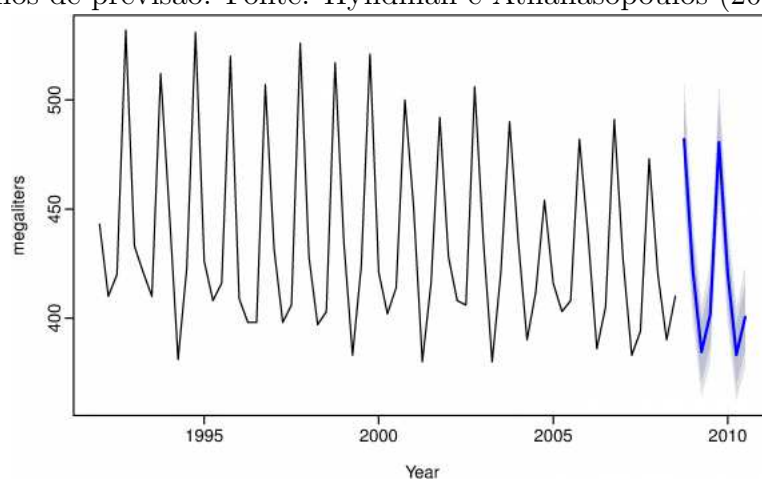
mesmo período, como por exemplo um conjunto de dados para estudo dos preços de imóveis em um determinado ano por número de quartos, área de construção, número de andares, etc. Para Hyndman e Athanasopoulos (2013) os dados transversais são usados quando deseja-se prever o valor de algo que nós não observamos usando informações de outros casos que observamos. O objetivo dos modelos de previsão em dados transversais é captar as relações destes dados e usá-las para as previsões.

As séries temporais, vistas na Seção 2.1, são úteis quando deseja-se prever valores que variam ao longo do tempo (HYNDMAN; ATHANASOPOULOS, 2013). Portanto a previsão em séries temporais tem como objetivo estimar valores futuros da série baseando-se nos valores passados da série, sem a necessidade de analisar-se fatores externos e tampouco tentar prevê-los.

As previsões em séries temporais são razoavelmente precisas para futuros imediatos e começam à fornecer dados mais imprecisos a medida que o futuro ao qual se quer prever distancia-se, com exceção de séries temporais com padrões de dados bastante estáveis (ALMEIDA, 2008).

A Figura 2.6 ilustra um caso de previsão em séries temporais. Na Seção 2.3 alguns métodos de previsão em séries temporais serão discutidos.

Figura 2.6: Produção quadrimestral de cerveja na Austrália: 1992/Q1 à 2008/Q3 com dois anos de previsão. Fonte: Hyndman e Athanasopoulos (2013).



### 2.2.3 Análise e preparação

Hyndman e Athanasopoulos (2013) estrutura a previsão em 5 etapas:

#### 1. Definição do problema:

Definir bem o problema necessita de um entendimento do propósito das previsões. É importante entender os aspectos e ambiente que circundam os objetos e seus dados observados.

#### 2. Coleta dos dados:

Dois tipos de informações são necessárias, dados históricos e conhecimento do domínio estudado. Em muitos casos é difícil coletar dados históricos suficientes para uma boa previsão e em outros casos dados históricos muito antigos são menos úteis devido à mudanças no domínio ao qual deseja-se realizar a previsão.

#### 3. Análise preliminar:

Sempre deve-se começar criando gráficos relevantes dos dados e então tentar achar respostas para as seguintes perguntas: existem padrões consistentes? Existe uma tendência significativa? Existe um comportamento sazonal dos objetos estudados? Existe a evidência de padrões cíclicos? Existem valores ou comportamentos discrepantes que precisam ser explicados por conhecimento especialista? Existem relações (e o quanto estas são fortes) entre as variáveis analisadas?

#### 4. Escolha e ajustes dos modelos:

Definir o melhor modelo a ser utilizado depende da disponibilidade dos dados históricos, da força das relações entre as variáveis e dos objetivos das previsões (inclusive o grau de precisão esperado). Portanto a escolha passa pela análise do problema e dos dados capturados e ainda é necessário verificar se os parâmetros requisitados pelos modelos serão preenchidos corretamente por estes dados históricos disponíveis.

## 5. Uso e avaliação do modelo:

Assim que um modelo tiver sido escolhido e seus parâmetros estimados, utiliza-se ele para realizar as previsões. O desempenho do modelo (ou a precisão de suas previsões) só pode ser avaliado depois que os dados reais do período previsto se tornem disponíveis. Diversos métodos para avaliar a precisão dos resultados de uma previsão foram desenvolvidos para este passo conclusivo, alguns deles serão discutidos na Seção 2.2.4 logo em seguida.

### 2.2.4 Avaliando a precisão

Segundo Barros (2009) o conhecimento do passado não nos diz exatamente como será o futuro e assim sempre existe incerteza associada às previsões. Assim os métodos de previsão buscam calcular os valores mais prováveis no futuro sendo necessário também limites de confiança e graus de precisão. Alguns métodos criados para avaliar o nível de acerto das previsões serão discutidos a seguir.

Hyndman e Athanasopoulos (2013) categoriza os principais métodos para cálculo da precisão de um modelo de previsão em: cálculos de erros dependentes da escala e cálculos erros percentuais.

Tomar  $y_t$  como o dado observado no instante  $t$  e  $\hat{y}_t$  o valor previsto de  $y_t$  nas descrições seguintes.

#### **Erros dependentes da escala**

De forma genérica os erros destes tipos de métodos são basicamente um cálculo de  $e_t = y_t - \hat{y}_t$ , que resulta em valores na mesma escala dos dados estudados.

Portanto os valores de precisão calculados através destes métodos são dependentes da escala e não podem ser usados para comparação entre previsão de dados em diferentes escalas (HYNDMAN; ATHANASOPOULOS, 2013). São bons métodos para comparação entre séries de uma mesma base de dados, além disso são simples de aplicar e compreender.

Os dois cálculos dependentes de escala mais populares são o MAE (*Mean Absolute Error*) e o RMSE (*Root Mean Squared Error*) eles fornecem valores de erros absolutos e quadráticos respectivamente. Seus modelos são definidos nas Equações 2.4 e 2.5 seguintes:

$$MAE = \frac{\sum_{i=1}^N |e_i|}{N} \quad (2.4)$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^N e_i^2}{N}} \quad (2.5)$$

### Erros percentuais

Os valores de erros destes tipos de métodos são basicamente cálculos de  $p_t = 100(e_t/y_t)$ , onde  $p_t$  é o erro percentual. Estes são independentes de escala e podem ser utilizados para comparar o desempenho de previsões em base de dados diferentes.

Segundo Hyndman e Athanasopoulos (2013) o método mais comum para cálculo de erros percentuais é o MAPE (*Mean Absolute Percentage Error*) demonstrado na Equação 2.6:

$$MAPE = \frac{\sum_{i=1}^N |100(\frac{e_i}{y_i})|}{N} \quad (2.6)$$

Para Hyndman e Athanasopoulos (2013) cálculos de erros percentuais possuem a desvantagem de produzirem valores infinitos ou indefinidos se  $y_t = 0$  para qualquer  $t$  no intervalo estudado, e valores extremos quando algum  $y_t$  tende à zero. Outro problema citado pelos autores de Hyndman e Athanasopoulos (2013) é o desequilíbrio entre os erros negativos e positivos, onde os erros negativos ficam com um peso maior no resultado final.

## 2.3 Previsões em séries temporais

Prever em séries temporais é analisar os valores passados de uma série e/ou de outras séries e através de algum método obter boas previsões de curto prazo de valores futuros desta série. A previsão de uma série no instante  $T + h$  é denotada como  $\hat{y}_{T+h}$ , onde  $h$ , chamado de horizonte de previsão, é o número de instantes à frente para o qual é feita a previsão.

Existem diversos métodos e modelos para previsão em séries temporais, porém serão descritos somente os métodos que foram definidos nos objetivos e escopo desta monografia para serem comparados e estudados. Portanto em seguida serão abordados estes quatro métodos: *naïve* sazonal, média móvel ponderada, por decomposição clássica (utilizando o método *naïve* e o método de Holt) e suavização exponencial de Holt-Winters.

### 2.3.1 Métodos *naïves*

Os métodos ditos *naïves* ou ingênuos são provavelmente os mais simples para se obter previsões. Almeida (2008) ainda acrescenta que estes métodos são “rápidos e fáceis de usar, com baixo custo e fáceis de serem entendidos”.

O exemplo mais representativo de um método *naïve* é aquele onde o valor previsto do próximo período é o mesmo do último período observado:  $\hat{y}_{t+1} = y_t$ .

Caso a série temporal estudada apresente forte comportamento sazonal, ainda podemos adicionar ao modelo *naïve* o mecanismo sazonal, temos então o chamado **método *naïve* sazonal** mostrado na Equação 2.7 a seguir:

$$\hat{y}_{T+h} = y_{T+h-km} \quad \text{onde} \quad k = [(h-1)/m] + 1 \quad \text{e} \quad m = \text{período sazonal} \quad (2.7)$$

Da mesma forma que o *naïve* simples, este método utiliza o valor do último

período para a previsão, porém o último período de mesma posição sazonal. Por exemplo, o valor do próximo mês de Fevereiro será o mesmo do mês Fevereiro anterior, e assim com dias da semana, quadrimestres, bimestres, semestres, etc.

### 2.3.2 Por média móvel

Segundo Afonso (2010) um modelo de média móvel é empregado “para suavizar as variações randômicas e gerar estimativas de previsão de maior grau de confiabilidade”. Não confundir os modelos de média móvel para previsão com o método de suavização com média móvel usado para estimar a componente tendência-ciclo de uma série temporal, visto na Seção 2.1.2.

O modelo de **média móvel simples** faz cada período a ser previsto como sendo a média das últimas  $n$  observações. Definir o valor de  $n$ , tamanho da janela como é chamado, é um dos problemas deste método, quanto maior o valor de  $n$  mais suave é a previsão, de forma contrária, valores menores de  $n$  tendem a tornar a previsão mais brusca. No modelo de média móvel simples todas as observações anteriores usadas no cálculo possuem mesmo peso na média ( $1/n$ ). A Equação 2.8 mostra uma formalização deste modelo.

$$\hat{y}_t = \frac{\sum_{i=0}^n y_{t-i}}{n} \quad (2.8)$$

Uma das formas de se solucionar o problema da escolha do valor de  $n$  é definir o inteiro positivo que minimiza a soma de quadrados dos erros de previsão 1 passo à frente (BARROS, 2009). Ou ainda de forma heurística simples para séries temporais com flutuações (sazonais ou cíclicas) estáveis se usa um valor de  $n$  grande e em séries onde estas flutuações são significativas utiliza-se um valor de  $n$  pequeno (AFONSO, 2010).

Para Barros (2009) “na prática é razoável supor que as observações mais recentes sejam mais relevantes para a previsão dos próximos valores da série, e portanto deveriam receber um peso maior que as observações mais antigas”. Para isso

acrescentam-se pesos no modelo de média móvel simples e temos então o modelo de média móvel ponderada apresentado na Equação 2.9.

$$\hat{y}_t = \frac{\sum_{i=0}^n n - i(y_{t-i})}{\sum_{i=1}^n i} \quad (2.9)$$

Os pesos no modelo da Equação 2.9 possuem um decrescimento aritmético, cada peso é menor que o anterior em uma unidade.

### 2.3.3 Por decomposição

Segundo Hyndman e Athanasopoulos (2013):

Enquanto decomposição é útil principalmente para estudar dados de séries temporais e explorar as mudanças históricas ao longo do tempo, ela também pode ser usada para previsões.

A partir dos conhecimentos acerca da decomposição de séries temporais vistos na Seção 2.1.2 temos uma técnica geral de previsão utilizando métodos de decomposição (HYNDMAN; ATHANASOPOULOS, 2013).

Dada uma decomposição aditiva apresenta-se o seguinte modelo de série temporal:  $y_t = \hat{S}_t + \hat{A}_t$ , onde  $\hat{A}_t = \hat{T}_t + \hat{E}_t$  é a componente sazonalmente ajustada da série decomposta. Para uma decomposição multiplicativa temos:  $y_t = \hat{S}_t \hat{A}_t$ , onde  $\hat{A}_t = \hat{T}_t \hat{E}_t$ .

Então para previsões usando-se uma série decomposta (e.g. através do método clássico de decomposição visto na seção 2.1.2), aplica-se modelos de previsão de forma separada nas componentes  $\hat{S}_t$  e  $\hat{A}_t$ . Por exemplo pode-se usar o método *naïve* sazonal na componente  $\hat{S}_t$  (assumindo que  $\hat{S}_t$  não se altera ou se altera vagarosamente) e para o componente  $\hat{A}_t$  qualquer método de previsão não-sazonal pode ser utilizado, como por exemplo o método de Holt discutido na Seção 2.3.4 seguinte.

### 2.3.4 Por suavização exponencial

A categoria de métodos de previsão por suavização exponencial possui como principal característica a utilização de pesos exponenciais decrescentes, da observação mais recente em direção às observações mais antigas, ou seja, os valores observados antigos vão possuindo gradualmente menor importância na previsão (ALMEIDA, 2008).

Devido à um fator custo benefício grande, em relação à precisão, rapidez e facilidade de uso, os métodos de suavização exponencial são citados por muitos autores como sendo os favoritos da maior parcela das pessoas. O modelo mais simples de suavização exponencial é justamente a Suavização Exponencial Simples ou *Single Exponential Smoothing* (SSE), temos então o modelo (HYNDMAN; ATHANASOPOULOS, 2013) na Equação 2.10:

$$\hat{y}_{T+1|T} = \alpha y_T + \alpha(1 - \alpha)y_{T-1} + \alpha(1 - \alpha)^2 y_{T-2} + \dots \quad (2.10)$$

Onde  $0 \leq \alpha \leq 1$  é o parâmetro de suavização. A previsão de um passo a frente  $T + 1$  é uma média ponderada de todas as observações da série temporal  $y_1, \dots, y_T$ . A taxa que os pesos decrescem é controlada pelo parâmetro  $\alpha$ . Quando  $\alpha = 1$  então  $\hat{y}_{T+1|T} = y_T$ , ou seja esta previsão será idêntica à uma feita por um método *naïve*. Segundo Hyndman e Athanasopoulos (2013) o parâmetro  $\alpha$  pode ser estimado através da minimização dos erros ou ainda de acordo com Afonso (2010) uma escolha comum é definir  $\alpha$  variando entre 0,01 e 0,3.

O método de suavização exponencial simples é mais aconselhado quando as flutuações de tendência e sazonalidade não são significativas (AFONSO, 2010). Para o caso de séries não sazonais porém com tendência linear pode-se aplicar o método de Holt e para séries temporais com tendência e sazonalidade pode-se aplicar o método Holt-Winters de suavização exponencial aditivo ou multiplicativo. Estes dois métodos serão descritos a seguir.

## Método de Holt

Holt (1957 apud HYNDMAN; ATHANASOPOULOS, 2013) melhorou a suavização exponencial simples para permitir previsões em séries com tendência. Esse método, também chamado de *Double Exponential Smoothing*, envolve a Equação de previsão 2.11 e as duas equações de suavização: a Equação 2.12 do nível da série (ou média) e a Equação 2.13 da tendência (HYNDMAN; ATHANASOPOULOS, 2013).

◇ Equação de previsão:

$$\hat{y}_{t+h|t} = \ell_t + hb_t \quad (2.11)$$

◇ Equação de suavização do nível:

$$\ell_t = \alpha y_t + (1 - \alpha)(\ell_{t-1} + b_{t-1}) \quad (2.12)$$

◇ Equação de suavização da tendência:

$$b_t = \beta(\ell_t - \ell_{t-1}) + (1 - \beta)b_{t-1} \quad (2.13)$$

Nestas equações  $\ell_t$  é a estimativa da média da série no período  $t$  e  $b_t$  é a estimativa da tendência no período  $t$  (HYNDMAN; ATHANASOPOULOS, 2013). Neste modelo apresentam-se também a constante  $0 \leq \alpha \leq 1$  para a suavização da média e a constante  $0 \leq \beta \leq 1$  para suavização da tendência da série (ALMEIDA, 2008). Assim como nos outros modelos de suavização exponencial os parâmetros podem ser estimados por minimização dos erros.

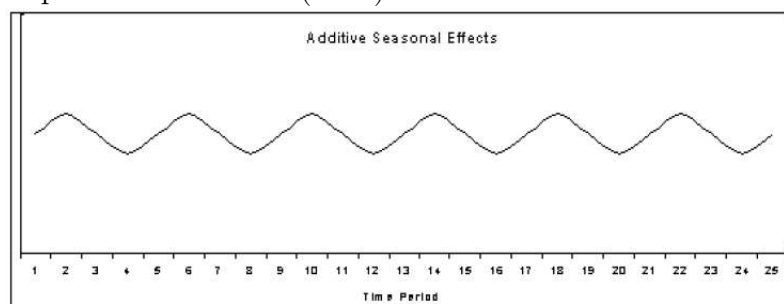
## Método Holt-Winters

Winters (1960) evoluiu o método de Holt (1957) para capturar também a sazonalidade, ou seja o método Holt-Winters como é chamado (ou também *Triple Exponential Smoothing*), é adequado para séries que apresentam além da componente tendência, uma componente sazonal.

São duas versões diferentes do método Holt-Winters, uma para se modelar a sazonalidade de forma aditiva e outra para modelar a sazonalidade de forma multiplicativa.

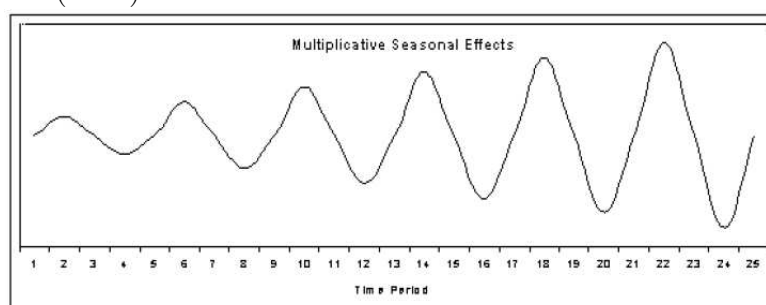
Na forma aditiva a amplitude da variação sazonal é praticamente constante ao longo do tempo, como ilustrado na Figura 2.7.

Figura 2.7: Série temporal com a amplitude sazonal praticamente constante ao longo do tempo. Fonte: Afonso (2010).



Na forma multiplicativa a amplitude da variação sazonal aumenta ou diminui em função do tempo, assim como é visto na Figura 2.8.

Figura 2.8: Série temporal com a amplitude sazonal variando ao longo do tempo. Fonte: Afonso (2010).



O método Holt-Winters compreende de forma geral uma equação de previsão e três equações de suavização. Das três equações de suavização uma é aplicada na média (nível)  $\ell_t$ , uma para a tendência  $b_t$  e a última para a componente sazonal  $s_t$ , com os respectivos parâmetros de suavização  $\alpha$ ,  $\beta$  e  $\gamma$  (HYNDMAN; ATHANASOPOULOS, 2013). Nas equações usa-se  $m$  como o período da sazonalidade, por exemplo para sazonalidade quadrimestral temos  $m = 4$ , para mensal  $m = 12$ , para semanal  $m = 7$  e assim por diante.

Para Becker (2010):

A dificuldade na determinação dessas constantes é uma das desvantagens do processo de suavização Holt-Winters. Como vantagem pode ser citada a aplicabilidade do processo, adequada a análise de séries com comportamento mais geral. Além disso, o modelo Holt-Winters não exige a normalidade dos erros, nem que a série seja um processo estacionário (ruído branco).

Dada esta caracterização do método Holt-Winters, sua forma aditiva compreende as Equações 2.14, 2.15, 2.16, 2.17 e 2.18 (HYNDMAN; ATHANASOPOULOS, 2013).

◇ Equação de previsão:

$$\hat{y}_{t+h|t} = \ell_t + hb_t + s_{t-m+h_m^+} \quad (2.14)$$

$$\text{onde } h_m^+ = \lfloor (h-1) \bmod m \rfloor + 1 \quad (2.15)$$

◇ Equação de suavização do nível:

$$\ell_t = \alpha(y_t - s_{t-m}) + (1 - \alpha)(\ell_{t-1} + b_{t-1}) \quad (2.16)$$

◇ Equação de suavização da tendência:

$$b_t = \beta(\ell_t - \ell_{t-1}) + (1 - \beta)b_{t-1} \quad (2.17)$$

◇ Equação de suavização sazonal:

$$s_t = \gamma(y_t - \ell_{t-1} - b_{t-1}) + (1 - \gamma)s_{t-m} \quad (2.18)$$

E também a forma multiplicativa do método Holt-Winters nas Equações 2.19, 2.20, 2.21, 2.22 e 2.23 (HYNDMAN; ATHANASOPOULOS, 2013):

◇ Equação de previsão:

$$\hat{y}_{t+h|t} = (\ell_t + hb_t)s_{t-m+h_m^+} \quad (2.19)$$

$$\text{onde } h_m^+ = \lfloor (h-1) \bmod m \rfloor + 1 \quad (2.20)$$

◇ Equação de suavização do nível:

$$\ell_t = \alpha \frac{y_t}{s_{t-m}} + (1-\alpha)(\ell_{t-1} + b_{t-1}) \quad (2.21)$$

◇ Equação de suavização da tendência:

$$b_t = \beta(\ell_t - \ell_{t-1}) + (1-\beta)b_{t-1} \quad (2.22)$$

◇ Equação de suavização sazonal:

$$s_t = \gamma \frac{y_t}{(\ell_{t-1} + b_{t-1})} + (1-\gamma)s_{t-m} \quad (2.23)$$

Nota-se que a Equação 2.22 de suavização da componente tendência continua igual à do método aditivo. Os valores ótimos das constantes de suavização nas duas formas do método Holt-Winters podem ser determinados pelo vetor  $(\alpha, \beta, \gamma)$  que minimiza a soma dos erros de previsão (BECKER, 2010).

## 2.4 Dispositivos móveis

Os dispositivos móveis são computadores de fácil locomoção e que possuem capacidade para desempenharem suas funcionalidades em movimento (B'FAR, 2005). Um dos componentes mais importantes para a mobilidade destes aparelhos é a bateria, que fornece energia armazenada para alimentar os circuitos internos do dispositivo. Os componentes de rede sem fio (Wi-Fi, 3G, Bluetooth) também possuem muita importância no uso destes dispositivos, interligando estes à internet ou outras redes.

Os *smartphones*, um tipo específico de dispositivo móvel, são celulares modernos de alta tecnologia que combinam a funcionalidade de uma comunicação portátil com capacidades computacionais (CARROLL; HEISER, 2010). Estas capacidades são possibilitadas por diversos dispositivos de *hardware* embarcados como: GPS, câmeras, acelerômetros, bússolas, magnetômetros, sensores de luminosidade, sensores infravermelho, placas de aceleração gráfica, dispositivos de comunicação sem fio (Bluetooth, 3G, Wi-Fi) e outros.

A popularização destes aparelhos fez surgir a cultura do estar sempre conectado: aos grupos de interesse comum, às comunidades profissionais e à informação (BOULOS et al., 2011). Juntos essa cultura, o uso intenso e a rica funcionalidade pressionam o tempo de vida das baterias, aumentando a importância de gerenciamentos de energia efetivos (CARROLL; HEISER, 2010).

### 2.4.1 Bateria

Segundo Dresch (2010):

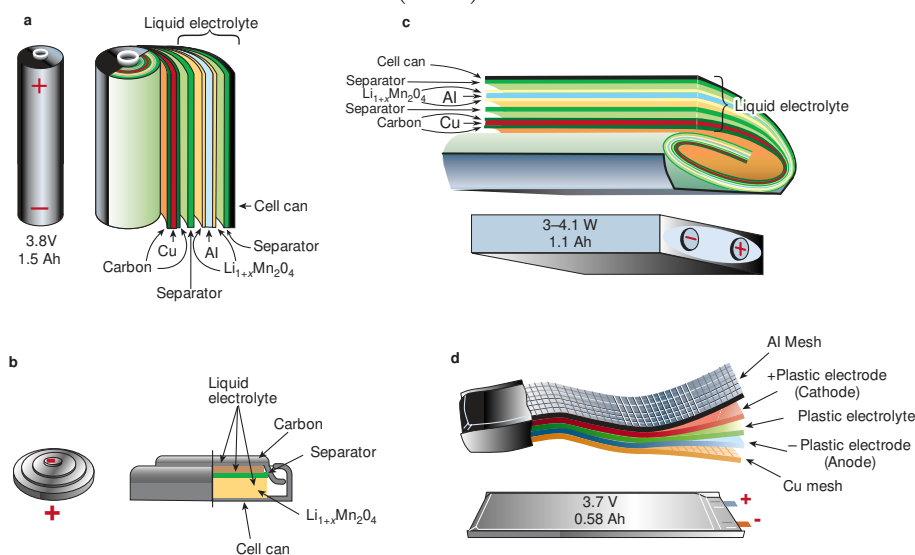
A bateria é um dispositivo que transforma a energia química armazenada nos materiais ativos em energia elétrica, através de uma reação eletroquímica de oxidação-redução (redox). No caso dos sistemas recarregáveis, a bateria é recarregada pelo processo reverso. Esse tipo de reação envolve a transferência de elétrons de um tipo de material para outro através de um circuito elétrico.

Um invento datado de 1800 e criado por Alessandro Volta, a pilha voltaica foi a primeira bateria (DRESCH, 2010). Em 1860 Gaston Plante desenvolveu as baterias chumbo-ácido, posteriormente em 1901 Thomas Alva Edison e Waldemar Junger criaram as baterias níquel-cádmio (DRESCH, 2010). Nenhuma dessas invenções tiveram sucesso comercial até o desenvolvimento das baterias NiMH (níquel-hidreto metálico) de 1950 até 1980, quando estas ganharam tal importância mercadológica (DRESCH, 2010). As baterias NiMH são a tecnologia das atuais pilhas recarregáveis comuns. A tecnologia mais utilizada em dispositivos

móveis atualmente, as baterias recarregáveis de *Lithium-ion* (Íon-lítio) foram desenvolvidas de 1970 à 1985, sendo comercialmente lançadas em 1991 pela Sony (TARASCON; ARMAND, 2001).

As baterias de Íon-lítio possibilitaram maiores densidades energéticas com menores pesos e tamanhos, mantendo-se as voltagens das baterias anteriores de aproximadamente 3,7 volts. A Figura 2.9 ilustra as configurações e formatos de tipos de baterias Íon-lítio, inclusive o tipo fino e leve dos *smartphones* modernos.

Figura 2.9: Desenho esquemático mostrando a forma e as componentes das configurações de baterias Íon-lítio. **a**, cilíndrica; **b**, moeda; **c**, prismática; e **d**, fina e plana. Fonte: Tarascon e Armand (2001).

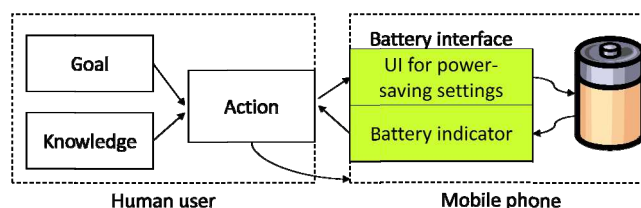


As baterias são geralmente os componentes mais pesados, custosos e menos ambientais dos dispositivos móveis (ARMAND; TARASCON, 2008). Elas evoluíram lentamente em comparação aos outros dispositivos ao qual fornecem energia. Dresch (2010) informa que “enquanto em sete anos a capacidade de processamento dos semicondutores aumentou em 2600%, as tecnologias das baterias aumentaram somente em 65%”. Os processadores de acordo com a lei de Moore dobram de capacidade a cada 18 meses, as baterias definitivamente não. Em termos gerais, Armand e Tarascon (2008) aponta que este “lento progresso é devido à falta de materiais de eletrodos e eletrólitos adequados em conjunto com dificuldades em dominar as interfaces entre eles” (Tradução livre).

## Interação Humano-Bateria nos dispositivos móveis

Rahmati, Qian e Zhong (2007) cunhou o termo HBI, a sigla para *human-battery interaction* ou interação humano-bateria em português, para designar as interações ou comportamento dos usuários de dispositivos móveis em relação ao gerenciamento do recurso energético dos aparelhos. Rahmati, Qian e Zhong (2007) também cita que “80% dos usuários de sua pesquisa tomam várias medidas para aumentar o tempo de vida de suas baterias” (Tradução livre). Essas medidas são ações ativadas através da interpretação dos sinais de nível energético do sistema operacional, guiadas por objetivos e conhecimentos práticos e realizadas com a utilização de interfaces para gerenciamento das configurações das funcionalidades. A figura 2.10 demonstra o modelo conceitual da HBI.

Figura 2.10: Modelo conceitual da HBI proposto em Rahmati, Qian e Zhong (2007). Fonte: Rahmati, Qian e Zhong (2007).



Através de um estudo com 4 mil participantes durante 4 semanas acerca do comportamento de carregamento das baterias pelos usuários, Ferreira, Dey e Kostakos (2011) definiu que estes:

- ◇ Demonstram um comportamento de carregamento das baterias sistemático porém as vezes errático (este último algumas vezes explica-se por um uso constante da conexão USB com um PC, que também fornece carga à bateria)
- ◇ Preferem interromper ciclos de carregamento e conseqüentemente encurtar a validade das baterias.
- ◇ Não deixam a carga de energia das baterias ficar abaixo dos 30%, possivelmente por causa dos avisos automáticos de bateria baixa dos sistemas.
- ◇ Repetidamente deixam seus aparelhos conectados à fonte de energia após uma carga completa de suas baterias, especialmente durante à noite.

Alguns tipos de comportamento dos usuários podem ser destacados e classificados, como a classificação em dois tipos mostrada em Rahmati, Qian e Zhong (2007):

- ◇ **Tipo A:** usuários que carregam as baterias de seus aparelhos independentemente do nível de carga, por exemplo a cada 1 ou 2 dias ou qualquer intervalo fixo conveniente em sua rotina.
- ◇ **Tipo B:** usuários que realizam o carregamento baseados nas informações e notificações sobre o nível de bateria fornecidos pelo sistema operacional.

Nota-se que as características citadas em Ferreira, Dey e Kostakos (2011), listadas anteriormente, encaixam-se e reforçam a descrição de Rahmati, Qian e Zhong (2007) apresentada anteriormente.

Outra classificação é proposta em Oliver e Keshav (2010), através de um estudo com 15500 usuários de *smartphones*. Esta classificação sobrepõem com maiores detalhes a classificação anterior de Rahmati, Qian e Zhong (2007) sugerindo os três tipos a seguir:

- ◇ **Oportunista:** é o tipo mais comum de usuário, representando 63% da população estudada. É bastante caracterizado por carregamentos frequentes e de curta duração, dispersados durante o horário comercial. É o tipo que mais consome energia, aproximadamente 4.8% do total de carga por hora.
- ◇ **De baixo consumo:** é o tipo que menos consome energia, esse usuário representa aproximadamente 20% da população do estudo. A média de duração dos carregamentos deste usuário é maior que a do **Oportunista**, porém a carga de sua bateria dura mais.
- ◇ **Noturno:** este tipo representa 17% da população analisada. Ele inicia suas cargas entre os períodos de 10 às 11 horas da noite, provavelmente antes de dormir, seus períodos de carregamento são mais longos e ele consomem a energia das baterias à uma taxa muito próxima ao do tipo **De baixo consumo**. Normalmente carrega seu aparelho diariamente.

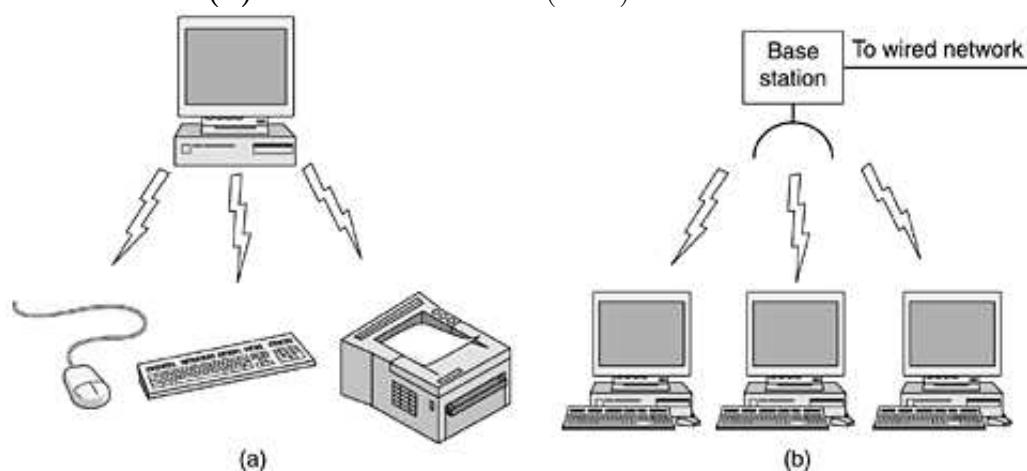
## 2.4.2 Conectividade

Em 1970, para conectar os campi da Universidade do Havaí espalhados em quatro ilhas, a primeira rede sem fio foi criada. Posteriormente na década de 1980 as redes sem fio surgem na computação pessoal (SOUZA; SILVA; AES, 2009). Ainda Engst e Fleischman (2005 apud SOUZA; SILVA; AES, 2009) define que:

As redes wireless seguem as mesmas características de todos os dispositivos sem fio. Um transceptor envia sinais através de ondas de radiação eletromagnética, que se propagam a partir de uma antena que recebe estes sinais propagados nas frequências corretas.

As redes sem fio podem ser classificadas em três tipos: redes para interconexão de sistemas, redes LAN sem fios e redes WAN sem fios. As redes para interconexão de sistemas caracterizam-se por interconectar componentes eletrônicos usando rádio de alcance limitado, como por exemplo teclados e mouse sem fios conectados em um computador, como ilustrado na Figura 2.11 (TANENBAUM, 2003). As redes LAN sem fios, assim como as redes LAN com fios, são conexões diretas ou através de um ponto de acesso comum entre sistemas computacionais, ver figura 2.11.

Figura 2.11: Configurações das **redes para interconexão de sistemas (a)** e **LAN sem fios (b)**. Fonte: Tanenbaum (2003).



O último tipo de rede sem fio, as redes WAN são definidas pela conexão de sistemas espalhados geograficamente em uma ampla malha de cobertura, um exemplo de uma rede deste tipo são as redes de rádio de telefonia celular e internet móvel (TANENBAUM, 2003).

Após um período onde os fabricantes dos equipamentos para comunicação de sistemas em redes sem fio proliferavam padrões próprios dificultando a transmutabilidade de dispositivos por estas redes, um protocolo padrão foi especialmente desenvolvido pela IEEE juntamente com a indústria para padronizar as LANs sem fio e nomeado como 802.11 (TANENBAUM, 2003). Este importante padrão moderno propõe funcionar em redes com ou sem a presença de bases fixas, chamadas de pontos de acesso, onde toda comunicação passa por estes componentes. No caso onde não há bases fixas, os computadores comunicam-se diretamente, formando as redes popularmente chamadas *Ad hoc*. O 802.11 define a banda de frequência de operação das transmissões (faixas livres em todo mundo), aspectos de segurança devido ao acesso físico livre, aspectos de saúde (principalmente pela exposição das ondas de rádio) e comerciais (como viabilidade das larguras de banda) (TANENBAUM, 2003).

Em 1999, dois anos após a aprovação do padrão 802.11, foi criada atual a Wi-Fi Alliance, uma organização sem fins lucrativos criada para popularizar a utilização das redes sem fio, certificando os dispositivos e tecnologias criadas a partir do padrão 802.11 e seus posteriores (ALLIANCE, 2013). O padrão 802.11 ou as redes sem fio em geral são popularmente chamados de Wi-Fi, apesar de o termo ter sido criado para designar o selo de qualidade e o grupo em si.

### **Comportamento dos usuários**

Assim como no caso das baterias existe uma relação de hábito dos usuários e a conectividade com as redes sem fio, ou mais especificamente com o uso da internet nos dispositivos móveis. Trestian et al. (2009) um estudo de análise acerca do comportamento de 280 mil usuários de uma rede de telefonia móvel metropolitana, destaca por exemplo que 70% da população de seu estudo visita pelo menos um

localidade comum todos os dias que acessam a rede móvel, sugerindo que os usuários repetidamente revisitam seus locais usuais.

Ainda segundo Trestian et al. (2009) em suas conclusões lista que os usuários de seu estudo passam a maior parte do tempo estacionados em zonas fixas, chamadas de zonas de conforto, como a casa ou o trabalho. Outro apontamento dos autores de Trestian et al. (2009) é que fora desta zona de conforto a utilização de aplicativos que consomem bastante bateria e banda de internet diminui consideravelmente, além disso usuários que saem destas zonas de conforto geralmente continuam conectados as redes de internet móvel para acesso à sites de redes sociais, e-mail e notícias.

Outro artigo, Nicholson e Noble (2008) descreve o desenvolvimento de um sistema de previsão da qualidade da conectividade através de um histórico de mobilidade do usuário entre as redes juntamente com as características de cada conexão. Para os autores de (NICHOLSON; NOBLE, 2008) conseguir a tendência destes dados com pouco volume de dados para treinamento foi uma tarefa facilitada pelo fato de as pessoas serem criaturas de hábitos.

## 2.5 Ferramentas básicas de desenvolvimento

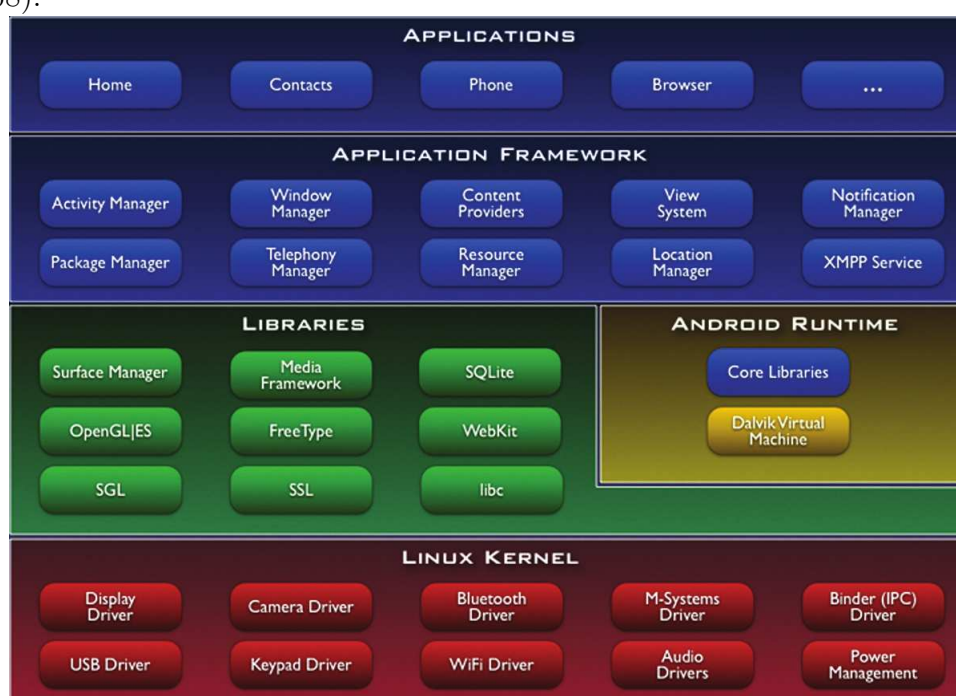
Duas ferramentas de desenvolvimento foram utilizadas nesta monografia: a plataforma Android para o monitor de dados e o ambiente de testes e a plataforma RubyOnRails para desenvolvimento do servidor web, onde os dados dos dois aplicativos Android eram recebidos, armazenados e visualizados. Nesta seção serão apresentados os detalhes técnicos destas duas plataformas.

### 2.5.1 Android

O Android é um sistema operacional para dispositivos móveis que utiliza o *kernel* 2.6 do sistema operacional Linux. Os artefatos das camadas *applications* e *application framework* vistas na Figura 2.12 são desenvolvidos na linguagem de

programação Java. Entretanto uma outra máquina virtual otimizada para o ambiente dos dispositivos móveis, a Dalvik foi construída para substituir a JVM (Java Virtual Machine) (LECHETA, 2010). A plataforma é escrita em código-fonte aberto, em sua maioria sob a licença Apache Software License versão 2.0, porém com algumas partes licenciadas em GPL versão 2 como é o caso dos pacotes do *kernel* do Linux (AOSP, 2013).

Figura 2.12: As camadas arquiteturais da plataforma Android. Fonte: Saha (2008).



Duas classes da plataforma Android, pertencentes a camada *application framework*, foram bastante utilizadas nos aplicativos construídos para esta monografia.

Uma delas é a *android.content.BroadcastReceiver*, uma classe utilizada para a captura de mensagens empacotadas em objetos chamados *Intents* enviados pelo sistema operacional ou outros aplicativos (ANDROID, 2013). O único uso desta classe pelos projetos desta monografia foi a captura de eventos enviados pelo SO, como eventos de conexão com fonte de energia, conexão com redes sem fio e desligamento ou reinicialização do sistema.

Quando um determinado aplicativo registra sua intenção de captura de determinado tipo de evento, este passa a recebê-los em um método da classe *BroadcastReceiver* relacionada. Os Algoritmos 2.1 e 2.2 mostram os trechos de códigos necessários para a captura de eventos de bateria no monitor de dados construído nesta monografia.

**Algoritmo 2.1:** Registrando a intenção de captura de um evento do sistema

```

1 IntentFilter batteryReceiverIntent = new IntentFilter();
2 batteryReceiverIntent.addAction(Intent.ACTION_POWER_CONNECTED);
3 batteryReceiverIntent.addAction(Intent.ACTION_POWER_DISCONNECTED);
4 this.registerReceiver(batteryReceiver, batteryReceiverIntent);

```

**Algoritmo 2.2:** Código de um *BroadcastReceiver* recebendo mensagens das mudanças no estado da bateria

```

1 public class BatteryReceiver extends BroadcastReceiver {
2     @Override
3     public void onReceive(Context context, Intent intent) {
4         String action = intent.getAction();
5         if(action.equals(Intent.ACTION_POWER_CONNECTED)){
6             new MonitorController(context)
7                 .initAsyncEmpathyMonitorScan(BATTERY, POWER_CONNECTED);
8         }
9         else if(action.equals(Intent.ACTION_POWER_DISCONNECTED)){
10            new MonitorController(context)
11                .initAsyncEmpathyMonitorScan(BATTERY, POWER_DISCONNECTED);
12        }
13    }
14 }

```

A segunda classe da plataforma Android mais importante nos projetos deste trabalho é a *android.app.Service* que representa de forma simples um serviço. Esta classe é descrita pela documentação Android (2013) como sendo um “componente

de aplicação que pode executar longas operações em segundo plano e não provê uma interface de usuário” (Tradução livre).

Um serviço pode inicializar e continuar executando mesmo sem nenhuma intervenção do usuário. Realizar downloads de arquivos da internet em segundo plano e de forma assíncrona é um típico requisito de *software* que pode utilizar a classe *android.app.Service*.

A monitoração dos dados de bateria e conectividade proposta por esta monografia foi realizada em um serviço, inicializado logo após a instalação e sempre na inicialização do SO, e que somente executava alguma tarefa caso os eventos de bateria ou conectividade fossem enviados pelo sistema e capturados pelos já citados *BroadcastReceiver's* do monitor. O Algoritmo 2.3, exemplo retirado do monitor de dados Android deste trabalho, mostra o principal método desta classe.

**Algoritmo 2.3:** Método chamado sempre que o serviço de monitoração é iniciado

```
1 @Override
2 public int onStartCommand(Intent intent, int flags, int startId) {
3     registerMonitorReceivers();
4     if(isToDoTheBootScan()){
5         doTheBootScan();
6     }
7     return START_STICKY;
8 }
```

## 2.5.2 RubyOnRails

O RubyOnRails, ou somente Rails, é um *framework* de aplicação WEB para a linguagem de programação Ruby. Seu código-fonte é aberto (licença MIT) e multi-plataforma (CARNEIRO; BARAZI, 2010). Ruby é totalmente orientada à objetos e definida por Ruby (2013), a comunidade que a desenvolve como:

Uma linguagem de programação de código aberto, dinâmica e com foco na simplicidade e produtividade. Ela possui uma sintaxe elegante que é natural para leitura e fácil para a escrita.

Carneiro e Barazi (2010) mostra que este *framework* usa o paradigma de *design* de convenção em vez de configuração (COC, *convention over configuration*), reduzindo a quantidade de decisões de baixo nível que devem ser tomadas pelos programadores. Um conjunto de decisões automáticas sobre o funcionamento do sistema, como por exemplo roteamento de páginas através de convenções de nomenclatura, permitem a construção ágil de aplicativos WEB.

O Rails adota o padrão de arquitetura MVC (*Model View Controller*), dividindo a lógica do sistema em 3 camadas: interface (*view*), controle das ações (*controller*) e lógicas de negócio e dados (*model*).

O Rails contém uma biblioteca suportando cada uma das camadas do MVC. Carneiro e Barazi (2010) descreve cada uma delas:

- ◇ ***ActiveRecord***: biblioteca que manipula a abstração e interação com o banco de dados.
- ◇ ***ActionView***: sistema de *templating* que gera os documentos HTML que o visitante recebe como resultado de uma requisição para uma aplicação Rails.
- ◇ ***ActionController***: biblioteca para manipulação tanto do fluxo da aplicação, quanto dos dados vindos do banco de dados para serem exibidos em uma *view*.

# Capítulo 3

## Estado da arte

*Neste capítulo são descritos alguns trabalhos que possuem relação com o problema desta monografia. Estes são apresentados em 3 temas no contexto de dispositivos móveis: estudos de campo, comportamento dos usuários e previsão em séries temporais. Ao final são descritas as oportunidades de pesquisa encontradas.*

## 3.1 Trabalhos relacionados

Uma revisão dos trabalhos disponíveis na literatura acerca do tema otimização em dispositivos móveis e previsão em séries temporais foi realizada para identificar os aspectos e problemas que compõem este cenário. Os trabalhos relacionados à esta monografia podem ser classificados em 3 grupos: **estudos de campo**, **comportamento dos usuários** e **previsão em séries temporais**.

### 3.1.1 Estudos de campo

Os seguintes artigos indicaram bons caminhos para construção do monitor de bateria e conectividade apresentado nesta monografia.

Oliver (2010a) apresenta uma pesquisa de campo onde as características de consumo de energia e carregamento de 17300 usuários de *smartphones* BlackBerry foram monitoradas e gravadas utilizando-se um *software* próprio embarcado nos aparelhos. Posteriormente para complementar o artigo anterior, Oliver (2010b) listou as dificuldades na realização de pesquisas de larga escala com usuários de *smartphones*.

Ferreira, Dey e Kostakos (2011) descreve um grande estudo com 4035 participantes anônimos durante 4 semanas realizado através de uma aplicação de monitoração distribuída como um aplicativo da loja oficial do Google para o Android. Os autores de Jones e Harrison (2012) descrevem em um artigo a construção do *Open Battery*, um monitor de dados de bateria para Android com código aberto.

Shepard et al. (2011) descreve uma metodologia para monitoração de dados (do uso de *smartphones* e de redes sem fio) por longos períodos, utilizando um aplicativo reprogramável para captura dos dados. Estes autores também discutem os desafios para a proteção da privacidade dos participantes e sobre o impacto energético do monitor.

### 3.1.2 Comportamento dos usuários

Os trabalhos citados a seguir demonstraram a previsibilidade do comportamento do usuário na interação com a bateria e redes sem fio ajudando na definição dos dados a serem utilizados no estudo comparativo desta monografia.

O artigo Ferreira, Dey e Kostakos (2011) discute os resultados de seu estudo de campo destacando a unicidade dos usuários e consequente necessidade das soluções de otimização aprenderem e se adaptarem ao comportamento do usuário. Os autores ainda identificaram padrões entre os diferentes participantes, demonstrando a previsibilidade do comportamento destes no carregamento das baterias. Ainda em Vallina-Rodriguez et al. (2010) é conduzido um estudo similar que leva as mesmas conclusões, este estudo mostra que alguns participantes partilham de comportamento homogêneo e outros não possuem um padrão definido.

Em Nicholson e Noble (2008) as pessoas são descritas como criaturas de hábitos que repetem os mesmos caminhos diariamente. Essa hipótese acerca da previsibilidade do comportamento dos usuários em relação a conectividade de redes sem fio como Wi-Fi e 3G é confirmada nos testes do protótipo de um aplicativo que monitora e prevê o movimento do usuário pelas redes *wireless*. Estes testes que duraram 2 semanas no total (uma das semanas utilizada para treinamento) mostraram uma precisão de 90% do modelo de previsão binária de conectividade proposto no mesmo trabalho.

### 3.1.3 Previsão em séries temporais

As pesquisas que se seguem utilizam métodos de previsão em séries temporais em soluções embarcadas em dispositivos móveis.

Os autores do artigo Kang et al. (2008) propuseram um sistema para previsão do tempo de vida da bateria em dispositivos móveis utilizando o método de suavização exponencial de Holt-Winters (descrito na Seção 2.3.4) em séries temporais do padrão de uso (tempo x gasto energético em um determinado contexto de uso).

Propondo uma solução relacionada à saúde, Khalil e Sufi (2008) descreve um sistema composto por um monitor cardíaco e um *smartphone* que prevê anomalias cardíacas e notifica os médicos especialistas. Para realizar as previsões os autores utilizaram o método de suavização exponencial linear de Holt descrito na Seção 2.3.4.

É desenvolvido no artigo Chang et al. (2012) o PAFCL (*Prediction Algorithm with Fixed Cycle Length*), um novo método para previsão em séries temporais do padrão de uso de aplicativos móveis otimizado para o ambiente móvel. Estes autores afirmam que os métodos tradicionais de previsão em séries temporais são muito complexos para serem executados em tempo real diretamente em dispositivos móveis e que novos métodos adaptados à este ambiente são urgentemente necessários, apesar dos trabalhos Kang et al. (2008) e Khalil e Sufi (2008) já servirem como exemplos contrários.

## 3.2 Conclusões parciais e oportunidades de pesquisa

Alguns trabalhos que propuseram soluções ou estudos na área de otimização energética em dispositivos móveis destacaram também a importância de soluções adaptadas em tempo real ao comportamento dos usuários, para isso é necessário mapear e prever as ações destes usuários. Então mais esforços de pesquisa são necessários para estabelecer os métodos de previsão em séries temporais como opções ferramentais a estas soluções embarcadas em dispositivos móveis. Assim os trabalhos que utilizam tais métodos neste novo ambiente, precisam acrescentar em seus escopos uma análise computacional e energética destes algoritmos.

Este tipo de resultado não foi encontrado, por exemplo nos artigos de Kang et al. (2008), Khalil e Sufi (2008) e Chang et al. (2012), este último por exemplo, não apresenta a relevância computacional do novo método proposto tampouco os desempenhos computacionais que indiquem a inviabilidade dos métodos clássicos.

Para finalizar também não foram encontrados estudos comparativos ou análises computacionais dos métodos de previsão em séries temporais tradicionais para dispositivos móveis, estes estudos concentram e suprem a faltam de informação deste tipo nos demais trabalhos. No momento de definição de qual método utilizar uma visão sistemática e geral é importante para assegurar uma boa escolha e otimizar recursos de pesquisa.

Assim esta monografia propõe assumir este papel como um estudo comparativo de alguns métodos já estabelecidos para previsão em séries temporais e fornecer esse conhecimento prévio e sistemático do desempenho computacional destes.

## Capítulo 4

# Projeto experimental

*Nesta parte da monografia serão abordados os detalhes práticos do projeto: o planejamento e os aplicativos construídos para a execução dos experimentos, além da descrição do ambiente de testes*

## 4.1 Visão geral do projeto

A definição de utilizar-se dados relacionados ao comportamento dos usuários no carregamento das baterias e na conectividade Wi-Fi e 3G neste projeto se deu pela importância destes dois contextos no uso dos dispositivos móveis, apontada nas Seções 2.4.1 e 2.4.2 respectivamente. Uma revisão da literatura relacionada à otimização energética em dispositivos móveis, como é feita no Capítulo 3 desta monografia e especialmente no artigo Vallina-Rodriguez e Crowcroft (2013), demonstra que esses aspectos (interação humano-bateria e dispositivos de comunicação sem-fio) são os mais problemáticos e os que mais apresentam espaço de avanço nesta área.

Outra escolha de projeto, a utilização da plataforma Android como ambiente de testes e implementação dos aplicativos desta monografia teve dois motivos. Primeiro o fato de ser uma plataforma aberta, bem documentada, com uma comunidade muito ativa e suportada por grandes empresas. O outro motivo vem de seu grande potencial de alcance, por ser um sistema operacional utilizado por diversas empresas em diversos tipos de dispositivos móveis: *smartphones*, *tablets*, *netbooks*, entre outros, com alguns aparelhos em preços acessíveis que levaram a uma grande popularização do sistema, que hoje detém 79% da fatia do mercado de *smartphones* segundo IDC (2013).

Este estudo comparativo necessitou de dois grupos de *software*. O primeiro grupo com artefatos responsáveis pela captura e gravação do comportamento de carregamento de bateria e conectividade com a internet de usuários reais de *smartphones* e o segundo grupo com partes relacionadas à execução dos algoritmos de previsão e dos passos necessários à estes testes. Tais grupos e suas partes internas serão descritos nas três seções seguintes.

## 4.2 Monitor de bateria e conectividade

O *Open Battery*, um monitor de dados de bateria para Android disponibilizado para download em sua loja oficial, foi construído e detalhado no artigo de Jones e Harrison (2012). É um projeto com código aberto e os dados recolhidos na pesquisa também estão disponíveis para *download* e análise. Porém este foi o único trabalho similar encontrado que disponibilizou os dados da pesquisa e o código-fonte do monitor de dados.

Pelo fato de o trabalho acima não ter monitorado dados de conectividade, para o projeto desta monografia foi escolhido construir um monitor de dados de bateria e conectividade próprio, para ter maior controle dos dados a serem coletados e do formato destes.

### 4.2.1 Dificuldades de um estudo de campo

Diversas dificuldades são encontradas ao se construir um monitor de dados embarcado em *smartphones* a ser utilizado em uma pesquisa de campo sobre o comportamento dos usuários. Especificamente em relação à captura de dados de bateria os trabalhos de Jones e Harrison (2012) e Oliver (2010b) destacam diversos aspectos a serem levados em conta quando se constrói aplicativos para tal objetivo, a seguir serão abordados os itens que foram aplicados ao monitor de dados desta monografia:

- ◇ **Sistemas de arquivos voláteis:** A gravação dos dados monitorados em arquivos é necessária para assegurar armazenamento persistente em caso de desligamento do aparelho ou outras interrupções. Porém em relação aos *smartphones* os sistemas de arquivos (cartões de memórias) podem ser desmontados a qualquer momento, manualmente pelo usuário ou automaticamente pelo sistema operacional quando o dispositivo é conectado à um computador via USB por exemplo. Em relação ao monitor desta monografia, o sistema de armazenamento utilizado foi um SGBD simples (SQLite)

integrado a plataforma. Os arquivos deste banco de dados são mantidos em segurança pelo Sistema Operacional, fora do alcance de terceiros e não estão armazenados nos cartões de memória desmontáveis.

- ◇ **Restrições energéticas:** aplicativos de monitoração não devem influenciar o comportamento do usuário, ou seja, aplicativos que monitoram a interação dos usuários com suas baterias, não podem causar um impacto energético notável no sistema. Para isso o monitor de bateria desta monografia implementou uma captura orientada a eventos e somente envia os dados capturados para o servidor durante um carregamento da bateria juntamente com uma conectividade com a rede Wi-Fi.
- ◇ **Influência por aplicativos de terceiros:** outros aplicativos instalados, maliciosos ou não, podem alterar o funcionamento normal do sistema e portanto o comportamento do usuário. O trabalho de Oliver (2010b) sugere ser feita uma gravação de quais aplicativos estão instalados no aparelho para posterior análise. Não foi possível por falta de recurso, implementar tal requisito no monitor desta monografia. Porém será feita uma filtragem criteriosa dos dados monitorados na tentativa de detectar e remover estes dados influenciados do grupo que será utilizado nos testes comparativos.
- ◇ **Tempo não linear:** A sincronização do relógio do aparelho é apresentada como o aspecto mais complicado em monitorações autônomas. Uma mudança manual ou automática errônea no horário e data é muito difícil de ser detectada automaticamente. Além das alterações no relógio, um estudo global possui o complicador do fuso horário. O fuso horário foi gerenciado pelo monitor desta monografia através do uso dos valores UTC nos tempos gravados em cada entrada monitorada. Outro ponto importante, apresentado por Jones e Harrison (2012), é o desligamento por longos períodos do aparelho. Para isso o autor sugere a captura de tais eventos e posterior gerenciamento de tais períodos com o objetivo de garantir a consistência das análises dos dados capturados. Esta sugestão também foi adotada no monitor desta monografia.

- ◇ **Usuários mal intencionados:** Parece não ser comum ou de interesse de outras pessoas sabotarem trabalhos acadêmicos, porém é necessário tomar as devidas precauções em relação às alterações intencionais no resultado da pesquisa. Garantir a segurança do armazenamento e transporte dos dados e detectar de alguma forma simulações de participação. O autor Oliver (2010b) sugere uma gravação e análise dos endereços IP dos participantes, verificando participantes com mesmo endereço, ou a análise rotineira dos dados para detectar anormalidades, esta última sugestão foi a adotada pelo autor desta monografia para este problema.
- ◇ **Erros de programação e diferenças nos aparelhos:** Apesar de a plataforma de desenvolvimento para Android tratar de maneira transparente e com padronização de código as diferenças nos aparelhos, implementações e alterações equivocadas por parte dos fabricantes destes dispositivos e erros de desenvolvimento podem causar problemas na monitoração dos dados. A única forma encontrada de detectar tais anormalidades foi também a verificação diária dos dados.

Com atenção à estes fatores o monitor de dados de bateria e conectividade desta monografia foi desenvolvido para a plataforma Android, construído com o SDK versão 4.2.2 e publicado na loja de aplicativos oficial da plataforma, o Google Play, durante cinco meses em publicação foram sete versões de atualização com melhorias e correções. O monitor estava disponível para instalação em todos os aparelhos com o sistema operacional Android de versão 2.2 para cima.

#### 4.2.2 Descrição do monitor

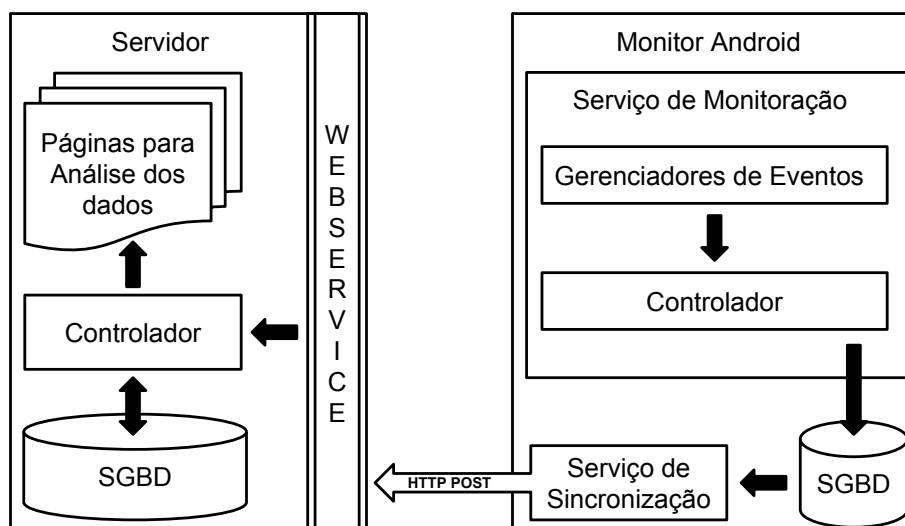
O aplicativo possui apenas uma tela para acesso pelo usuário, o resto de sua composição funciona em segundo plano. O serviço de monitoração é ativado assim que o aplicativo é aberto pela primeira vez ou o sistema operacional é reiniciado. O usuário tem acesso aos termos de participação para leitura, o endereço da página do projeto na internet, e o endereço de uma página com seus dados já enviados

para o servidor do monitor, cada usuário tem acesso somente à seus dados. Assim que o aplicativo é desinstalado, nenhum dado pode ser mais monitorado e enviado.

O monitor é constituído de dois serviços, executando em segundo plano sempre que determinados eventos são detectados. De forma mais simples um dos serviços é responsável pelo envio dos dados monitorados para o servidor e executa sempre que o aparelho encontra-se conectado à internet através de uma rede Wi-Fi e à uma fonte de energia. Os dados enviados ao servidor com sucesso são imediatamente apagados do armazenamento local do aparelho participante.

O segundo serviço, responsável pela monitoração e gravação dos dados, captura os eventos de conexão e desconexão com uma fonte de energia (via USB, conector AC, conector veicular ou fonte sem fio), os eventos de conexão e desconexão com redes sem fio (redes de telefonia e Wi-Fi) e também os eventos desligar e ligar do aparelho. Cada evento capturado é gravado imediatamente na base de dados local. A Figura 4.1, representa uma visão geral da arquitetura do monitor (cliente e servidor) com a navegabilidade dos dados monitorados representada pelas setas.

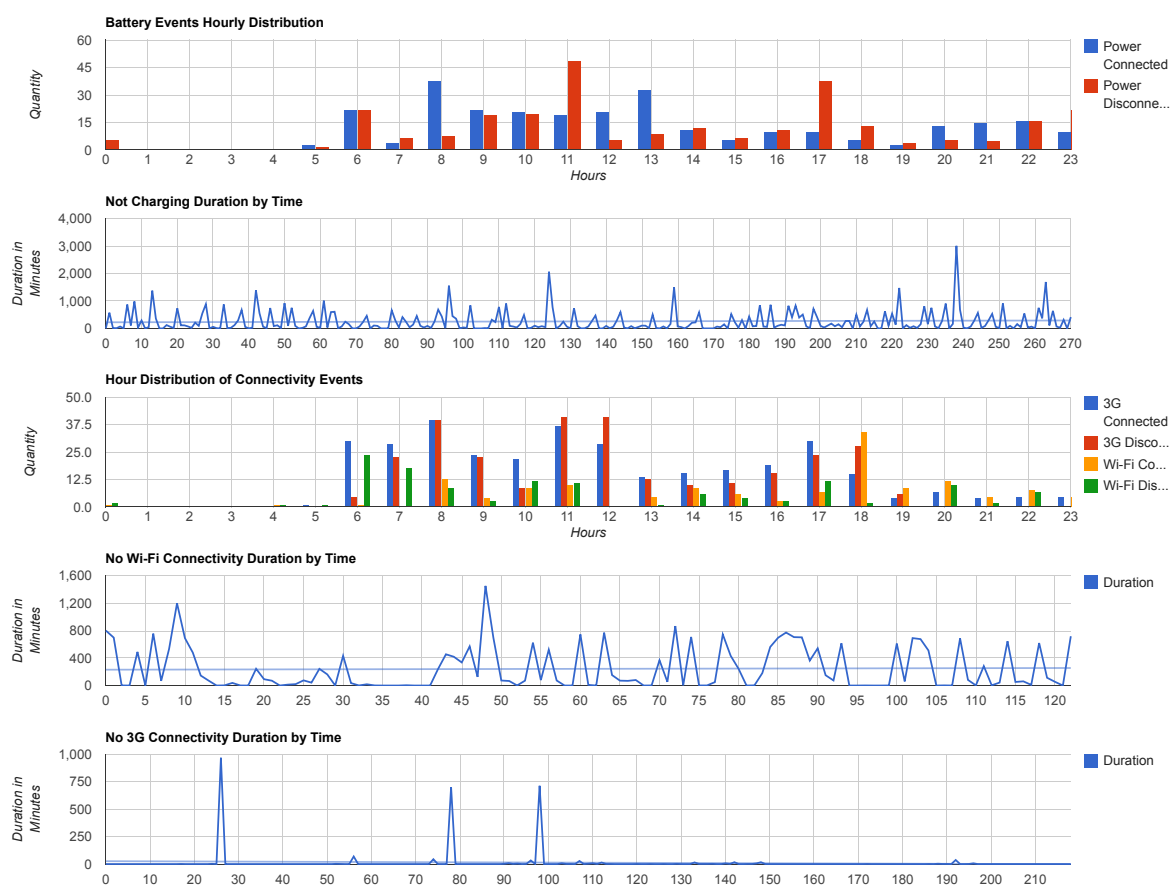
Figura 4.1: Arquitetura do monitor de bateria e conectividade. Fonte: Elaborada pelo autor.



### 4.2.3 Dados capturados pelo monitor

Os dados capturados, gravados e enviados para o servidor pelo monitor de bateria são a data e hora de captura do evento, a classe do evento (bateria ou conectividade), o tipo do evento (conexão ou desconexão), a fonte do evento (cabo USB, cabo AC, rede Wi-Fi ou 3G), o fuso horário UTC e o código de identificação única do aparelho participante (IMEI, CDMA, IMSI, Android ID ou um UUID gerado).

Figura 4.2: Dados de um participante da pesquisa de campo. Fonte: Elaborada pelo autor.



A Figura 4.2 mostra de forma gráfica os dados capturados de um dos participantes anônimos. Este são dados de 68 dias de participação entre o primeiro e

último evento enviado, foram capturados 566 eventos de bateria e 920 eventos de conectividade com as redes sem fio, um total de 1501 eventos capturados deste usuário. Esta representação é a mesma disponibilizada para os usuários participantes visualizarem seus dados recolhidos (uma participação transparente) e utilizada pelo autor desta monografia para controle da pesquisa.

O primeiro e terceiro gráficos da Figura 4.2 representam a distribuição por hora do dia dos eventos capturados, de bateria e conectividade respectivamente. Nota-se que este usuário possui possivelmente o comportamento mais comum e esperado (como visto nas Seções 2.4.1, 2.4.2 e 3.1.2), um hábito diurno das 6 horas da manhã até as 22 horas da noite, com mais ocorrências durante o horário comercial. Os outros três gráficos são séries temporais da duração em minutos dos períodos de desconexão com uma fonte de energia, com a rede Wi-Fi e com redes móveis, respectivamente à ordem em que se apresentam na Figura 4.2.

A reta em azul claro nos três gráficos da Figura 4.2 representa a média das séries (componente explicada na Seção 2.1.2), nota-se nas séries de bateria e rede Wi-Fi uma média mais alta que a da série das redes móveis onde é observado uma média de valores baixíssima muito próxima de zero, o que pode representar uma maior utilização desta rede e períodos curtos de desconexão. As três séries são estacionárias, como a reta da média demonstra.

A privacidade dos dados dos usuários foi garantida pelo termo de uso que acompanha a página de download e o aplicativo. Este termo de uso pode ser lido no Apêndice A. Os códigos-fonte do monitor de dados Android e do servidor RubyOnRails estão disponíveis respectivamente nos seguintes endereços WEB: <http://code.google.com/p/empathy-monitor/> e <http://code.google.com/p/empathy-server/>.

### 4.3 Implementação dos métodos de previsão

As implementações dos métodos de previsão em séries temporais foram escritas na linguagem de programação Java, pois esta é a linguagem de desenvolvimento

de aplicativos Android. Estes códigos foram reunidos em um projeto biblioteca utilizado pelo ambiente dos testes comparativos desta monografia.

Os métodos propostos para os testes comparativos e aqui transcritos funcionam em séries com tendência e sazonalidade. A seguir as quatro implementações dos respectivos métodos serão apresentadas.

### 4.3.1 *Naïve sazonal*

A formalização do método de previsão **naïve sazonal** vista na Seção 2.3.1, foi transcrita em código Java pelo autor desta monografia no Algoritmo 4.1:

**Algoritmo 4.1:** Algoritmo do método naïve sazonal em Java

```

1  int m = 7;
2  int t = timeSeries.length-horizon;
3  for (int h = 1; h <= horizon; h++) {
4      int ft = Math.abs(t+h-(((h-1)/m)+1)*m);
5      timeSeries[t+h-1] = timeSeries[ft-1];
6  }
```

A transcrição apresenta um ponto a ser destacado: o valor da variável "m"(período sazonal) foi definido como 7, período sazonal provável de um usuário de dispositivos móveis, ou seja, uma rotina com repetições a cada 7 dias (sazonalidade semanal).

### 4.3.2 *Média móvel ponderada*

O modelo de previsão por **média móvel ponderada** visto na Seção 2.3.2, foi transcrito em código Java para o Algoritmo 4.2:

**Algoritmo 4.2:** Algoritmo do método de média móvel ponderada em Java

```

1  for (int h = 1; h <= horizon; h++) {
2      int meanSum = 0;
3      int totalWeight = 0;
4      for (int i = 0; i < n; i++) {
5          int weight = n-i;
6          meanSum += timeSeries[t-i]*weight;
7          totalWeight += weight;
8      }
9      timeSeries[t+1] = meanSum/totalWeight;
10     t++;
11 }

```

Apenas uma observação a ser feita sobre este código: o tamanho da janela de soma das médias (escolhido como 21 ou três semanas, os três últimos períodos sazonais) foi definido através da heurística indicada por Afonso (2010), apresentada na Seção 2.3.2 (nesta implementação foi definido um valor alto pois as séries temporais utilizadas nos testes possuem sazonalidade com pouca flutuação).

### 4.3.3 Por decomposição clássica

Para a implementação do método de previsão por decomposição descrito na Seção 2.3.3 foram utilizados:

- ◇ O método clássico de decomposição das componentes apresentado em 2.1.2 transcrito no Algoritmo B.1 demonstrado no Apêndice B.
- ◇ O método *naïve* sazonal para previsão da componente sazonal decomposta. Este método foi apresentado na Seção 4.3.1.
- ◇ O método de previsão exponencial linear de *Holt* (apresentado na Seção 2.3.4) aplicado na componente não-sazonal (tendência-ciclo mais erro). Este

método foi transcrito no Algoritmo 4.4 através de pequenas alterações do algoritmo Java de Chandra (2012b).

- ◇ A soma final das previsões da componente sazonal e não-sazonal mostrada no Algoritmo 4.3

**Algoritmo 4.3:** Trecho final do método de previsão por decomposição (Soma das previsões de  $S$  e  $T$ ) escrito em Java

```

1 for (int t = y.length-horizon; t < y.length; t++) {
2     y[t] = S[t] + fT[t];
3 }

```

**Algoritmo 4.4:** Algoritmo do método de previsão exponencial linear de Holt escrito em Java

```

1 //initializations
2 l[0] = fT[0] = T[k];
3 b[0] = T[k+1] - T[k];
4 fT[1] = (int) (l[0] + b[0]);
5 int yt = k+1;
6 //forecasting
7 for (; yt < T.length - horizon -k; yt++) {
8     l[yt] = alpha * T[yt] + (1 - alpha) * (l[yt - 1] + b[yt - 1]);
9     b[yt] = beta * (l[yt] - l[yt - 1]) + (1 - beta) * b[yt - 1];
10    fT[yt+1] = (int) (l[yt] + b[yt]);
11 }
12 for (int j = 0; j < horizon + k; j++, yt++) {
13     fT[yt] = (int) (l[l.length - 1] + (j + 1) * b[b.length - 1]);
14 }

```

Foram utilizados os mesmos valores de  $\alpha$  e  $\beta$  (parâmetros de suavização vistos na Seção 2.3.4 do código de Chandra (2012b), 0.3 e 0.2 respectivamente.

Por se tratar de um método mais complexo que os métodos naïve sazonal e o método por média móvel ponderada também implementados para o estudo comparativo desta monografia, uma validação da implementação deste método de previsão por decomposição foi feita através de comparação com resultados de uma ferramenta WEB já consolidada. Esta ferramenta é a Wessa (2013), utilizada em diversos trabalhos científicos que envolvem séries temporais e estatística, e que utiliza a também consolidada linguagem e ambiente de estatística R et al. (2005) para implementação de seus métodos para análise de séries temporais.

Uma mesma série temporal foi utilizada como entrada para o método de decomposição clássica desta monografia e da ferramenta Wessa (2013), tal série é a mesma mostrada na Figura 4.2 proveniente dos dados do usuário descrito na Seção 4.2.3. As decomposições foram comparadas através de médias absolutas das diferenças entre as componentes de cada implementação, que resultaram em 36,33, 7,13 e 38,46 minutos respectivamente para tendência-ciclo, sazonalidade e componente remanescente. Esta proximidade das decomposições é ilustrada através das Figuras 4.3 (decomposição desta monografia) e 4.4 (decomposição da Wessa (2013)).

Figura 4.3: Respectivamente: série original; e componentes tendência-ciclo, sazonal e remanescente, decompostas por Wessa (2013). Fonte Wessa (2013)

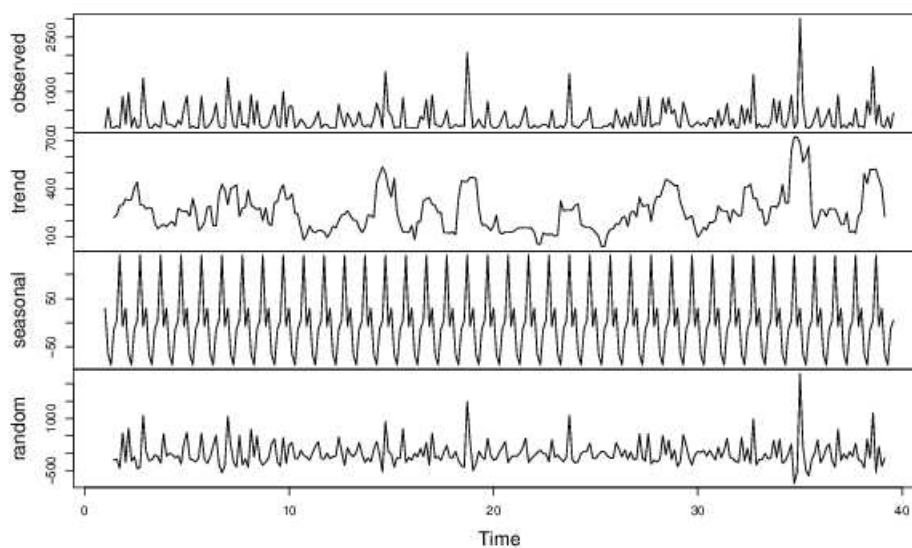
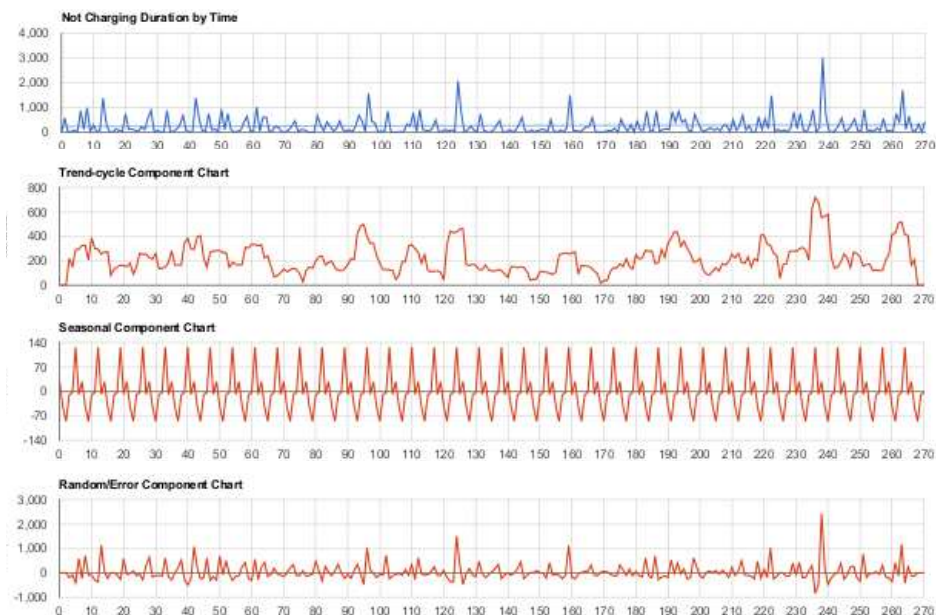


Figura 4.4: Respectivamente: série original; e componentes tendência-ciclo, sazonal e remanescente, decompostas nesta monografia. Fonte: Elaborada pelo Autor.



Esta análise trouxe maior segurança em relação a eficácia da implementação do método de previsão por decomposição desta monografia.

#### 4.3.4 Holt-Winters

Apresentado na Seção 2.3.4 o método de previsão por suavização exponencial Holt-Winters foi transcrito em sua forma **multiplicativa** para a linguagem de programação Java como mostrado no Algoritmo 4.5.

Esta implementação foi inspirada no código de Chandra (2012a), que também foi escrito em Java mas que necessitou de algumas alterações para adequar-se ao modelo formal apresentado por Hyndman e Athanasopoulos (2013), demonstrado na seção 2.3.4.

**Algoritmo 4.5:** Algoritmo do método de previsão exponencial tripla Holt-Winters escrito em Java, inspirado na implementação de Chandra (2012a)

```

1 for (int t = 1; t < Y.length; t++) {
2     // Calculate level smoothing
3     if ((t - m) >= 0) {
4         L[t]=alpha*(Y[t]/S[t-m])+(1.0-alpha)*(L[t-1]+T[t-1]);
5     }
6     else {
7         L[t]=alpha*Y[t]+(1.0-alpha)*(L[t-1]+T[t-1]);
8     }
9     // Calculate trend smoothing
10    T[t]=beta*(L[t]-L[t-1])+(1-beta)*T[t-1];
11    // Calculate seasonal smoothing
12    if ((t - m) >= 0) {
13        S[t] = gamma*(Y[t]/(L[t-1]+T[t-1]))+(1.0-gamma)*S[t-m];
14    }
15    // Calculate forecast
16    if ((t + h) >= m) {
17        F[t+h]=(int)((L[t]+(h*T[t]))*S[t-m+((h-1)% m)+1]);
18    }
19 }

```

Os trechos de código que contém a inicialização das componentes nível, sazonal e tendência-ciclo da série temporal a ser prevista estão demonstrados no apêndice C.

O mesmo método de validação da implementação apresentada na Seção 4.3.3 foi utilizado para o método Holt-Winters implementado para esta monografia. O MAE da previsão realizada pela implementação desta monografia ficou em 248 minutos e a da ferramenta Wessa (2013) em 252 minutos. O teste utilizou a mesma série temporal decomposta na Seção 4.3.3 e apresentada na Seção 4.2.3 através da figura 4.2.

Figura 4.5: Previsão Holt-Winters aplicada pela ferramenta Wessa (2013). Em vermelho a previsão, em preto a série original. Fonte: Wessa (2013).

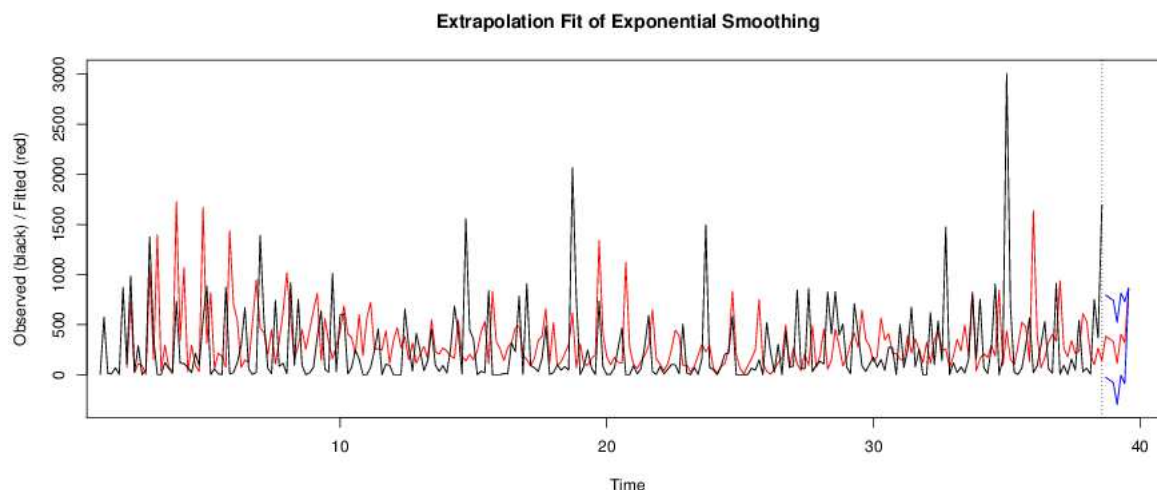
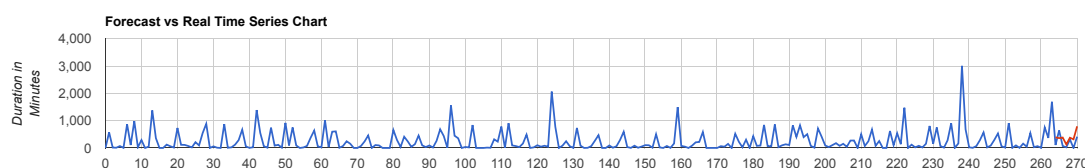


Figura 4.6: Previsão Holt-Winters aplicada pelo algoritmo desta monografia. Em vermelho a previsão, em azul a série original. Fonte: Elaborada pelo Autor.



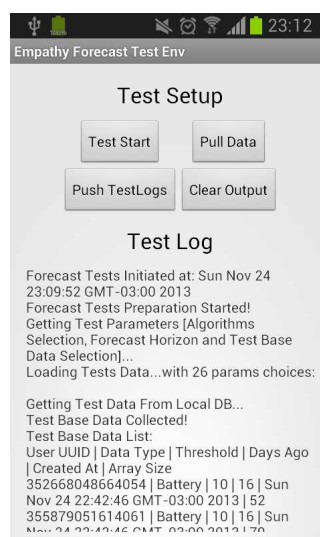
As Figuras 4.5 e 4.6 representam a mesma série temporal utilizada e a previsão (em vermelho) dos dois algoritmos neste teste de comparação, o da ferramenta Wessa (2013) e o desta monografia respectivamente.

Os valores dos fatores de suavização *alpha*, *beta* e *gamma* da implementação desta monografia foram definidos com os mesmo valores utilizados pela ferramenta Wessa (2013) no teste realizado com a série da Figura 4.5, pois a mesma usa otimização por minimização dos erros e também devido as demais séries temporais que serão utilizadas nos testes comparativos finais desta monografia possuírem componentes sazonais e de tendência-ciclo bem próximas da série utilizada na validação com a Wessa (2013).

## 4.4 Ambiente de testes

Um aplicativo Android foi desenvolvido para a execução dos testes dos algoritmos de previsão. Neste aplicativo os testes são configurados em tempo de compilação com os parâmetros de formação das séries temporais (ver Seção 4.4.1) e o algoritmo de previsão a ser testado. Este aplicativo ainda possui uma tela para controle dos testes (iniciar o carregamento das séries temporais do servidor, enviar os resultados e iniciar os testes) e um visualização em tempo real dos procedimentos, como é visto na Figura 4.7.

Figura 4.7: Ambiente dos testes comparativos para Android. Fonte: Elaborada pelo autor.



O código fonte do aplicativo construído como ambiente para os testes dos algoritmos de previsão está disponível em <http://code.google.com/p/empathy-research/> sob a licença Apache License 2.0.

### 4.4.1 Execução dos algoritmos

As entradas dos algoritmos de previsão, independentemente dos métodos utilizados, são as séries temporais (vetores de inteiros) e o horizonte de previsão (valor

inteiro positivo). A saída é um vetor com tamanho do horizonte de previsão definido, onde cada posição deste representa um horizonte intermediário, ou seja, caso o horizonte de previsão tenha valor três, serão previstos os instantes  $y_{T+1}$ ,  $y_{T+2}$  e  $y_{T+3}$ .

As séries temporais são formadas pelas durações em minutos dos períodos de desconexão com uma fonte de energia ou rede sem fio, dependendo do tipo de dado. Estas séries temporais são construídas ainda no servidor a partir dos eventos capturados e armazenados durante a pesquisa de campo. Para os cálculos de precisão dos modelos de previsão, o trecho final das séries temporais formadas foram excluídos dos testes ( $Y_{T-h}$ ), onde  $h$  é o horizonte de previsão.

A constituição das séries pode ser modificada através de dois parâmetros de filtragem. Primeiro o tipo de dado: de bateria, de conectividade Wi-Fi ou de conectividade rede móvel. Em segundo, um valor de corte que retira da série pontos abaixo deste valor, por exemplo, excluí-se todos os períodos de desconexão com menos de 15 minutos de duração.

Para a execução dos algoritmos de previsão nos testes finais definiu-se 4 (valor pequeno arbitrário) como o valor do horizonte máximo de previsão, pois aconselha-se valores baixos para esta variável devido ao crescimento significativo do erro de previsão a medida que o horizonte distancia-se do último valor observado da série. Lembrando que os horizontes de 1 à 4 serão previstos nos testes através da aplicação dos métodos de previsão para cada um destes horizontes intermediários.

#### 4.4.2 Métricas computacionais e de precisão

Durante a execução de um teste, três métricas são colhidas:

1. O tempo de execução ( $T$ ) em nanosegundos:

$$T_{total} = T_{final} - T_{inicial} \quad (4.1)$$

Onde os tempos finais e iniciais são capturados utilizando-se o método *System.nanoTime()* do Android que faz uma chamada ao relógio mais preciso do sistema. Equivalente ao *CLOCK\_MONOTONIC* do Linux.

2. A quantidade de memória gasta (M) em *kilobytes*:

$$M_{usada} = (MT_{final} - ML_{final}) - (MT_{inicial} - ML_{inicial}) \quad (4.2)$$

Onde a memória total (MT) e livre (ML) pertencem ao processo que está executando os algoritmos e são adquiridas respectivamente através dos métodos Android *Runtime.getRuntime().totalMemory()* (tamanho atual da memória *heap*) e *Runtime.getRuntime().freeMemory()* (espaço disponível da memória *heap*).

3. A energia gasta:

$$J = I_{cpu} \times V_{cpu} \times T_{exec} \quad (4.3)$$

Onde  $J$  é a energia total gasta em Joules,  $I_{cpu}$  em Amperes é a corrente elétrica da CPU em determinada frequência,  $V_{cpu}$  é a voltagem da CPU e  $T_{exec}$  é o tempo total de execução em segundos. As frequências e correntes elétricas da CPU estão descritas em arquivos do sistema operacional, a frequência atual da CPU é adquirida através de arquivos do programa *proc* (programa de monitoração do sistema) do Linux.

Por ser uma métrica mais complexa, durante o desenvolvimento do ambiente de testes, experimentos preliminares foram realizados e os valores do gasto energético destes experimentos foram comparados com as medidas de um aplicativo Android, de nome PowerTutor, um monitor de gasto energético desenvolvido em Zhang et al. (2010). A métrica de custo energético desta monografia foi em média 40% menor do que os valores apontados pelo aplicativo PowerTutor, porém esta diferença pode ser explicada pelo fato de o aplicativo PowerTutor não realizar medidas à nível de métodos, mas no máximo a nível de processos e portanto a medida de gasto energético

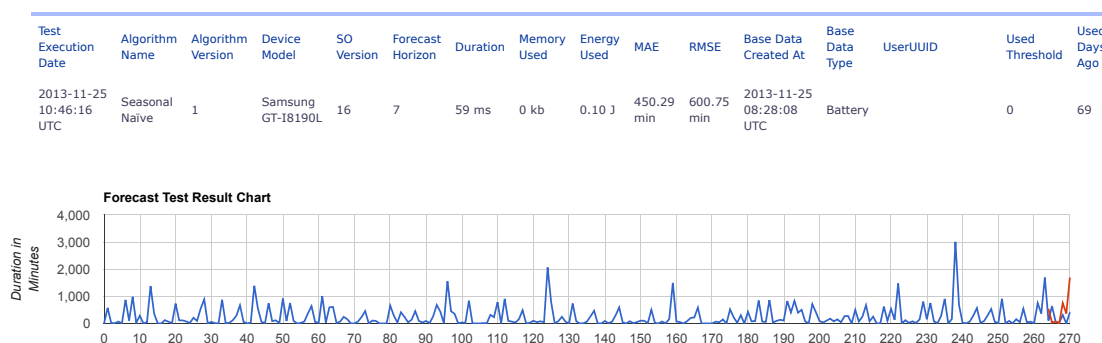
desta monografia por ser realizada à nível de método possui maior precisão para os objetivos propostos.

Em relação as **métricas de precisão**, após a execução dos algoritmos, dadas suas séries temporais com as previsões realizadas, calculamos os erros em relação aos dados reais já capturados (trechos das séries que foram excluídos propositalmente antes dos testes:  $Y_{T-h}$ ). Foram utilizados de forma distinta, dois métodos para cálculo da precisão dos modelos de previsão: o MAE (*Mean Absolute Error*) e o RMSE (*Root Mean Squared Error*) descritos na Seção 2.2.4.

### 4.4.3 Obtenção dos resultados

A cada teste uma lista de informações sobre o mesmo é armazenada para ser enviada ao servidor: ambiente de execução, os dados base utilizados, as métricas de execução e o resultado da previsão. A Figura 4.8 mostra as informações do resultado de um teste exibidas em uma página do servidor, após serem enviadas pelo ambiente de testes Android.

Figura 4.8: Resultados de um teste no servidor. Em vermelho a previsão, em azul a série original. Fonte: Elaborada pelo Autor.



#### 4.4.4 Características do ambiente de testes

A fim de diminuir o impacto energético e computacional dos objetos externos ao aplicativo de testes, foram tomadas algumas medidas possíveis de serem aplicadas em um sistema operacional Android com um usuário sem privilégios de administrador. O modo avião foi ativado, desligando todos os dispositivos de comunicação sem fio. Os dispositivos de localização foram desligados. Os testes foram executados em um serviço alguns segundos depois de a tela ser desligada através do bloqueio do sistema. O sistema foi reinicializado para configurações de fábrica e foram desinstalados todos os aplicativos permitidos, assim como foram finalizados todos os processos permitidos. Durante os testes o aparelho não esteve conectado à uma fonte de energia, porém antes dos testes sua bateria estava com carga total.

Durante os testes foram utilizados dois aparelhos Android. Um aparelho Android de modelo GT-I8190L da Samsung, com o sistema Android versão 4.1.2 (*build* JZO54K.I8190LUBAMH3), com o processador ARM Cortex A9 dual-core de 1 GHZ, memória RAM de 1GB, bateria com capacidade de 1500 mAh.

O outro aparelho Android de modelo GT-I5500 da Samsung, com o sistema Android versão 2.3.7 (*build* CyanogenMod-7-20120807-MADTEAM-galaxy5), um dispositivo com o processador Qualcomm MSM7227 Single-Core de 600Mhz, memória RAM de 256MB, bateria com capacidade de 1200 mAh. O aplicativo que executou os testes nestes dois aparelhos foi desenvolvido com a SDK Android versão 4.2.2.

#### 4.4.5 Dados utilizados nos testes

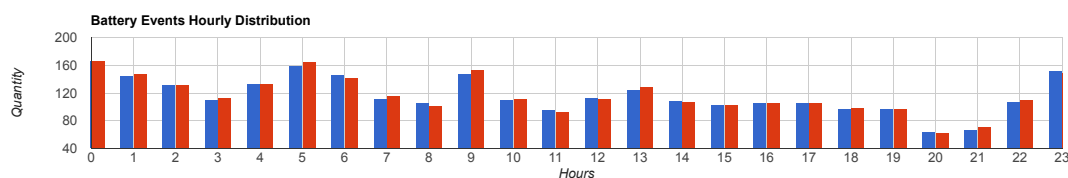
Foram 4 meses e 10 dias de duração do estudo de campo desta monografia, com um total de 120 participantes que enviaram 58213 eventos de bateria e conectividade como os descritos na Seção 2.5.1. Destes 120 participantes 43,33% enviaram seus eventos do fuso horário UTC-3 (a zona do Brasil) e o resto 56,67% enviaram de outras 17 zonas diferentes, o que mostra a distribuição geográfica do estudo.

Para os testes comparativos dos métodos de previsão não serão utilizados todos estes dados, e sim uma seleção filtrada a partir de alguns critérios, para garantir uma uniformidade e retirar possíveis ruídos provenientes de problemas nas capturas (usuários maliciosos, erros, compatibilidade, etc) citados na Seção 4.2.1.

De um modo geral foram aplicados os seguintes critérios de seleção:

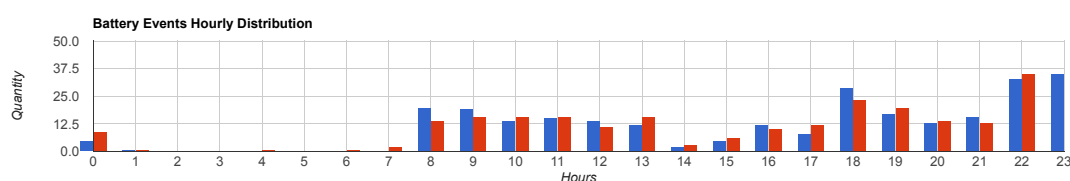
- ◇ Somente para a filtragem dos dados de bateria, analisou-se os gráficos de distribuição dos eventos por hora e foram retirados os usuários que apresentaram distribuições divergentes com os estudos de comportamento de bateria pesquisados durante esta monografia (ver Seções 2.4.1 e 3.1.2). Ou seja, usuários com distribuição uniforme em todas as horas do dia (como na Figura 4.9) para os eventos (2 usuários foram retirados da seleção final).
- ◇ As séries temporais geradas para os testes foram referentes as durações dos períodos em minutos, de desconectividade ou descarregamento, e esta geração só levou em consideração tais períodos com valores maiores que 1 minuto.
- ◇ A partir destas duas filtrações anteriores, os usuários que apresentaram tais séries temporais com poucos pontos, menos que o dobro do período sazonal mais o horizonte a ser utilizado nas previsões (7 dias, ou uma semana mais 4 do horizonte), ou seja 18 pontos, também foram retirados da população.

Figura 4.9: A distribuição por hora dos eventos de bateria de um usuário que foi removido da população dos testes. Em azul eventos de conexão com a fonte de energia, em vermelho momentos de desconexão. Fonte: Elaborada pelo Autor.



A distribuição diurna dos eventos mostrada na Figura 4.10 está mais de acordo com um dos tipos de comportamento dos usuários (como detalhado na Seção 2.4.1), atividades durante o dia e quase nenhuma atividade durante a noite (das 0 às 8 horas).

Figura 4.10: A distribuição por hora dos eventos de bateria de um dos usuários pertencentes a população selecionada. Em azul eventos de conexão com a fonte de energia, em vermelho momentos de desconexão. Fonte: Elaborada pelo Autor.



Seguindo tais critérios, a seleção final dos dados de bateria somou 29 usuários, 8387 eventos de bateria, 2215 pontos totais com uma média de 137 pontos por série temporal, em uma população de estudo geograficamente distribuída em zonas UTC (fuso horário) diferentes: UTC-3 (zona que inclui o Brasil) com 62,07% da população e 37,93% em outras zonas (8 zonas diferentes).

A seleção dos dados de conectividade Wi-Fi manteve 42 usuários, com um total de 7746 eventos, e 4649 pontos totais com uma média de 110 pontos por série formada, a população ficou distribuída em diversas zonas UTC: a zona UTC-3 com 61,90% da população e 38,10% nas demais (9 zonas).

Para os dados de conectividade 3G foram selecionados um total de 19 usuários, 2242 eventos, 801 pontos com uma média de 57 pontos por série, distribuídos geograficamente na zona UTC-3 com 57,89% da população e 42,11% em outras zonas (6 zonas diferentes).

Descritos os itens práticos e funcionais deste estudo comparativo, especialmente o planejamento dos testes finais, resta discutir a execução destes testes e analisar seus resultados.

# Capítulo 5

## Resultados experimentais

*Neste capítulo serão analisados e comparados os resultados dos quatro algoritmos de previsão em relação ao custo energético, custo computacional e precisão.*

## 5.1 Análise geral dos experimentos

De acordo com as métricas descritas na Seção 4.4.2, a média absoluta de todos os resultados dos testes ficou em 1 KB de memória utilizada, MAE de 912 minutos com 19.44% da população abaixo dos 60 minutos, RMSE de 1289 minutos. A média dos resultados da métrica de tempo de execução ficou em 0 milissegundo e consequentemente o da métrica de gasto energético se manteve em 0 Joule. Os testes realizados com o outro modelo de aparelho Android obtiveram os mesmos resultados computacionais e de previsão.

O valor de 60 minutos como limite máximo aceitável para um erro foi definido heurísticamente e contextualmente. Para o Autor desta monografia 1 hora de erro de previsão da duração do período de desconexão com uma fonte de energia ou rede sem fio é um valor funcional para por exemplo, um aplicativo que precisa definir momentos de condição energética favoráveis (e.g. conectado ao Wi-Fi e ao USB) para o envio de dados **tolerantes a atraso**. Erros de uma ou mais horas são então resultados gradualmente ruins.

De um modo geral em termos computacionais os algoritmos de previsão executaram cada teste (um teste representa a previsão em uma série temporal da população definida na Seção 4.4.5) com quase nenhuma demanda de recursos computacionais (processamento, memória e energia). Os algoritmos de previsão por suavização exponencial Holt-Winters e por decomposição clássica apresentados na Seção 2.3 requisitam demandas levemente superiores aos demais, mas mesmo assim ainda insignificantes neste contexto. Essa baixa demanda pode ser explicada pelo tamanho das séries temporais (uma média 102 pontos por série temporal), pelo horizonte de previsão estabelecido como 4 (definido e explicado na Seção 4.4.1) e pela baixa complexidade destes algoritmos.

## 5.2 Resultados

Diferentemente das médias gerais, as médias de memória utilizada por algoritmo ficaram em 3 KB de média para os algoritmos de Holt-Winters e Decomposição Clássica, e 0 KB para o naïve sazonal e para o de média móvel ponderada. As médias de tempo de execução e gasto energético ficaram com os mesmos valores das médias gerais.

Foram no total 360 testes realizados, distribuídos segundo a Tabela 5.1 demonstra, onde 19.44% destes testes tiveram métricas MAE abaixo dos 60 minutos, porém dos 116 testes realizados com dados de bateria somente 2.59% tiveram resultados MAE abaixo deste valor. Já para dados de conectividade 3G a porcentagem de testes com MAE abaixo dos 60 minutos foi de 50%. A média dos MAE de todos os testes ficou em 912,79 minutos. Em termos gerais esse valores mostram que apesar de os métodos de previsão terem baixa complexidade, principalmente para a previsão de dados de bateria, estes métodos não apresentaram erros aceitáveis para a grande maioria das aplicabilidades da previsão em séries temporais.

Tabela 5.1: Médias por tipo de dado e porcentagem dos testes com MAE abaixo dos 60 min

| Tipo de Dado        | Total de Testes | MAE < 60 min | MAE     | RMSE    |
|---------------------|-----------------|--------------|---------|---------|
| Todos               | 360             | 19,44%       | 912,79  | 1289,77 |
| Bateria             | 116             | 2.59%        | 1225,73 | 1650,77 |
| Conectividade 3G    | 76              | 50%          | 587,2   | 1023,19 |
| Conectividade Wi-Fi | 168             | 17.26%       | 844     | 1161,11 |

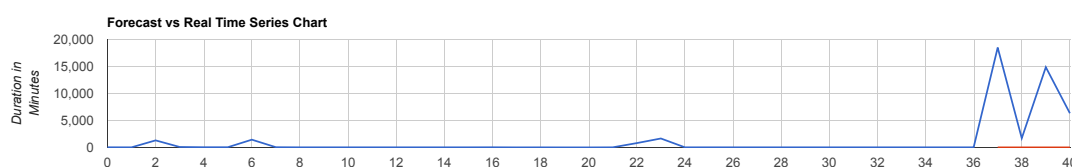
Este alto valor de erro é em grande parte influenciado pela natureza dos dados e das séries temporais (comportamento do usuários). Por exemplo, o método de Holt-Winters apresentou o maior valor MAE de todos os testes como visto na Tabela 5.2 (cada letra da coluna **Usuário** representa a mesma série ou usuário somente naquela tabela) e claramente teve seu desempenho prejudicado neste resultado específico pela série usada neste teste e que está representada graficamente na Figura 5.1. Esta série inicialmente apresenta uma componente nível de

valor baixo, nenhuma tendência e pouca sazonalidade, ou seja, em grande parte esta é uma série estacionária até que de forma explosiva muda completamente suas características no trecho final, que foi exatamente o trecho da série a ser previsto no teste.

Tabela 5.2: Os 10 maiores MAEs de todos os testes

| Algoritmo             | Usuário | Tipo de Dado        | MAE     | RMSE     |
|-----------------------|---------|---------------------|---------|----------|
| Holt-Winters          | A       | Conectividade Wi-Fi | 10332   | 12291,08 |
| Naïve Sazonal         | A       | Conectividade Wi-Fi | 10318   | 12279,32 |
| Decomposição Clássica | A       | Conectividade Wi-Fi | 10280,5 | 12213,89 |
| Média móvel ponderada | A       | Conectividade Wi-Fi | 10249   | 12220,29 |
| Decomposição Clássica | B       | Bateria             | 6765,25 | 9950,07  |
| Holt-Winters          | B       | Bateria             | 6717,75 | 9848,87  |
| Naïve Sazonal         | B       | Bateria             | 6597,75 | 9821,36  |
| Média móvel ponderada | B       | Bateria             | 6053,5  | 9030,73  |
| Média móvel ponderada | C       | Conectividade 3G    | 5020,75 | 9311,83  |
| Naïve Sazonal         | D       | Bateria             | 4977    | 7112,11  |

Figura 5.1: Previsão com pior MAE dos testes com dados de conectividade Wi-Fi. Em vermelho a previsão, em azul a série original. Fonte: Elaborada pelo Autor.



Para retirar a possibilidade de erro na implementação ou problemas nas definições dos parâmetros de suavização foi realizada uma comparação dos resultados do método Holt-Winters implementado nesta monografia com o método Holt-Winters da ferramenta Wessa (2013). Apesar de mais outros três métodos terem sido testados nesta monografia, dois deles também possuindo parâmetros de suavização, só foi possível realizar esta validação com o método Holt-Winters (único testado nesta monografia e também disponibilizado pela Wessa (2013)) pela dificuldade de encontrar-se ferramentas estabelecidas e robustas para a comparação dos outros métodos. Assim foi diminuída a possibilidade de tais problemas na

implementação e testes com o método Holt-Winters desta monografia, os resultados destes testes foram similares aos resultados da ferramenta Wessa (2013) como apresentado nas Tabelas 5.3 e 5.4

Tabela 5.3: Comparação dos maiores MAEs do Holt-Winters com o Wessa por tipo de dado

| <b>Tipo de Dado</b> | <b>MAE</b> | <b>WESSA-MAE</b> |
|---------------------|------------|------------------|
| Bateria             | 6717,75    | 6591,25          |
| Bateria             | 3359,5     | 3096,5           |
| Conectividade 3G    | 4822       | 4810,5           |
| Conectividade 3G    | 3145,25    | 2875,25          |
| Conectividade 3G    | 1975,5     | 1971,25          |
| Conectividade Wi-Fi | 10332      | 10292            |
| Conectividade Wi-Fi | 3168,75    | 2703,25          |

Tabela 5.4: Comparação dos menores MAEs do Holt-Winters com o Wessa por tipo de dado

| <b>Tipo de Dado</b> | <b>MAE</b> | <b>WESSA-MAE</b> |
|---------------------|------------|------------------|
| Bateria             | 101        | 436,25           |
| Bateria             | 95         | 78,25            |
| Bateria             | 54,5       | 77               |
| Conectividade 3G    | 3,25       | 10,5             |
| Conectividade 3G    | 3,5        | 211              |
| Conectividade 3G    | 3,75       | 2,5              |
| Conectividade Wi-Fi | 3,25       | 396              |
| Conectividade Wi-Fi | 7,25       | 24               |

Ainda para reforçar a influência da natureza destas séries temporais nos resultados ruins, foi notado que nos 10 piores resultados para cada grupo de testes por tipo de dado existe uma recorrência das mesmas séries. Nas Tabelas 5.5, 5.6, 5.7 pode ser visto que as séries dos mesmos usuários se repetem nos resultados ruins em métodos de previsão diferentes (entre uma tabela e outra as letras da coluna **Usuário** não representam os mesmos sujeitos, a correspondência fica restrita à cada tabela).

Por exemplo, o usuário A e sua correspondente série temporal na Tabela 5.5 participam dos 4 piores resultados. Isto também acontece para o caso dos 10 menores valores de cada tipo de dado e pode ser conferido nas tabelas do Apêndice D.

Tabela 5.5: Os 10 maiores MAEs dos testes de bateria

| <b>Algoritmo</b>      | <b>Usuário</b> | <b>MAE</b> | <b>RMSE</b> |
|-----------------------|----------------|------------|-------------|
| Decomposição Clássica | A              | 6765,25    | 9950,07     |
| Holt-Winters          | A              | 6717,75    | 9848,87     |
| Naïve Sazonal         | A              | 6597,75    | 9821,36     |
| Média móvel ponderada | A              | 6053,5     | 9030,73     |
| Naïve Sazonal         | B              | 4977       | 7112,11     |
| Média móvel ponderada | B              | 4500,25    | 5297,03     |
| Decomposição Clássica | B              | 4022,75    | 6578,65     |
| Holt-Winters          | C              | 3359,5     | 4531,54     |
| Naïve Sazonal         | C              | 3241,75    | 4359,5      |
| Decomposição Clássica | C              | 3096,5     | 4480,57     |

Tabela 5.6: Os 10 maiores MAEs dos testes de conectividade 3G

| <b>Algoritmo</b>      | <b>Usuário</b> | <b>MAE</b> | <b>RMSE</b> |
|-----------------------|----------------|------------|-------------|
| Média móvel ponderada | A              | 5020,75    | 9311,83     |
| Holt-Winters          | A              | 4822       | 9581,62     |
| Decomposição Clássica | A              | 4815,5     | 9581,61     |
| Naïve Sazonal         | A              | 4806,5     | 9566,56     |
| Holt-Winters          | B              | 3145,25    | 5548,31     |
| Decomposição Clássica | B              | 3018,25    | 5537,62     |
| Naïve Sazonal         | B              | 2897       | 5514,75     |
| Média móvel ponderada | B              | 2864,75    | 5393,03     |
| Decomposição Clássica | C              | 1985,75    | 2347,52     |
| Holt-Winters          | C              | 1975,5     | 2420,21     |

O bom resultado das previsões nas séries temporais dos dados de conectividade 3G descrito no primeiro parágrafo desta seção e na visto na Tabela 5.1, reforça-se ao ser notado que dos 10 testes de menores erros apresentados na Tabela 5.8, os 7 primeiros pertencem a testes realizados em séries temporais formadas com este tipo de dado, e ainda para reforçar também a importância do comportamento das

séries nestes resultados, os três primeiros resultados pertencem ao mesmo usuário (ou série do usuário). A série deste usuário **A** está representada na Figura 5.2, onde podemos ver componentes média, tendência-ciclo e sazonalidade mais estáveis (uma série estacionária do início ao fim), e a previsão feita pelo método sazonal naïve representada pela série menor (dos quatro últimos períodos).

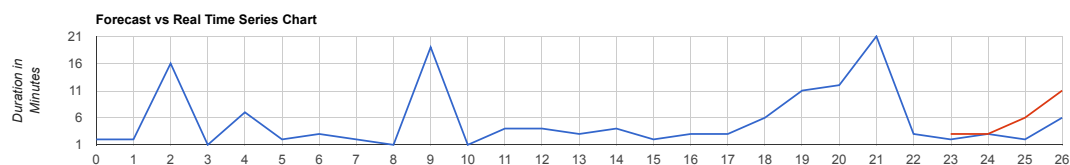
Tabela 5.7: Os 10 maiores MAEs dos testes de conectividade Wi-Fi

| <b>Algoritmo</b>      | <b>Usuário</b> | <b>MAE</b> | <b>RMSE</b> |
|-----------------------|----------------|------------|-------------|
| Holt-Winters          | A              | 10332      | 12291,08    |
| Naïve Sazonal         | A              | 10318      | 12279,32    |
| Decomposição Clássica | A              | 10280,5    | 12213,89    |
| Média móvel ponderada | A              | 10249      | 12220,29    |
| Naïve Sazonal         | D              | 4405,5     | 8243,75     |
| Naïve Sazonal         | E              | 4205,25    | 7243,28     |
| Holt-Winters          | B              | 3168,75    | 5523,54     |
| Média móvel ponderada | C              | 3141       | 3229,56     |
| Decomposição Clássica | B              | 2914       | 5392,59     |
| Média móvel ponderada | B              | 2899       | 5297,28     |

Tabela 5.8: Os 10 menores MAEs de todos os testes

| <b>Algoritmo</b>      | <b>Usuário</b> | <b>Tipo de Dado</b> | <b>MAE</b> | <b>RMSE</b> |
|-----------------------|----------------|---------------------|------------|-------------|
| Naïve Sazonal         | A              | Conectividade 3G    | 2,5        | 3,24        |
| Média móvel ponderada | A              | Conectividade 3G    | 2,75       | 3,2         |
| Decomposição Clássica | A              | Conectividade 3G    | 2,75       | 3,5         |
| Naïve Sazonal         | B              | Conectividade 3G    | 3,5        | 4,06        |
| Holt-Winters          | C              | Conectividade 3G    | 3,25       | 4,27        |
| Holt-Winters          | D              | Conectividade 3G    | 3,5        | 6,04        |
| Holt-Winters          | A              | Conectividade 3G    | 3,75       | 4,03        |
| Holt-Winters          | E              | Conectividade Wi-Fi | 3,25       | 4,27        |
| Naïve Sazonal         | F              | Conectividade Wi-Fi | 4          | 4,85        |
| Decomposição Clássica | B              | Conectividade 3G    | 5,75       | 7,19        |

Figura 5.2: Previsão com melhor MAE dos testes com dados de conectividade 3G. Em vermelho a previsão, em azul a série original. Fonte: Elaborada pelo Autor.



Uma análise dos resultados por algoritmo, apresentada na Tabela 5.9, mostra o algoritmo naïve sazonal com menor média geral de MAE e tendo um bom desempenho em conjunto com os dados de conectividade 3G. Também mostra que as séries temporais dos dados de bateria foram as mais difíceis de serem previstas e apresentaram as médias mais altas de MAE em todos os algoritmos, com a média móvel ponderada como algoritmo de resultado melhor para previsão nestas séries. O algoritmo de decomposição clássica teve o melhor resultado para dados de conectividade Wi-Fi e o algoritmo Holt-Winters teve o pior resultado de todos, para dados de bateria.

Tabela 5.9: Média MAE por algoritmo e tipo de dado

| Algoritmo             | Tipo de Dado        | MAE     | RMSE    |
|-----------------------|---------------------|---------|---------|
| Naïve Sazonal         | Bateria             | 1238,15 | 1684,24 |
| Naïve Sazonal         | Conectividade 3G    | 574,43  | 1021,14 |
| Naïve Sazonal         | Conectividade Wi-Fi | 914,73  | 1328,3  |
| Média móvel ponderada | Bateria             | 1191,24 | 1514,5  |
| Média móvel ponderada | Conectividade 3G    | 593,26  | 999,92  |
| Média móvel ponderada | Conectividade Wi-Fi | 881,83  | 1090,65 |
| Decomposição Clássica | Bateria             | 1232,63 | 1735,84 |
| Decomposição Clássica | Conectividade 3G    | 588,37  | 1032,48 |
| Decomposição Clássica | Conectividade Wi-Fi | 772,73  | 1106,68 |
| Holt-Winters          | Bateria             | 1240,91 | 1668,49 |
| Holt-Winters          | Conectividade 3G    | 592,74  | 1039,23 |
| Holt-Winters          | Conectividade Wi-Fi | 806,71  | 1118,8  |

## Capítulo 6

### Conclusões e trabalhos futuros

*Neste último capítulo dissertativo, a conclusão do estudo comparativo e as propostas de trabalhos futuros.*

Diversos trabalhos aplicam métodos de previsão em séries temporais nos dispositivos móveis, e cada vez mais a necessidade de tal aplicabilidade aumenta, como foi visto na revisão da literatura para esta monografia. Porém nestes trabalhos o desempenho computacional, energético e de forma mais detalhada a precisão destes métodos não são apresentados.

Neste contexto e para os quatro métodos de previsão em séries temporais definidos, esta monografia serve como base teórica e prática (pelos códigos apresentados nas Seções 4.3, B e C aplicáveis à plataforma Android). Além disso esta monografia atingiu seu objetivo principal ao torna-se o primeiro estudo comparativo de métodos de previsão em séries temporais nos dispositivos móveis, utilizando dados de bateria e conectividade, apresentando o desempenho computacional, energético e a precisão das previsões destes métodos (*naïve* sazonal; média móvel ponderada; por decomposição clássica utilizando o método *naïve* e o método de Holt; e suavização exponencial de Holt-Winters) para futuros trabalhos que precisem resolver problemas neste contexto.

Este estudo comparativo mostrou então que os métodos testados são computacionalmente aplicáveis em dispositivos móveis. Quanto à previsibilidade de séries temporais relacionadas aos momentos de carregamento das baterias dos aparelhos e de conectividade com as redes Wi-Fi e 3G este estudo demonstrou que em determinadas situações o comportamento do usuário torna-se um imenso dificultador. Esta dificuldade tem origem nas componentes tendência-ciclo e sazonalidade com bastante flutuação em algumas séries, ou devido à séries com longos períodos de estacionariedade interrompidos por comportamentos explosivos. Este grupo de usuários e/ou séries podem atrapalhar a realização de previsões aceitáveis para a maioria dos propósitos e soluções em dispositivos móveis, especificamente para aquelas que buscam uma otimização energética por meio do gerenciamento dos recursos de forma adaptável ao comportamento dos usuários.

Portanto, apesar da complexidade e demandas de recursos computacionais baixos, os métodos de previsão mais simples podem não ser uma alternativa viável, sendo necessário em consequência o uso de outros métodos mais complexos como os modelos autorregressivos (ARIMA e ARMA). Assim uma continuação deste

estudo comparativo pode ter como escopo a inclusão de um grupo de métodos e modelos de previsão mais complexos. Também pode ser incluso outros tipos de dados relacionados ao comportamento e uso das funcionalidades dos dispositivos móveis, como localização georreferência, navegação e uso de aplicativos, utilização das funções de telefonia, etc.

Outras possíveis melhorias em um próximo estudo comparativo são listadas: aumento da precisão da métrica de energia, aumento da precisão da métrica de uso da memória, aumento dos cenários de testes incluindo-se mais modelos de aparelhos, teste destes algoritmos apresentados nesta monografia e dos sugeridos na conclusão utilizando parâmetros de suavização otimizados por minimização dos erros.

# Apêndice A

## Termos de participação da pesquisa do monitor de dados de bateria e conectividade

Você deseja participar de uma pesquisa científica para melhoria no gasto de bateria em *smartphones*? Ao instalar este aplicativo em seu aparelho Android você concorda automaticamente em participar desta pesquisa e afirma ter lido os termos abaixo:

Esta pesquisa coleta dados de bateria (carregamento *on/off* e tipo do carregador), internet (conectividade *on/off* e tipo da rede) e o desligamento e ligamento dos aparelhos, sem alterar o funcionamento do aparelho ou do sistema Android e suas funções. Seus dados serão coletados de maneira anônima e somente enviados quando o aparelho estiver em carregamento de bateria e conectado com à internet através de rede Wi-Fi, sem utilização de planos de dados de telefonia móvel e sem o descarregamento da bateria.

Sua ajuda é imprescindível para uma análise detalhada da situação atual de gasto de bateria em aplicativos que enviam e recebem dados pela internet, o principal objetivo da pesquisa é construir uma ferramenta de desenvolvimento que permita a redução no consumo de energia das aplicações.

Ao aceitar participar da pesquisa você permite o total funcionamento do serviço de monitoração e envio anônimo de dados de bateria e conectividade no aparelho ao qual este aplicativo está instalado. Também afirma ter entendido o objetivo da pesquisa, o motivo de sua colaboração e concorda em participar.

Sua participação é voluntária. Se depois de consentir em sua participação você desistir de continuar participando, tem o direito e a liberdade de retirar seu consentimento em qualquer fase da pesquisa, seja antes ou depois da coleta dos dados, independente do motivo e sem nenhum prejuízo a sua pessoa. Assim o serviço para de funcionar e enviar dados ao nosso servidor.

Você não terá nenhuma despesa e também não receberá nenhuma remuneração. Os resultados da pesquisa serão analisados e publicados, mas sua identidade não será divulgada, sendo guardada em sigilo. Para qualquer outra informação você poderá entrar em contato com o pesquisador através do e-mail [raulccabreu@gmail.com](mailto:raulccabreu@gmail.com).

Em breve o aplicativo Empathy Monitor de Bateria (<http://is.gd/EB7YeE>) permitirá a visualização das estatísticas de bateria e conectividade do próprio usuário de forma gráfica intuitiva.

Visite a página do projeto <http://www.empathymobile.info> para maiores detalhes (documentos de pesquisa, andamento do projeto, resultados gerais, códigos fontes, etc.)

# Apêndice B

## Algoritmo da decomposição clássica com média móvel

Algoritmo B.1: Método de decomposição clássica com média móvel escrita em um algoritmo Java

```
1  int m = 7; //seasonal rate
2  //Trend-Cycle Component
3  int[] T = new int[timeSeries.length];
4  int k = (m-1)/2;
5  for (int t = k; t < timeSeries.length-horizon-k; t++) {
6      int avgSum = 0;
7      for (int j = -k; j < k; j++) {
8          avgSum += timeSeries[t-j];
9      }
10     T[t] = avgSum/m;
11 }
12 //Seasonal Component
13 //Detrended series yt - Tt
14 int[] YminusT = new int[timeSeries.length];
15 for (int t = k; t < timeSeries.length-horizon-k; t++) {
16     YminusT[t] = timeSeries[t] - T[t]; //Aditive
```

```
17 }
18 //Season indexes averaged
19 float [] seasonalIndexes = new float[m];
20 for (int mi = 0; mi < m; mi++) {
21     int sums = 0;
22     for (int t = mi; t < timeSeries.length; t+=m) {
23         seasonalIndexes[mi] += YminusT[t];
24         sums++;
25     }
26     seasonalIndexes[mi] = (int)(seasonalIndexes[mi]/sums);
27 }
28 //Normalizing to add zero
29 float factor = 0;
30 for (int mi = 0; mi < m; mi++) {
31     factor += seasonalIndexes[mi];
32 }
33 factor = factor/(float)-m;
34 for (int mi = 0; mi < m; mi++) {
35     seasonalIndexes[mi] += factor;
36 }
37 //Seasonal Indexes stringed together
38 int[] S = new int[timeSeries.length];
39 for (int mi = 0; mi < m; mi++) {
40     for (int t = mi; t < timeSeries.length; t+=m) {
41         S[t] = (int) seasonalIndexes[mi];
42     }
43 }
```

# Apêndice C

## Inicialização das componentes para o método de previsão Holt-Winters

Algoritmo C.1: Código Java de inicialização das componentes para o método de previsão Holt-Winters

```
1 //initial L
2 L[0] = Y[0];
3 //Initial T
4 double sum = 0;
5 for (int t = 0; t < m; t++) {
6     sum += (Y[m + t] - Y[t]);
7 }
8 T[0] = sum / (m * m);
9 //Initial Seasonal Indexes
10 double[] seasonalAverage = new double[seasons];
11 double[] averagedObservations = new double[Y.length];
12 for (int i = 0; i < seasons; i++) {
13     for (int j = 0; j < m; j++) {
14         seasonalAverage[i] += Y[(i * m) + j];
```

```
15     }
16     seasonalAverage[i] /= m;
17 }
18 for (int i = 0; i < seasons; i++) {
19     for (int j = 0; j < m; j++) {
20         averagedObservations[(i*m)+j]=Y[(i*m)+j]/seasonalAverage[i];
21     }
22 }
23 for (int t = 0; t < m; t++) {
24     for (int i = 0; i < seasons; i++) {
25         S[t] += averagedObservations[(i * m) + t];
26     }
27     S[t] /= seasons;
28 }
```

## Apêndice D

### Os 10 menores MAE para cada tipo de dado (Bateria, Conectividade 3G e Wi-Fi)

Tabela D.1: Os 10 menores MAEs dos testes de bateria

| <b>Algoritmo</b>      | <b>Usuário</b> | <b>MAE</b> | <b>RMSE</b> |
|-----------------------|----------------|------------|-------------|
| Decomposição Clássica | C              | 30,75      | 35,45       |
| Decomposição Clássica | B              | 54,50      | 106,51      |
| Holt-Winters          | B              | 54,50      | 106,51      |
| Decomposição Clássica | A              | 92,75      | 114,23      |
| Holt-Winters          | A              | 95,00      | 115,98      |
| Média móvel ponderada | E              | 96,75      | 108,73      |
| Média móvel ponderada | A              | 97,75      | 109,48      |
| Naïve Sazonal         | A              | 101,00     | 120,21      |
| Holt-Winters          | F              | 101,00     | 195,08      |
| Decomposição Clássica | D              | 112,50     | 146,98      |

Tabela D.2: Os 10 menores MAEs dos testes de conectividade 3G

| <b>Algoritmo</b>      | <b>Usuário</b> | <b>MAE</b> | <b>RMSE</b> |
|-----------------------|----------------|------------|-------------|
| Naïve Sazonal         | B              | 2,5        | 3,24        |
| Média móvel ponderada | B              | 2,75       | 3,2         |
| Decomposição Clássica | B              | 2,75       | 3,5         |
| Holt-Winters          | A              | 3,25       | 4,27        |
| Naïve Sazonal         | C              | 3,5        | 4,06        |
| Holt-Winters          | E              | 3,5        | 6,04        |
| Holt-Winters          | B              | 3,75       | 4,03        |
| Decomposição Clássica | C              | 5,75       | 7,19        |
| Média móvel ponderada | D              | 6,5        | 9,35        |
| Decomposição Clássica | A              | 7          | 9,46        |

Tabela D.3: Os 10 menores MAEs dos testes de conectividade Wi-Fi

| <b>Algoritmo</b>      | <b>Usuário</b> | <b>MAE</b> | <b>RMSE</b> |
|-----------------------|----------------|------------|-------------|
| Holt-Winters          | F              | 3,25       | 4,27        |
| Naïve Sazonal         | D              | 4          | 4,85        |
| Decomposição Clássica | A              | 5,25       | 6,06        |
| Naïve Sazonal         | A              | 6,25       | 6,95        |
| Naïve Sazonal         | E              | 6,25       | 7,16        |
| Decomposição Clássica | C              | 6,75       | 7,12        |
| Holt-Winters          | A              | 7,25       | 8,23        |
| Decomposição Clássica | B              | 8,25       | 11,08       |
| Média móvel ponderada | B              | 8,75       | 12,44       |
| Naïve Sazonal         | B              | 9,5        | 14,58       |

## Referências Bibliográficas

- AFONSO, M. W. *Análise dos Modelos de Previsão de Demanda Aplicados ao Controle de Estoque de uma Farmácia Hospitalar*. 2010. 63 p.
- ALLIANCE, W.-F. *Wi-Fi Alliance about organization*. 2013. Disponível em: <<http://www.wi-fi.org/about/organization>>. Acesso em: 20.11.2013.
- ALMEIDA, P. M. de. *Avaliação de ferramentas para análise de séries temporais e métodos de previsão*. 2008. 142 p.
- ANDROID, D. *Android Developer*. 2013. Disponível em: <<http://developer.android.com>>. Acesso em: 14.11.2013.
- AOSP. *Android Open Source Project*. 2013. Disponível em: <<http://source.android.com>>. Acesso em: 14.11.2013.
- ARMAND, M.; TARASCON, J.-M. Building better batteries. *Nature*, v. 451, n. 7179, p. 652–7, fev. 2008.
- AZEREDO, T. M. A. *Previsão de séries de Impostos do Município de Belo Horizonte*. 2010. 81 p.
- BARROS, M. Capítulo 5: Introdução às Séries Temporais e aos Modelos ARIMA. In: *Processos Estocásticos*. [S.l.]: Papel Virtual, 2009. cap. 5, p. 432.
- BECKER, M. H. *Modelos para Previsão em Séries Temporais: uma Aplicação para a Taxa de Desemprego na Região Metropolitana de Porto Alegre*. 2010. 46 p.
- B'FAR, R. *Mobile computing principles: designing and developing mobile applications with UML and XML*. [S.l.]: Cambridge University Press, 2005.
- BOULOS, M. N. K. et al. How smartphones are changing the face of mobile and participatory healthcare: an overview, with example from eCAALYX. *Biomedical engineering online*, BioMed Central Ltd, v. 10, n. 1, p. 24, jan. 2011.
- BOX, G.; LUCENO, A. *Statistical control by monitoring and feedback adjustment*. [S.l.]: John Wiley, New York, 1997.

CARNEIRO, C.; BARAZI, R. A. *Beginning Rails 3*. 1st. ed. Berkely, CA, USA: Apress, 2010.

CARROLL, A.; HEISER, G. An Analysis of Power Consumption in a Smartphone. In: *Proceedings of the 2010 USENIX conference on USENIX annual technical conference*. [S.l.]: USENIX Association, 2010. p. 14.

CHANDRA, N. *Implementation of Holt Winters Triple exponential smoothing and other methods*. 2012. Disponível em: <<https://github.com/nchandra/ExponentialSmoothing>>. Acesso em: 27.11.2013.

CHANDRA, P. *Code for Single, Double and Triple Exponential Forecasting*. 2012. Disponível em: <<http://pratyush-chandra.blogspot.com.br/2012/05/code-for-single-double-and-triple.html>>. Acesso em: 26.11.2013.

CHANG, T. et al. Prediction for Mobile Application Usage Patterns. In: *Nokia MDC Workshop*. [S.l.]: Nokia, 2012. p. 4.

DRESCH, R. d. F. V. *Método de análise do estado de carga restante de baterias de celular*. 2010. 58 p.

EARTHQUAKE.IT. *Il grafico sismico Haiti*. 2010. Disponível em: <<http://www.earthquake.it/blog/2010/haiti-devastante-terremoto-12-gennaio-2010/>>. Acesso em: 21.11.2013.

EHLERS, R. S. *Análise de séries temporais*. 2007.

ENGST, A.; FLEISCHMAN, G. *Kit do Iniciante em Redes Sem Fio: O guia prático sobre redes WiFi para Windows e Macintosh*. 2nd. ed. São Paulo: Pearson Makron Books, 2005.

FALAKI, H. et al. Diversity in smartphone usage. In: *Proceedings of the 8th international conference on Mobile systems, applications, and services - MobiSys '10*. [S.l.]: ACM Press, 2010. p. 179–194.

FERREIRA, D.; DEY, A. K.; KOSTAKOS, V. Understanding Human-Smartphone Concerns: A Study of Battery Life. In: *Proceedings of the 9th International Conference on Pervasive Computing*. [S.l.]: Springer, 2011. p. 19–33.

HEIKKINEN, M. V. J.; NURMINEN, J. K. Consumer Attitudes Towards Energy Consumption of Mobile Phones and Services. In: *2010 IEEE 72nd Vehicular Technology Conference - Fall*. [S.l.]: IEEE, 2010. p. 1–5.

HOLT, C. C. *Forecasting Seasonals and Trends by Exponentially Weighted Moving Averages*. [S.l.], 1957.

- HYNDMAN, R. J.; ATHANASOPOULOS, G. *Forecasting: principles and practice*. 2013. Disponível em: <<http://otexts.com/fpp/>>. Acesso em: 18.09.2013.
- IDC. *IDC Worldwide Mobile Phone Tracker*. 2013. Disponível em: <<http://www.idc.com/getdoc.jsp?containerId=prUS24257413>>. Acesso em: 16.11.2013.
- JONES, G. L.; HARRISON, P. G. Collecting battery data with Open Battery. In: *2012 Imperial College Computing Student Workshop*. [S.l.]: Schloss Dagstuhl–LZI, 2012. p. 75–80.
- KALIC, G.; BOJIC, I.; KUSEK, M. Energy Consumption in Android Phones when using Wireless Communication Technologies. In: *MIPRO, 2012 Proceedings of the 35th International Convention*. [S.l.]: IEEE, 2012. p. 754–759.
- KANG, J.-M. et al. User-Centric Prediction for Battery Lifetime of Mobile Devices. In: *Proceedings of the 11th Asia-Pacific Symposium on Network Operations and Management: Challenges for Next Generation Network Operations and Service Management*. [S.l.]: Springer-Verlag, 2008. p. 531–534.
- KHALIL, I.; SUFI, F. Mobile device assisted remote heart monitoring and Tachycardia prediction. In: *2008 International Conference on Technology and Applications in Biomedicine*. [S.l.]: IEEE, 2008. p. 484–487.
- LECHETA, R. R. *Google Android*. 2nd. ed. São Paulo: Novatec, 2010.
- MORETTIN, P. A.; TOLOI, C. M. *Séries Temporais*. 2nd. ed. São Paulo: Atual, 1987. 136 p.
- NICHOLSON, A. J.; NOBLE, B. D. BreadCrumbs: Forecasting Mobile Connectivity. In: *Proceedings of the 14th ACM international conference on Mobile computing and networking - MobiCom '08*. [S.l.]: ACM Press, 2008. p. 46.
- OLIVER, E. Diversity in smartphone energy consumption. In: *Proceedings of the 2010 ACM workshop on Wireless of the students, by the students, for the students - S3 '10*. [S.l.]: ACM Press, 2010. p. 25.
- OLIVER, E. The challenges in large-scale smartphone user studies. In: *Proceedings of the 2nd ACM International Workshop on Hot Topics in Planet-scale Measurement - HotPlanet '10*. [S.l.]: ACM Press, 2010. p. 1.
- OLIVER, E.; KESHAV, S. *Data Driven Smartphone Energy Level Prediction*. [S.l.], 2010. 1–12 p.
- PATHAK, A. *Research Statement of Abhinav Pathak*. [S.l.], 2012. 4 p.

- PERRUCCI, G. P.; FITZEK, F. H. P.; WIDMER, J. Survey on Energy Consumption Entities on the Smartphone Platform. In: *2011 IEEE 73rd Vehicular Technology Conference (VTC Spring)*. [S.l.]: IEEE, 2011. p. 1–6.
- R, D. T. C. et al. *R: A language and environment for statistical computing*. [S.l.]: R Foundation for Statistical Computing. Vienna, Austria, 2013. url: <http://www.R-project.org>, 2005.
- RAHMATI, A.; QIAN, A.; ZHONG, L. Understanding human-battery interaction on mobile phones. In: *Proceedings of the 9th international conference on Human computer interaction with mobile devices and services - MobileHCI '07*. [S.l.]: ACM Press, 2007. p. 265–272.
- RUBY, L. *Ruby Lang Org*. 2013. Disponível em: <<https://www.ruby-lang.org/en/>>. Acesso em: 14.11.2013.
- SAHA, A. K. A Developer's First Look At Android. *Linux For You Magazine*, p. 47–50, jan. 2008.
- SHEPARD, C. et al. LiveLab: Measuring Wireless Networks and Smartphone Users in the Field. *ACM SIGMETRICS Performance Evaluation Review*, v. 38, n. 3, p. 15–20, 2011.
- SOUZA, F. R. A.; SILVA, C. M. d.; AES, C. G. Segurança em Redes Wireless. *e-Xacta*, v. 2, n. 2, 2009.
- TANENBAUM, A. S. *Redes de computadores*. 4. ed. [S.l.]: Campus, 2003. 950 p.
- TARASCON, J. M.; ARMAND, M. Issues and challenges facing rechargeable lithium batteries. *Nature*, v. 414, n. 6861, p. 359–67, nov. 2001.
- TRESTIAN, I. et al. Measuring Serendipity: Connecting People, Locations and Interests in a Mobile 3G Network. In: *Proceedings of the 9th ACM SIGCOMM conference on Internet measurement conference - IMC '09*. [S.l.]: ACM Press, 2009. p. 267.
- VALLINA-RODRIGUEZ, N.; CROWCROFT, J. Energy Management Techniques in Modern Mobile Handsets. *IEEE Communications Surveys & Tutorials*, v. 15, n. 1, p. 179–198, 2013.
- VALLINA-RODRIGUEZ, N. et al. Exhausting battery statistics. In: *Proceedings of the second ACM SIGCOMM workshop on Networking, systems, and applications on mobile handhelds - MobiHeld '10*. [S.l.]: ACM Press, 2010. p. 9.
- WESSA, P. *Time Series Analysis in Free Statistics Software of ORDE*. 2013. Disponível em: <<http://www.wessa.net/tsa.wasp>>. Acesso em: 26.11.2013.

WINTERS, P. R. Forecasting Sales by Exponentially Weighted Moving Averages. *Management Science*, v. 6, n. 3, p. 324–342, 1960.

ZHANG, L. et al. Accurate online power estimation and automatic battery behavior based power model generation for smartphones. In: *Proceedings of the eighth IEEE/ACM/IFIP international conference on Hardware/software codesign and system synthesis - CODES/ISSS '10*. [S.l.]: ACM Press, 2010. p. 105.