UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO

Daniel José de Carvalho

Métodos de previsão de consumo de energia elétrica residencial em grande volume de dados

Recife

2019

Daniel José de Carvalho

Métodos de previsão de consumo de energia elétrica residencial em grande volume de dados

Monografia apresentada ao Curso de Bacharelado em Sistemas de Informação da Universidade Federal Rural de Pernambuco, como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação.

Universidade Federal Rural de Pernambuco – UFRPE

Departamento de Estatística e Informática

Curso de Bacharelado em Sistemas de Informação

Orientador: Victor Wanderley Costa de Medeiros

Coorientador: Glauco Estácio Gonçalves

Recife

2019

À minha família, amigos e todo ser do universo que, direta ou indiretamente, contribuíram para minha formação...

Agradecimentos

Agradeço à minha vida, que o universo, de certa forma, conspirou para que existisse e durasse até esse momento.

Agradeço a minha noiva, futura esposa, Rafaela, que sempre acreditou em mim e me permitiu cursar mais uma graduação, mesmo que isso significasse prolongar ainda mais nossos planos de casamento, mas que sabia que eu estava fazendo o que era melhor para mim. Muito obrigado pela compreensão, e isso é uma das coisas que me fazem te amar.

Agradeço a minha família, minha mãe Jeane, que me colocou no mundo e sempre me ajudou da forma que pôde. Minhas tias Girlaide, Girleide, Girlene, Marlene, que também buscaram sempre me ajudar no que fosse preciso. Minha prima Gabriela, e seu noivo Matheus, que também me ajudaram bastante.

Agradeço aos amigos que fiz no curso, inicialmente Filipe, que me ajudou sempre que podia, seja no deslocamento até a rural ou ajudando nos projetos do curso, e com isso construindo uma amizade forte. Em seguida Nichene e Demis, duas pessoas muito inteligentes que tive o prazer de conhecer e partilhar conhecimentos e bons momentos no curso e na iniciação científica. O bando de Jonathan, Udney, João Vitor, Vinícius e Romero, pessoas que apareceram sem eu esperar, criando fortes laços de amizade, dividindo projetos e conhecimentos durantes nossas noites de reunião, além de discussões de assuntos mais mundanos. Já na metade do curso, apareceu mais um, Airton, que também demonstrou ser um grande amigo para qualquer hora. Demais pessoas e amigos, sintam-se mencionados.

Aos professores Victor Medeiros e Glauco Gonçalves, dois mestres fantásticos que em nenhum momento hesitaram de me fornecer conhecimento e conselhos para minhas dúvidas e questionamentos. Boa parte do meu conhecimento devo a vocês dois. Aos professores do BSI como um todo, um time de mentes incríveis e conhecimentos diversos, que transmitiram seus conhecimentos e ajudaram na constituição de minhas competências profissionais e acadêmicas.

Ao pessoal do LINCS - CETENE, especialmente Pyetro e Vanessa, que compartilhou seu conhecimento comigo, e que forneceu uma chance para que eu pudesse aprender assuntos e técnicas que hoje uso no meu cotidiano profissional.

“Não há nada mais difícil de executar, mais perigoso para conduzir, ou mais incerto em seu sucesso, do que assumir a liderança na introdução de uma nova ordem de coisas. O reformador tem inimigos em todos aqueles que lucram com a velha ordem, e apenas defensores tépidos em todos aqueles que se beneficiariam da nova ordem. Essa tepidez surge em parte do medo de seus adversários e em parte da incredulidade da humanidade, que não acredita verdadeiramente em nada novo até que eles tenham tido uma experiência real.”

(Nicolau Maquiavel)

“Piratas são maus? Os marinheiros são justos? Esses termos sempre mudaram ao longo do curso da história! Crianças que nunca viram a paz e crianças que nunca viram a guerra têm valores diferentes! Aqueles que estão no topo determinam o que está errado e o que está certo! Este lugar é muito neutro! Justiça irá prevalecer, você diz? Mas é claro que vai! Quem vencer esta guerra se torna a justiça!”

(Donquixote Doflamingo - One Piece)

“Quando vocês acham que as pessoas morrem? Quando elas levam um tiro de pistola bem no coração? Não! Quando são vencidas por uma doença incurável? Não! Quando bebem uma sopa de cogumelo venenoso? Não! Elas morrem... Quando são esquecidas.”

(Dr. Hiluluk - One Piece)

“Jogue somente para ganhar, e não para perder. A vida é assim, cheia de dificuldades. Fale o que quiser.”

(Ednaldo Pereira)

Resumo

A energia elétrica é uma das principais fontes de energia utilizadas pela humanidade. A crescente preocupação com a preservação do meio-ambiente estimulou a busca por fontes de energia renováveis capazes de reduzir os impactos à natureza. O crescimento populacional e o uso cada vez mais comum de dispositivos eletrônicos, na quase totalidade das atividades cotidianas, demandam o uso mais eficiente da energia elétrica. Diante destes desafios é essencial a realização de um planejamento para dimensionar a estrutura de geração e transmissão de energia elétrica. Uma das ferramentas capazes de auxiliar neste dimensionamento é a previsão de demanda. Outro grande desafio nesta área está na realização destas previsões em cenários de grandes dados (Big Data).

Este trabalho tem como objetivo principal avaliar o desempenho de dois métodos de previsão, ARIMA e Holt-Winters, utilizando séries temporais aplicados a um grande volume de dados. A base de dados utilizada foi fornecida no evento DEBS 2014 Grand Challenge, a qual contém dados de consumo de energia elétrica, de um grande número de residências, durante o período de um mês. Para a aplicação dos métodos de previsão, foram utilizadas bibliotecas na linguagem R. Para processar os dados, utilizou-se o *framework* Apache Spark em conjunto com a linguagem R, para paralelizar o processamento da leitura dos dados e a filtragem dos parâmetros desejados. Os dados tratados foram convertidos em séries temporais com valores de consumo horários, durante todo o mês compreendido pela base de dados original. Foram executadas previsões para a região das residências como um todo e para cada residência individualmente. Os resultados mostraram uma vantagem do ARIMA frente ao Holt-Winters no cenário utilizado, utilizando a métrica RMSE como base comparativa de desempenho. Contudo, baseado em experimentos similares encontrados na literatura, resguardando as devidas proporções, ambos os valores de RMSE estão dentro de uma faixa aceitável.

Palavras-chave: Big Data, consumo de energia elétrica, séries temporais, ARIMA, Holt-Winters.

Abstract

Electricity is one of the primary sources of energy used by humanity. Growing concern for the preservation of the environment has stimulated the search for renewable energy sources capable of reducing impacts on nature. Population growth and the increasingly frequent use of electronic devices in almost all daily activities demand the most efficient use of electricity. Due to these challenges, it is essential to carry out planning to dimension the structure of generation and transmission of electric energy. One of the tools capable of assisting in this sizing is the demand forecasting. Another major challenge in this area lies in the realization of these forecasts in large data scenarios (Big Data).

This work aims to evaluate the performance of two prediction methods, ARIMA and Holt-Winters, using temporal series applied to a large volume of data. The database was provided by the DEBS 2014 Grand Challenge event, which contains electricity consumption data for a large number of households for one month. For the application of the prediction methods, we used libraries in the R language. In order to process data, the Apache Spark framework was used in conjunction with the R language to parallelize the data reading processing and filtering parameters. The treated data were converted into time series with hourly consumption values, throughout the month comprised by the original database. Predictions were made for the region of the households as a whole and each residence individually. The results showed an advantage of ARIMA versus Holt-Winters in the scenario used, using the RMSE metric as a comparative basis of performance. However, based on similar experiments found in the literature, with due proportions, both RMSE values are within an acceptable range.

Keywords: Big Data, electricity consumption, time series, ARIMA, Holt-Winters.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

ACF	*Autocorrelation Function*
AIC	*Akaike Information Criterion*
AICc	*Corrected Akaike Information Criterion*
ARIMA	*Auto Regressive Integrated Moving Average*
ARMA	*Auto Regressive Moving Average*
BIC	*Bayesian Information Criterion*
CI	Critério de Informação
DEBS	*Distributed and Event-Based Systems*
I/O	*Input/Output*
IoT	*Internet of Things*
GB	Gigabyte
GWh	Gigawatt-hora
KPSS	Kwiatkowski–Phillips–Schmidt–Shin
kWh	Quilowatt-hora
MAE	*Mean absolute error*
MAPE	*Mean absolute percentage error*
MMS	Média Móvel Simples
RMSE	*Root Mean Squared Error*
RTC	*Real-time Clock*
TWh	Terawatt-hora
Wh	Watt-hora

Sumário

1 Introdução

A sociedade do século XXI usufrui de um modo de vida bastante distinto da sociedade de 50 anos atrás. Em 1979, apesar de já terem sido criados[1], ainda não eram comercializados aparelhos celulares. A Internet ainda não era como atualmente, dado que ainda existia a ARPANET, e a base para a *World Wide Web* (WWW), a ferramenta mais conhecida e utilizada da Internet hoje, ainda viria a ser elaborada nos trabalhos de Timothy John Berners-Lee em 1980[2]. Os automóveis eram bem mais arcaicos comparados com os atuais, no entanto, não mudaram tanto com relação a utilização majoritária de combustíveis fósseis. O uso de fontes de energia renováveis para automóveis, como os carros elétricos, ainda não era amplamente discutido como atualmente.

Com o passar dos anos, o uso da tecnologia por parte da população foi crescendo continuamente, seja no ambiente doméstico quanto no corporativo. Segundo o IBGE (Instituto Brasileiro de Geografia e Estatística) (IBGE, 1990), em 1952 o consumo de energia elétrica era de 8.513 GWh, e em 1987 apresentava um consumo de 192.127 GWh, significando um aumento na ordem de 183.614 unidades de medida. Segundo dados da EPE (Empresa de Pesquisa Energética) (EMPRESA DE PESQUISA ENERGÉTICA – EPE, 2018), entre 1995 e 2018, o consumo geral de energia elétrica no Brasil aumentou 229.168 unidades de medida, partindo de 243.074 GWh em 1995 a 472.242 GWh em 2018. Com isso, percebe-se que em um menor intervalo de tempo, o aumento no consumo foi maior.

Utilizando o período compreendido entre 1995 e 2018, e decompondo o consumo em residencial, industrial e comercial, verificou-se que o consumo de energia elétrica comercial teve um aumento na ordem de 175,17%, o consumo residencial aumentou em 113,95% e o industrial aumentou em 51,89%. Os dados completos de consumo geral de energia elétrica no Brasil estão demonstrados na Figura 1. Esses dados sustentam a mudança no modo de vida da sociedade com o passar dos anos, dado que o aumento no consumo residencial foi, proporcionalmente, maior que o dobro do aumento verificado no consumo industrial. Ou seja, as pessoas passaram a utilizar mais dispositivos eletrônicos, e com isso, consumir mais energia para alimentar estes dispositivos. Segundo (GONTIJO et al., 2017), o consumo de energia elétrica constitui um forte indicador de desenvolvimento econômico e de qualidade de vida, pois a demanda energética expressa o ritmo de atividade industrial e comercial, e consequen-

[1] Invenção do aparelho celular- https://www.terra.com.br/noticias/tecnologia/celular/voce-sabia-quem-inventou-o-telefone-celular,9ae917e79a3207d7e3ced75f7a4f1f05tebbkqzg.html

[2] História da Internet - http://dicas.ufpa.br/net1/int-h198.htm

temente, a capacidade de consumo das pessoas.

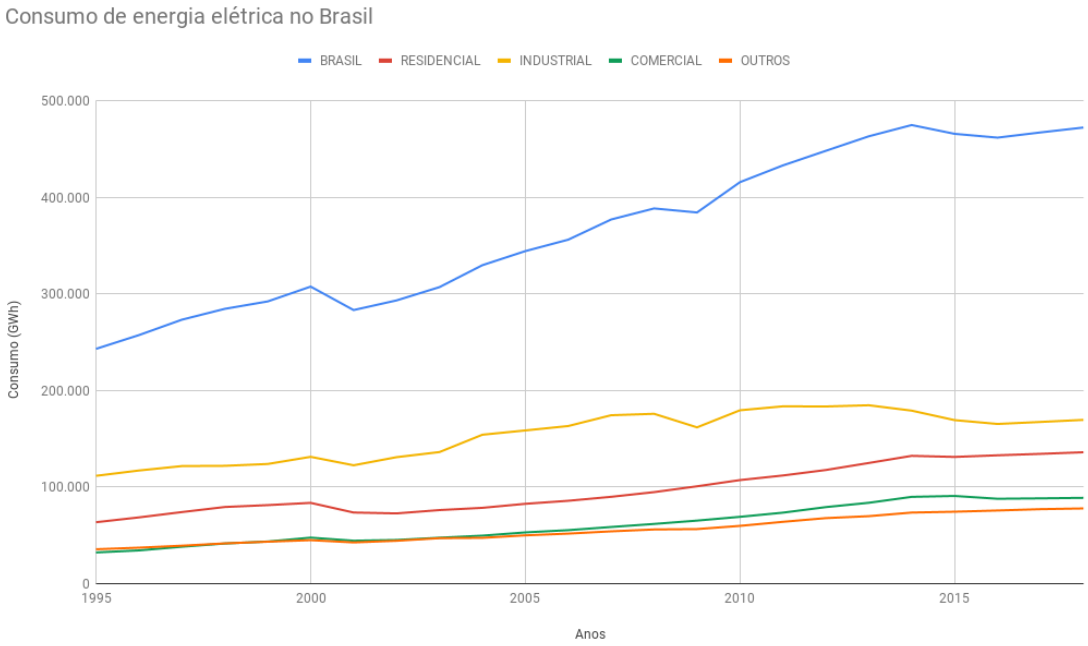


Figura 1 – Histórico de consumo de energia elétrica no Brasil entre 1995 e 2018
fonte: (EMPRESA DE PESQUISA ENERGÉTICA – EPE, 2018)

De acordo com (FIGURES, 2018), o consumo de energia no mundo, entre 1995 e 2016, aumentou em 88,27%, totalizando o consumo de 24.973 TWh. Em 2016, o Brasil consumiu 461,78 TWh durante o ano todo, representando apenas 1,85% do consumo mundial. Na Figura 2, é verificada a contribuição de cada tipo de fonte de geração de energia elétrica para o montante consumido. A maior quota é representada pela geração de energia proveniente de fontes sólidas (38,4%), conseguida a partir da queima de determinados materiais, liberando calor no processo, e muito utilizada em usinas termoelétricas. Esse tipo de usina é bastante nociva ao meio ambiente. Há uma tendência de transição no tipo de fonte energética adotada no mundo. Entre os anos de 1995 e 2016, a geração de energia por meio de fontes sólidas aumentou 92%, enquanto a mediante fontes de energia renováveis, menos nocivas à natureza, aumentou 125% (FIGURES, 2018).

Segundo (AHMAD et al., 2014), a demanda por moradia e desenvolvimento das nações irá aumentar devido ao aumento populacional, consequentemente, aumentando o consumo de energia elétrica. Isso significa um maior volume de dados a ser coletado e processado, adentrando na área de Big Data. De acordo com (SHORO; SOOMRO, 2015), quanto mais dados a organização tiver, mais precisos serão os resultados das análises, levando a tomadas de decisões mais assertivas e reduzindo riscos. Com isso, saber lidar com um grande volume de dados é uma característica essencial para os sistemas contemporâneos.

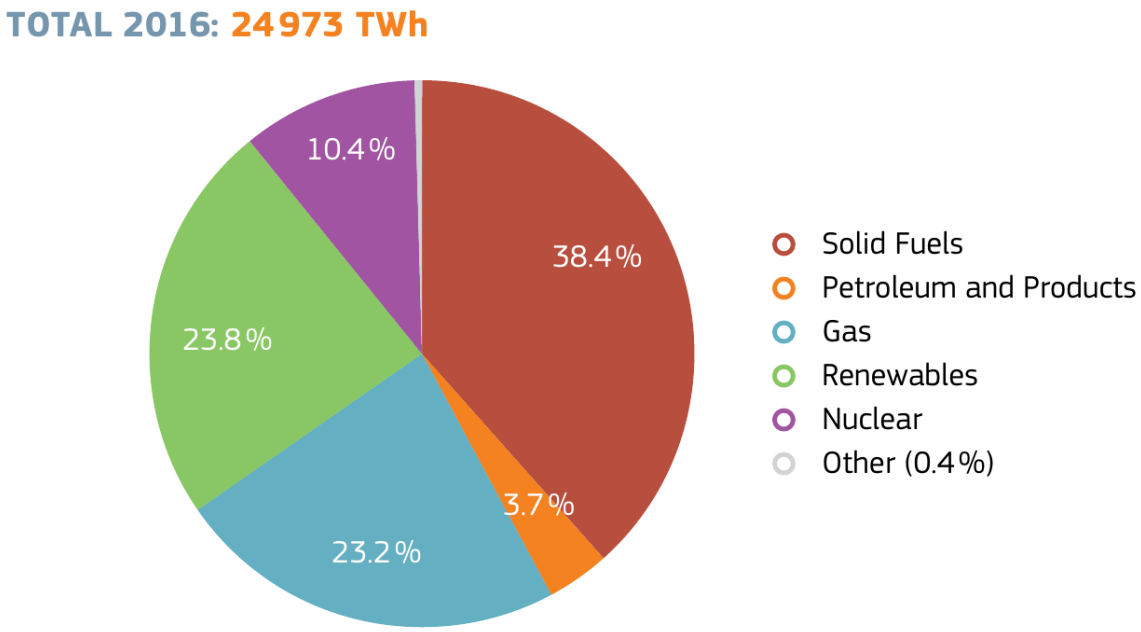


Figura 2 – Consumo de energia elétrica no Mundo em 2016 com as fontes de geração de energia detalhadas
fonte: (FIGURES, 2018)

1.1 Justificativa e Motivação

O aumento no consumo de energia elétrica traz à tona algumas necessidades por parte das empresas fornecedoras de energia, dentre elas, realizar um eficaz dimensionamento da rede elétrica a fim de suportar a crescente demanda da população. A rede elétrica que fornece energia para as residências, oficialmente, é dimensionada pela empresa fornecedora de energia elétrica da localidade. Para isso, é realizado um levantamento da carga a ser suportada na região, considerando fatores como a soma das potências nominais dos equipamentos ligados à rede e a demanda.

Consumidores residenciais apresentam um comportamento mais variado frente à outros tipos de consumidores, como dito em (AHMAD et al., 2014), o que dificulta a otimização do dimensionamento da rede, devido a liberdade que tais consumidores possuem de adquirir dispositivos e ligá-los a rede elétrica a qualquer momento.

Além disso, com as residências dispondo de cada vez mais dispositivos eletrônicos, as cidades incorporando serviços digitais e o advento de IoT (*Internet of Things* ou Internet das Coisas), o consumo tende a aumentar, gerando um grande volume de dados. Isso demonstra a necessidade de lidar com um cenário desse tipo de forma satisfatória.

Para tentar minimizar o erro no dimensionamento, uma das ferramentas utilizadas são **Modelos de Previsão de Demanda**, onde, fazendo uso de conhecimentos estatísticos, conseguem obter valores futuros que apresentam uma alta probabilidade

de se concretizar. Conforme (KAYTEZ et al., 2015), previsão de demanda de energia a longo prazo é a base para planejar investimentos e para o desenvolvimento dos países. Além disso, paralelizando o processamento, possibilita tratar um grande volume de dados em tempo hábil, se adequando ao cenário de Big Data.

1.2 Objetivos

Este trabalho tem como objetivo principal realizar um estudo comparativo de métodos de previsão em séries temporais aplicados a um grande conjunto de dados históricos de consumo energético residencial. Para o estudo, serão utilizados os métodos de previsão Holt-Winters e ARIMA. De forma específica, pretende-se atingir os seguintes objetivos:

- Desenvolver um protótipo funcional do ReCast, nome dado ao sistema, com os métodos de previsão de demanda na linguagem R utilizando o Apache Spark para leitura paralela dos dados de entrada;
- Avaliar quantitativamente a precisão dos métodos de previsão de séries temporais executados;

1.3 Organização do trabalho

Este trabalho está dividido em seis capítulos, em que o presente capítulo é introdutório. No capítulo 2, são abordados trabalhos existentes na literatura similares e/ou relacionados ao proposto por esse projeto. O capítulo 3 demonstra a fundamentação teórica dos assuntos utilizados para o desenvolvimento deste trabalho, enquanto no capítulo 4 é detalhada a metodologia aplicada para a execução dos experimentos, demonstrando cada etapa. No capítulo 5 são realizadas as análises a respeito dos resultados obtidos, e no capítulo 6 são feitas as considerações finais.

2 Fundamentação Teórica

2.1 Séries Temporais

Estimar valores futuros necessita primordialmente de valores do passado. Sem ter valores concretos para basear uma previsão, a confiabilidade desta será deveras insignificante frente a previsões fortemente baseadas em dados históricos. De forma geral, no ambiente profissional e/ou acadêmico, esses dados são conhecidos como **Séries Temporais**. Segundo (DEB et al., 2017), séries temporais são uma sequência ordenada de valores de acordo com um determinado intervalo de tempo.

As séries temporais podem ser decompostas em **Tendência (*Trend*)**, **Sazonalidade (*Seasonal*)**, **Ciclo (*Cycle*)** e **Erro (*Error*)**. Na Figura 3, pode-se visualizar a decomposição de uma série temporal de dados de consumo mensal de energia elétrica no Sudeste brasileiro[1], onde é possível identificar, além do valor Observado (*Observed*), a Tendência, Sazonalidade e o Erro (*Random*).

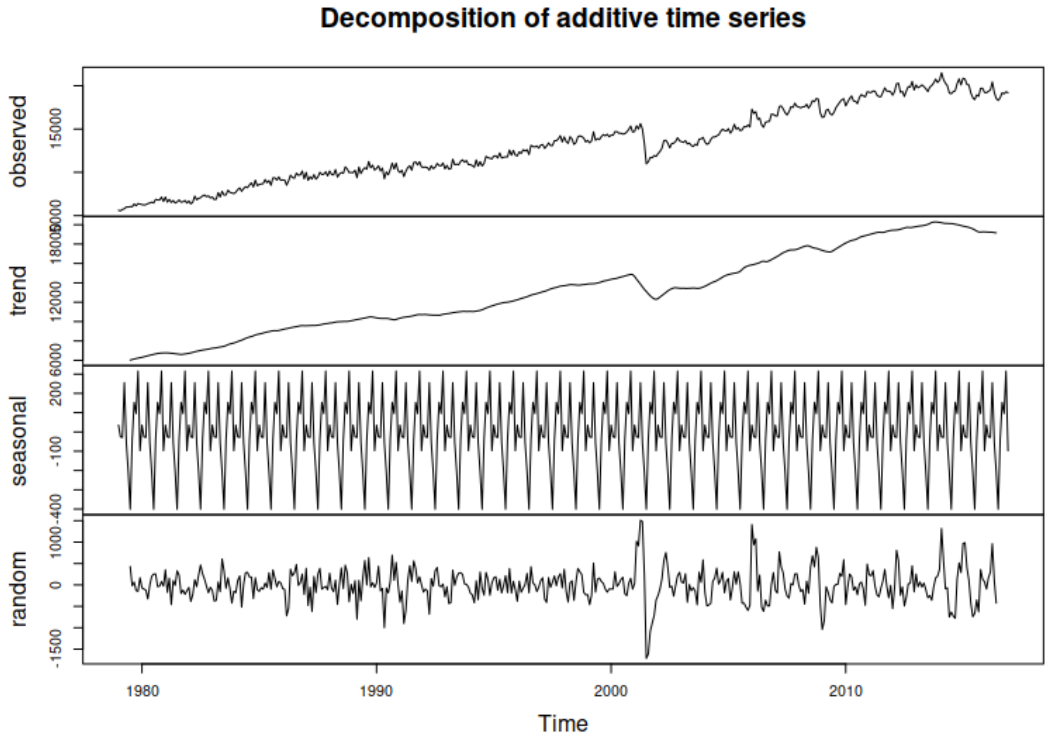


Figura 3 – Série temporal decomposta.
fonte: próprio autor

De acordo com (FERREIRA et al., 2018) e (GONÇALVES, 2018), Tendência é

[1] Consumo de energia elétrica no Sudeste do Brasil - http://bit.ly/consumoSudeste

um componente existente em séries temporais quando os valores demonstram uma propensão em seguir uma direção, seja ela crescente ou decrescente, e não obrigatoriamente há de ser linear. Para identificar a Tendência em uma dada série temporal, podem ser realizados diferentes tipos de técnicas, onde a mais comum é o ajuste de Regressão Linear Simples, evidenciando a inclinação da reta de tendência.

A Sazonalidade consiste em um padrão repetitivo em um dado intervalo de tempo menor que 1 ano, seja ele horário, diário, semanal ou qualquer outro. Um exemplo de um padrão dessa magnitude pode ser visualizado em determinadas estações do ano, segundo (COCEL, 2019), onde o consumo de energia elétrica aumenta em comparação com outras épocas do ano, devido ao uso intenso de alguns equipamentos para prover mais conforto aos respectivos usuários, como ar-condicionados e chuveiros elétricos.

O Ciclo, por sua vez, é um padrão recorrente identificado, tal qual a Sazonalidade, contudo em um período maior que o de 1 ano, e não apresenta um período fixo para sua ocorrência. Geralmente, Ciclos são influenciados por questões econômicas, representando o chamado “Ciclo de negócios”. Por exemplo, uma série histórica de vendas imóveis pode transparecer um ciclo de vendas onde é possível visualizar durante o período o aumento, estagnação e declínio de vendas. Esse ciclo, como dito anteriormente, não precisa respeitar um intervalo fixo de tempo, podendo apresentar 2, 6, 10 ou mais anos.

Na decomposição é possível também identificar um “Erro”, também chamado de “Ruído”, “*Random*” ou “*Noise*”, que consiste em um componente não-sistemático não pertencente a nenhum dos outros padrões dentro dos dados analisados.

Outra característica bastante relevante das séries temporais é a de **Autocorrelação**, que consiste em uma medida da correlação entre os valores de uma série temporal separados por um valor em unidades de tempo. Diferente de outros modelos estatísticos, onde a ordem dos registros é irrelevante para se obter os resultados esperados, as séries temporais necessitam saber a ordem dos eventos. Dados de venda de um produto no mês anterior pode influenciar nas vendas do mês presente, por exemplo, e isso é muito importante para se realizar uma previsão de demanda mais precisa. A autocorrelação pode estar em diferentes ordens, a depender de quantos níveis estão distantes as variáveis correlacionadas, e para identificar, é utilizada a ACF (Função de Autocorrelação ou *Autocorrelation function*).

De acordo com (MINITAB, 2019), a ACF é uma medida de correlação entre os valores observados de uma série temporal, separados por unidades de tempo. Analisando seus valores, e considerando um determinado nível de confiança, é possível afirmar se os valores são significativos ou não, apontando a intensidade da relação dos valores com os valores passados. Na Figura 4, é possível visualizar um exemplo de

gráfico representando a ACF de uma série temporal qualquer, onde é identificado que é uma autocorrelação de primeira ordem, devido ao ponto no lag 1 ser significativo.

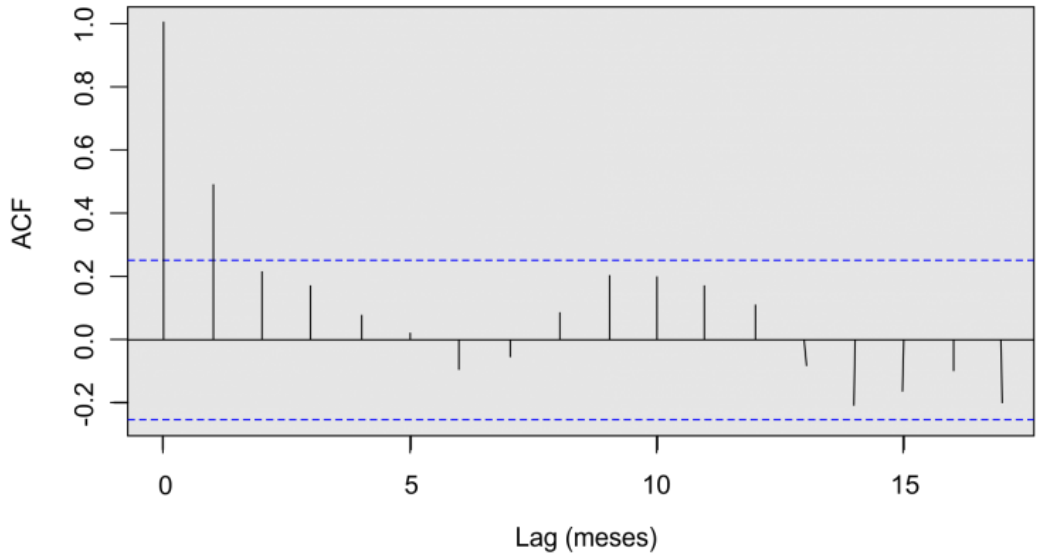


Figura 4 – Exemplo de um gráfico de ACF de uma série temporal
fonte: (GONÇALVES, 2018)

Em conjunto com a ACF, é utilizada também a PACF (Função de Autocorrelação Parcial ou *Partial Autocorrelation Function*). A PACF é uma medida da correlação entre os valores de uma série temporal separados por unidades de tempo, após a remoção dos efeitos de correlações devidos a valores passados.

Estacionariedade também é uma peculiaridade das séries temporais. Uma série temporal é considerada "estacionária" caso seus valores estejam distribuídos ao redor de uma média fixa no decorrer do tempo, demonstrando estabilidade. De antemão, é possível afirmar que a "tendência" em uma série temporal contribui para torná-la não-estacionária, pois contradiz o requisito essencial, dado que, a depender da inclinação da tendência, a média dos valores irá se alterar com o tempo. A maior parte dos modelos estatísticos de previsão assumem que a série analisada é estacionária, e caso não seja, será necessário transformar os dados. A transformação mais utilizada realiza diferenças sucessivas da série temporal em questão, segundo a expressão:

$$y_t^{'} = y_t - y_{t-1}$$

onde:

- $y_t^{'}$ é a diferença
- y_t e y_{t-1} são observações consecutivas

Para verificar se a série é estacionária, são realizados testes, e um deles é o teste **KPSS (Kwiatkowski–Phillips–Schmidt–Shin)**. Segundo (STEPHANIE, 2016), o KPSS se baseia em regressão linear, decompondo a série temporal em tendência determinística, *random walk* e erro. Utilizando o Método dos Mínimos Quadrados, tenta encontrar a equação de regressão. Se os dados forem estacionários, a série irá ser distribuída ao redor de uma média constante. A equação de regressão é a seguinte:

$$x_t = r_t + \beta_t + \varepsilon_t$$

onde:

- x_t é a série
- r_t é o *random walk*
- β_t é a tendência determinística
- ε_t é o erro

2.2 Métodos de previsão estatística

De acordo com (HYNDMAN; ATHANASOPOULOS, 2018), "Previsão" ou "*Forecasting*" consiste em tentar prever valores futuros utilizando todos os recursos e informações existentes e de conhecimento de eventos futuros que possam influenciar nos resultados. As técnicas de previsão são bastante utilizadas por organizações para basear as tomadas de decisões, e essas antevisões podem ser de Curto, Médio ou Longo prazo. Com relação aos métodos utilizados, há dois tipos de previsão: **quantitativa** e **qualitativa**.

O tipo qualitativo, de acordo com (BRAGG, 2018), faz uso do conhecimento e experiência de profissionais da respectiva área que se está analisando, em vez de utilizar análises numéricas. Apesar da suposta falta de base de dados mais consistentes, esse tipo é bastante útil quando não há dados históricos suficientes para se realizar as análises, ou quando se percebe que os valores futuros irão depender cada vez menos dos valores passados. Contudo, esse tipo de previsão pode ser tendencioso, pois é comum os resultados serem influenciados pela visão pré-concebida dos profissionais que realizarão as análises.

Já o tipo quantitativo, segundo (PLANNING, 2019), recorre à dados do passado para realizar as previsões, empregando métodos estatísticos e matemáticos à esses dados para obter os resultados esperados. Seu uso é válido em situações, contrastando com as previsões qualitativas, onde há uma quantidade de dados históricos

suficiente para realizar as análises, e quando há o entendimento de que os valores futuros dependem fortemente do passado. Por isso, as séries temporais são recursos essenciais para esse tipo de previsão. No entanto, previsões quantitativas podem utilizar também dados de corte transversal (*cross-sectional data*), abordado em (ALLASSIGNMENTHELP, 2018), em que o momento que ocorreu cada registro não tem importância. Nesse trabalho, serão utilizados métodos quantitativos utilizando séries temporais.

Os valores obtidos das análises representam um valor intermediário de um intervalo de possíveis valores que possam vir a ocorrer, de acordo com o respectivo nível de confiança. Por exemplo, com um nível de confiança de 95%, a probabilidade do valor real estar contido no intervalo resultante da análise é de 95%. Na Figura 5 é demonstrado um gráfico representando valores previstos de 10 meses de consumo de energia elétrica na região Sudeste do Brasil, utilizando uma série temporal de registros mensais durante 37 anos, com uma probabilidade de 95%.

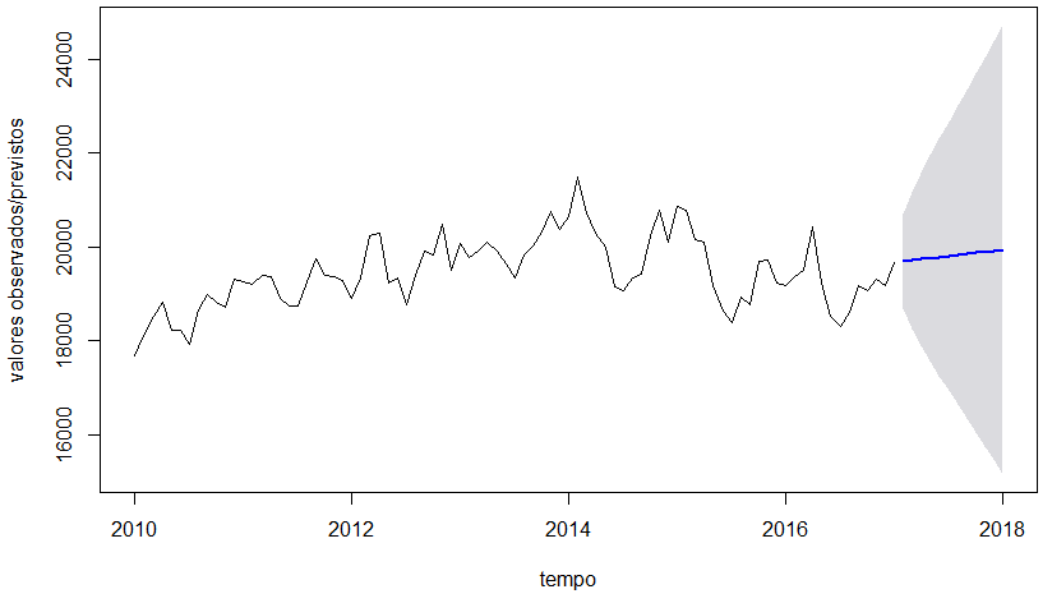


Figura 5 – Exemplo de uma previsão de consumo de energia com nível de confiança de 95%

fonte: próprio autor

2.2.1 ARIMA

De acordo com (JIANG et al., 2018) e (VEIGA; TORTATO; SILVA, 2014), o método de previsão ARIMA (*Auto Regressive Integrated Moving Average*) é um dos mais estudados e aplicados no campo de análise de previsões, devido às suas propriedades e por suportar fortemente evidências empíricas, aplicado a série temporais univariadas.

Ele é uma generalização de um modelo ARMA (*Autoregressive Moving Average*), e é aplicado em situações onde a série temporal a ser analisada apresenta traços de não-estacionariedade, havendo a necessidade de se realizar diferenciações para torná-la estacionária. O ARIMA apresenta algumas características essenciais, como:

- **Autorregressivo**
  - Valores passados são considerados para calcular os valores futuros, onde são atribuídos pesos de acordo com o tempo do dado.
- **Médias móveis**
  - Essa característica elimina o não determinismo ou movimentos aleatórios da série temporal.
- **Integrado**
  - Constitui a propriedade do ARIMA em reduzir a sazonalidade de uma série temporal, no caso de ser utilizada uma série temporal com tendência, e consequentemente, ser não estacionária.

O ARIMA existe no modo **SAZONAL** e **NÃO SAZONAL**. O modelo **NÃO SAZONAL** é o mais comum, pois ele desconsidera a sazonalidade da série. Normalmente, se representa modelos ARIMA como **ARIMA(p,d,q)**, em que "p" é a ordem do modelo auto-regressivo, "d" é o número de diferenciações feitas nos dados e "q" é a ordem do modelo de médias móveis.

A fórmula geral de um ARIMA **NÃO SAZONAL**, segundo (FLANDOLI, 2011), pode ser expressa da seguinte forma:

$$(1 - \sum_{i=1}^{p} \phi_i L^i)(1 - L)^d X_t = (1 + \sum_{k=1}^{q} \theta_k L^k)\varepsilon_t$$

onde:

- **p, d e q** são os parâmetros do ARIMA;
- **L** é o operador defasagem;
- ϕ_i são os termos do polinômio ligado ao operador AR;
- θ_i são os termos do polinômio ligado ao operador MA;
- ε_t é o ruído branco (sinal discreto cujas amostras são vistas como uma sequência de variáveis aleatórias não auto-correlacionadas com média zero e variância finita).

O ARIMA **SAZONAL**, também conhecido como **SARIMA**, é uma extensão do ARIMA, onde é suportado sazonalidade nos dados analisados. Ele possui mais 3 parâmetros (**P, D** e **Q**), representando a parte sazonal do modelo, que são referentes a, respectivamente, **Ordem autoregressiva da sazonalidade**, **Ordem de diferenciação da sazonalidade** e **Ordem de média móvel da sazonalidade**. Com isso, costuma ser representado como **ARIMA(p,d,q)[P,D,Q]**. Segundo (WALTER et al., 2013), a fórmula geral do SARIMA é:

$$\phi_p(B)\Phi_p(B^s)\Delta^d\Delta_S^D Z_t = \theta_q(B)\Theta_Q(B^s)a_t$$

onde:

- **p, d e q** são os parâmetros do ARIMA;
- **P, D, Q** são os parâmetros adicionais do SARIMA;
- **B** é o operador de defasagem;
- $\Delta^d\Delta_S^D Z_t = (1 - B^S)^D Z_t$
- $\Phi_p(B^s)$ é o operador sazonal AR(p);
- $\Theta_Q(B^s)$ é o operador sazonal MA(q);
- $\phi_1 ... \phi_p$ são parâmetros do modelo sazonal AR(p);
- $\Theta_1 ... \Theta_Q$ são parâmetros do modelo sazonal MA(q);

Selecionar o melhor modelo ARIMA para uma determinada série temporal não é algo simples, e dificilmente haverá um modelo perfeito. De acordo com (EMILIANO et al., 2010), a escolha entre os modelos possíveis é na verdade uma sugestão da melhor configuração para a série em questão, e uma das formas de fazer isso é utilizando valores de referência, denominados **Critérios de Informação (CI)**. Os CI são valores de ajuste de modelos baseados na verossimilhança, considerando uma penalidade de acordo com a complexidade. Alguns dos mais utilizados são o **AIC** (*Akaike Information Criterion*), **AICc** (*Corrected Akaike Information Criterion*) e **BIC** (*Bayesian Information Criterion*). As fórmulas destes CI são:

$$AIC = 2k - 2ln(\hat{L})$$
$$AICc = AIC + \frac{2k^2 + 2k}{n - k - 1}$$
$$BIC = ln(m)k - 2ln(\hat{L})$$

onde:

- k é o número de parâmetros estimados no modelo
- L é o valor máximo da função de verossimilhança para o modelo
- n é o tamanho da amostra
- m é o número de observações

2.2.2 Holt-Winters

O **Holt-Winters** é um modelo do tipo **Suavização Exponencial**, também conhecido como **Suavização Exponencial Tripla**. Proposto por Peter Winters, baseando-se no modelo desenvolvido por seu professor Charles Holt, o Holt-Winters leva em consideração a sazonalidade e/ou a tendência da série temporal(SMARTEN, 2018). Para cenários como consumo de energia elétrica, onde esses dois fatores são predominantes, esse modelo se mostra como uma boa opção para calcular valores futuros.

A **Suavização Exponencial** é uma técnica que, diferente da MMS (Média Móvel Simples) a qual atribui pesos iguais para dados passados, os pesos atribuídos para os dados históricos são decrescentes exponencialmente à medida que os dados vão se tornando cada vez mais antigos. Logo, os dados mais recentes possuem uma influência maior nos resultados das previsões que os mais velhos.

O modelo Holt-Winters pode ser utilizado com os métodos **Aditivo** e **Multiplicativo**. De acordo com (FERONI; ANDREÃO, 2017) e (SILVA; SANTOS; COSTA, 2016), o modelo aditivo pressupõe uma variação constante da sazonalidade no decorrer do tempo. Já o modelo multiplicativo considera que a amplitude da sazonalidade varia com o tempo, e que é muito provável essa variação ocorrer de forma crescente.

São utilizadas três constantes, chamados de **parâmetros de suavização**, α, β e γ, que são, respectivamente, os parâmetros de valores passados, tendência e sazonalidade. Esses valores são estimados a fim de minimizar o erro, e para fins práticos, essa estimação é delegada para um software realizar de forma automática. Segundo (DANTAS; OLIVEIRA; REPOLHO, 2017), as fórmulas desses modelos são:

- **Modelo Aditivo**

$$l_t = \alpha(y_t - s_{t-m}) + (1-\alpha)(l_{t-1} + b_{t-1})$$

$$b_t = \beta(l_t - l_{t-1}) + (1-\beta)b_{t-1}$$

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

$$\hat{y}_{t+h} = l_t + hb_t + s_{t-m+h}$$

- **Modelo Multiplicativo**

$$l_t = \alpha(\frac{y_t}{s_{t-m}}) + (1-\alpha)(l_{t-1} + b_{t-1})$$
$$b_t = \beta(l_t - l_{t-1}) + (1-\beta)b_{t-1}$$
$$s_t = \gamma(\frac{y_t}{l_{t-1} - b_{t-1}}) + (1-\gamma)s_{t-m}$$
$$\hat{y}_{t+h} = (l_t + hb_t)s_{t-m+h}$$

onde:

- l_t é o componente de nível
- b_t é o componente de tendência
- s_t é o componente de sazonalidade
- t é o tempo
- m é o período da sazonalidade
- h é o horizonte de previsão
- α, β e γ são os parâmetros de suavização

2.2.3 Validação dos métodos de previsão

Para avaliar e validar a acurácia de um modelo, são calculadas métricas correspondentes aos erros dos valores previstos, ou seja, o quão distante estão dos valores reais. Logo, segundo (HYNDMAN, 2014), uma boa prática é separar os dados das bases utilizadas em duas partes, uma de **treino** e outra de **teste**. A parte de treino é utilizada nos métodos de previsão para "treinar" o sistema, capacitando-o a calcular os valores futuros desejados. Já a parte de teste é utilizada para comparar com os valores previstos, permitindo dessa forma avaliar a precisão das previsões. Um valor de proporção comum utilizado é de 80% para treino e 20% para teste, mas esses valores podem variar de acordo com o contexto.

Supondo uma série de dados $y_1, ..., y_T$, em que a parte de treino é denominada por $y_1, ..., y_N$ e de teste $y_{N+1}, ..., y_T$, o sistema iria fornecer os próximos $T - N$ valores, e compará-los com a parte de teste. Os erros são obtidos a partir da diferença do valor real e do valor previsto, segundo a fórmula:

$$e_t = y_t - \hat{y}_{t|N}, \textit{para t = N+1,...,T}$$

onde:

- e_t é o valor do erro
- y_t é o valor real
- $\hat{y}_{t|N}$ é o valor previsto

Existem vários tipos de métricas que podem ser utilizadas para avaliação da previsão em uma série temporal. A determinação de qual utilizar depende bastante do experimento, do que se está tentando prever e como são as bases de dados.

- **MAE (*Mean absolute error*)**

$$MAE = \frac{\sum_{i=1}^{n} |e_i|}{n}$$

É a média da soma dos erros absolutos encontrados. Utiliza a mesma escala do dados que está sendo utilizado, por isso não é indicado para experimentos que estejam utilizando séries de dados com diferentes escalas. Essa métrica é bastante popular por ser fácil de interpretar e calcular. Quanto mais próximo de 0 (zero) for o resultado, melhor.

- **RMSE (*Root mean squared error*)**

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} e_i^2}{n}}$$

É a raiz quadrada da média da soma dos quadrados dos erros. Como característica, o RMSE atribui um maior peso aos erros de maior magnitude, demonstrando ser uma métrica deveras útil em situações onde os maiores erros são indesejáveis. Por isso, ele é bastante sensível a *outliers*. Seus resultados nunca são negativos, logo, quanto menor for, melhor.

- **MAPE (*Mean absolute percentage error*)**

$$MAPE = \frac{1}{n} * \sum_{i=1}^{n} |\frac{e_i}{y_i}|$$

É o erro percentual médio absoluto. Por utilizar valores percentuais, apresenta uma vantagem em relação aos demais apresentados, que é a da independência de escala, por isso é muito utilizado em experimentos onde há dados com escalas diferentes.

2.3 Programação paralela

A programação paralela, ou computação paralela, é um modelo de computação no qual duas ou mais operações são executadas concorrentemente. De acordo com (DIAS, 2010), a demanda computacional para resolver problemas de maior escala aumentou no decorrer dos anos, o que vem fortalecendo o uso e desenvolvimento de algoritmos que consigam processar mais dados simultaneamente, reduzindo assim o tempo para obtenção dos resultados.

Em meados do século XX, os hardwares fabricados não seguiam um padrão, com cada fabricante desenvolvendo interfaces dedicadas às suas aplicações. Isso dificultava o uso de *software* que executassem tarefas em paralelo, pois os custos de adaptação do *hardware* eram altos, como também os custos de uma equipe de desenvolvimento de software para esse fim. No entanto, os processadores se deparam com dificuldades cada vez maiores de aumentar sua frequência de clock, devido a aspectos físicos principalmente.

Com isso, a necessidade de se realizar as tarefas em paralelo veio à tona, pois é uma forma de reduzir o tempo de processamento sem necessariamente depender da frequência de clock do processador. Na última década, processadores com mais de um núcleo de processamento foram se tornando cada vez mais comuns, tanto no ambiente corporativo quanto no doméstico, permitindo dessa forma o uso de soluções que utilizam paralelismo.

2.3.1 Apache Spark

Existem ferramentas dedicadas a realizar processamento em paralelo, sendo uma delas o Apache Spark[2]. O Apache Spark é um framework de computação distribuída de código aberto suportada pela Fundação Apache, utilizada por grandes empresas da área de tecnologia[3], principalmente em projetos que envolvem Big Data e processamento de dados em larga escala. Essa capacidade de lidar com muitos dados provém do modelo de programação utilizado, o *MapReduce*, descrito em (DEAN; GHEMAWAT, 2008), que utiliza o paradigma funcional para permitir o paralelismo com operações simples, distribuindo os dados em vários nós (*map*) e agregando os resultados (*reduce*).

Todos os seus componentes estão integrados no Spark, não havendo a necessidade de utilizar ferramentas externas para concluir seus trabalhos, como no Apache Hadoop[4]. Sua arquitetura pode ser conferida na Figura 6. A MLib é sua biblioteca de Aprendizado de Máquina, contando com algoritmos prontos para atividades relevantes

[2] Apache Spark - https://spark.apache.org/docs/latest/
[3] Empresas que usam o Spark - https://data-flair.training/blogs/spark-careers-job-opportunity/
[4] Apache Hadoop - https://hadoop.apache.org/

nessa área, como clustering. O Spark Streaming permite o processamento de dados em tempo real. O SQL realiza consultas aos dados armazenados na sintaxe SQL. O GraphX é utilizado para realizar operações em grafos. O Spark Core contém as funções básicas de processamento, com as implementações nas linguagens demonstradas (WOODIE, 2019).

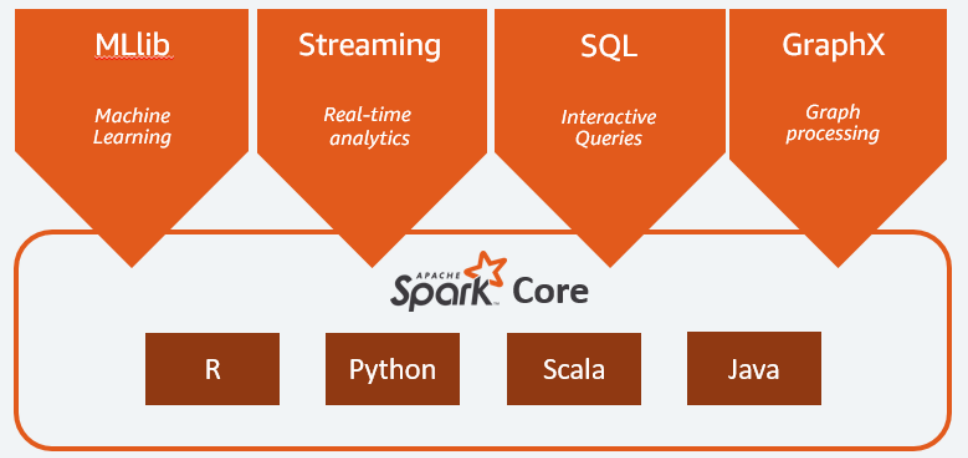


Figura 6 – Componentes do Apache Spark
fonte: (WOODIE, 2019)

Na Figura 7 é possível ver a arquitetura de uma aplicação Spark[5]. O *Driver Program* é responsável por gerenciar a criação do SparkContext e executar as instruções definidas pelo código da aplicação. Os *Workers* são as máquinas que executarão as tarefas delegadas pelo *Driver Program*. O *Cluster Manager* é um componente essencial apenas se o Spark estiver sendo executado em um cluster, gerenciando as máquinas que serão utilizadas como *workers*. Se o Spark for utilizado em uma máquina isolada, essa máquina assumirá o papel tanto de *Driver Program* quanto de *Worker*.

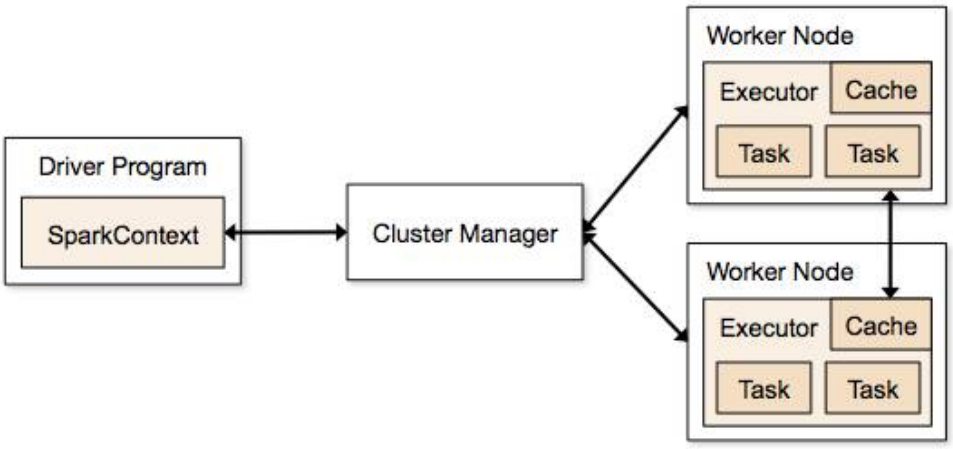


Figura 7 – Arquitetura do Apache Spark
fonte: (SANTANA, 2016)

5

3 Trabalhos Relacionados

A preocupação com a previsão de demanda de energia elétrica não é algo tão recente, no entanto vem ganhando mais espaço com o passar dos anos devido à preocupação com uma melhor distribuição de energia, evitando gastos desnecessários e apagões. Neste trabalho serão abordados projetos e estudos que utilizam métodos de previsão para demanda de energia, a curto e longo prazo, que foram utilizados na fundamentação deste trabalho. Na Tabela 1 estão sumarizados os métodos utilizados, e os de melhor desempenho, em cada trabalho.

(GONTIJO et al., 2017) realizou um estudo comparativo de métodos de previsão de consumo de energia no mercado industrial brasileiro. Os métodos utilizados para o trabalho foram o ARMA, Suavização Exponencial, Média Móvel e Média Móvel Ponderada. Os dados de consumo utilizados foram o consumo de energia industrial no Brasil entre abril de 2007 e janeiro de 2017. Como resultado, verificou-se que o método com melhor resultado foi o ARMA, o qual apresentou o menor erro, seja nas previsões mensais quanto anuais. O presente trabalho se propõe a realizar um trabalho similar a (GONTIJO et al., 2017), porém com o foco no consumo residencial, além de contar com o processamento paralelizado dos dados.

O trabalho de (TAYLOR; MENEZES; McSharry, 2006) tem como objetivo avaliar seis métodos univariados de previsão de demanda de energia a curto prazo, visando a previsão de consumo de um dia a frente. Os dados utilizados são provenientes de consumo de energia elétrica do estado brasileiro do Rio de Janeiro, da Inglaterra e Gales. Segundo o trabalho, séries temporais univariadas são suficientes para realizar as previsões propostas pelos autores. Dentre os métodos utilizados, estão ARIMA e Suavização Exponencial Holt-Winters modificada para suportar dupla sazonalidade das séries temporais utilizadas, os dois métodos a serem utilizados no presente trabalho. Uma das conclusões foi a de que o modelo Holt-Winters utilizado foi considerado o mais simples e rápido de se implementar, além de demonstrar uma boa performance nas suas previsões. Essa foi uma das contribuições deste trabalho para o estudo realizado, auxiliando na definição dos métodos de previsão utilizados.

(CLEMENTS; HURN; LI, 2016) se propõe a demonstrar que um modelo de séries temporais de equações múltiplas, estimado pela aplicação repetida de mínimos quadrados ordinários, consegue apresentar resultados tão bons, ou melhores, quanto os resultados de modelos de previsão não-lineares e não-paramétricos mais complexos. Considerando que fatores como temperatura e feriados públicos influenciam no consumo de energia elétrica, é elaborado um modelo de múltiplas equações capaz

de obter bons resultados. Utilizando uma base de dados de 11 anos do mercado de energia Australiano, ao comparar com modelos como ARIMA e Holt-Winters de dupla sazonalidade, o modelo de múltiplas equações apresentou um MAPE menor, indicando uma melhor eficácia para o cenário proposto.

(VEIGA; TORTATO; SILVA, 2014) realiza um trabalho de previsão de demanda para empresas de varejo do ramo alimentício, utilizando os métodos ARIMA e Holt-Winters. No trabalho, é dito que métodos de previsão seguem duas etapas: análise de série temporal e seleção do melhor modelo de previsão para o conjunto de dados. Como resultados, verificou-se que o Holt-Winters se mostrou o melhor modelo para os dados analisados. Um dos destaques deste trabalho são as métricas de precisão e eficácia utilizadas, que serviram como base para análise dos resultados deste estudo.

(TRATAR; STRMčNIK, 2016) utiliza modelos de previsão baseados em regressão múltipla e suavização exponencial para prever consumo de carga térmica. O artigo utiliza uma base de dados de uma empresa eslovena de produção, distribuição e fornecimento de energia, incluindo energia térmica na forma de água aquecida, vapor e resfriamento, no intervalo entre setembro de 2008 e fevereiro de 2013. Ela já utiliza um sistema de previsão de demanda de um dia a frente, porém simples, mostrando um grande potencial para implementação de um sistema mais complexo e produtivo. Foram testados os modelos de previsão de Regressão Múltipla e Holt-Winters para previsões de curto e longo prazo, com diferentes periodizações. Os resultados mostraram que, para previsões diárias e semanais a curto prazo, a Regressão Múltipla atendia bem, e para previsões mensais a curto prazo e qualquer uma a longo prazo, o modelo Holt-Winters apresentou os melhores resultados.

(CHUJAI; KERDPRASOP; KERDPRASOP, 2013) tem como objetivo encontrar um modelo de previsão de consumo de energia elétrica em uma residência e o melhor período de previsão (diário, semanal, mensal, trimestral). A base de dados utilizada é um histórico de consumo de uma moradia localizada na cidade francesa de Sceaux, no intervalo entre dezembro de 2006 e novembro de 2010. Os modelos utilizados para teste foram o ARIMA e o ARMA, e utilizou como métrica de desempenho o AIC (*Akaike Information Criterion*) e RMSE (*Root Mean Squared Error*), onde quanto menor o valor de ambos, melhor o desempenho do respectivo método. A base de dados não estava completamente preenchida para se realizar os estudos, contando com alguns registros sem valores. Para corrigir essa situação, os autores preencheram os campos vazios com os valores de registros anteriores, assumindo a premissa de que os dados atuais serão semelhantes aos anteriores. Como resultados, verificou-se que o método ARIMA apresentou melhores resultados em previsões mensais e trimestrais, enquanto o ARMA foi melhor em previsões diárias e semanais. Para ambos, o melhor período foi o de curto prazo.

Trabalhos	Tipo de previsão	ARMA/ARIMA	Suav. Exp./Holt-Winters	Médias Móveis	Modelo Equações Múltiplas	Regressão Múltipla	Rede Neural	Análise de componentes principais	Melhor
(GONTIJO et al., 2017)	Consumo energia elétrica	X	X	X					ARMA
(TAYLOR; MENEZES; McSharry, 2006)	Consumo energia elétrica	X	X				X	X	Holt-Winters
(CLEMENTS; HURN; LI, 2016)	Consumo energia elétrica	X	X		X				Modelo Equações Múltiplas
(VEIGA; TORTATO; SILVA, 2014)	Demanda de alimentos	X	X						Holt-Winters
(TRATAR; STRMčNIK, 2016)	Consumo de carga térmica		X			X			Regressão Múltipla - Previsões diárias e semanais; Holt-Winters - Previsões mensais
(CHUJAI; KERDPRASOP; KERDPRASOP, 2013)	Consumo energia elétrica	X							ARMA - Previsões diárias e semanais; ARIMA - Previsões mensais e trimestrais

Tabela 1 – Métodos utilizados em cada trabalho e os avaliados como melhores para cada situação

4 Metodologia

Esta seção descreve as ferramentas, base de dados e etapas do sistema desenvolvido para realizar as previsões. Foi escolhido o tema de consumo de energia elétrica por ser um assunto deveras relevante no cenário mundial, dado a crescente preocupação com o meio ambiente(DINO, 2017), ser capaz de dimensionar adequadamente um sistema de energia elétrica residencial é um dos fatores que ajudam na economia desse recurso essencial na sociedade.

No entanto, é notório que a quantidade de dispositivos utilizados para as mais diversas finalidades, tanto industriais quanto domésticos, vem crescendo bastante nos últimos anos. Isso resulta em um imenso volume de dados gerados em um curto período de tempo, requisitando que o sistema que for analisá-los tenha de suportar processar esse volume. Portanto, a utilização de computação paralela é de suma importância para o funcionamento do sistema de previsão proposto, agregando características como resiliência e robustez.

4.1 Base de dados

A base de dados utilizada neste projeto é fornecida pela DEBS (*Distributed and Event-Based Systems*), uma conferência anual sobre Sistemas Distribuídos que ocorre em várias cidades do mundo. Todo ano ela oferece uma competição, denominada *Grand Challenges*, e na edição de 2014[1], o tema foi previsão de energia elétrica residencial.

Campo	Descrição	Tipo do dado
id	identificador único do registro	32 bits *unsigned intenger*
timestamp	data e hora do registro no formato timestamp	32 bits *unsigned intenger*
value	valor do consumo	32 bits *floating point*
property	tipo do valor	*boolean*
plug_id	identificador único da tomada do cômodo	32 bits *unsigned intenger*
household_id	identificador único do cômodo da casa	32 bits *unsigned intenger*
house_id	identificador único da casa	32 bits *unsigned intenger*

Tabela 2 – Campos presentes na base de dado do DEBS *Grand Challenges* 2014

A base do DEBS é formada de dados coletados de cada tomada presente nas residências na ordem de milissegundos, condensados em um único arquivo de formato **csv**, cujo tamanho é de 136 GB. Ela contempla 4,055 bilhões de registros de 2125

[1] DEBS *Grand Challenges* 2014 - https://debs.org/grand-challenges/2014/

tomadas distribuídas em 40 residências. O período de coleta parte de 01 de setembro de 2013, às 00:00:00h, a 30 de setembro de 2013, às 23:59:59h. Esses dados podem ser verificados na Tabela 3.

Quantidade total de residências	40
Quantidade total de tomadas	2125
Quantidade total de registros (bilhões)	4,055
Data de início dos registros (UTC)	01/09/2013 - 00h00m00s
Data final dos registros (UTC)	30/09/2013 - 23h59h59s

Tabela 3 – Informações gerais da base de dados

A Tabela 2 contém os campos que constituem a base de dados escolhida. Pode-se constatar que os interruptores são organizados por cada cômodo das casas, que por sua vez os cômodos são organizados por cada casa. Isso permite uma medida de consumo bem específica das residências.

O campo “property” se refere ao tipo do valor de consumo registrado na coluna “value”, que são **work (0)** e **load (1)**. O tipo “work” é um valor acumulado de consumo desde o início da sua operação, na unidade de **kWh (Quilowatt-hora)**, tendendo apenas a aumentar, exceto se a respectiva tomada for reiniciada, ou “resetada”, fazendo com que esse valor volte a zero reiniciando o processo de acumulação. Já o tipo “load” se refere ao valor instantâneo do consumo, na unidade de **W (Watt)**.

4.2 Métodos de previsão

A linguagem de programação utilizada para desenvolver o sistema proposto neste trabalho foi a linguagem R[2], cuja competência para projetos que envolvem conhecimentos matemáticos e estatísticos é reconhecida na comunidade de desenvolvedores. As bibliotecas utilizadas para executar os métodos de previsão são referentes à linguagem R.

Para o experimento, foram utilizados os métodos de previsão ARIMA e Holt-Winters. A implementação do método ARIMA foi obtido na biblioteca `forecast`[3], utilizando a função `auto.arima()`. A função `auto.arima()` retorna o melhor modelo ARIMA para a série passada como parâmetro, de acordo com os Critérios de Informação AIC, AICc e BIC, além de uma série de valores ajustados de acordo com os parâmetros estimados.

Já o método Holt-Winters, foi utilizada a função “HoltWinters”, oriunda da biblioteca `stats`[4]. Essa função estima os valores dos parâmetros α, β e γ para série passada

[2] Linguagem R - https://www.r-project.org/
[3] Biblioteca forecast - https://github.com/robjhyndman/forecast
[4] Biblioteca stats - https://stat.ethz.ch/R-manual/R-devel/library/stats/html/00Index.html

como parâmetro, visando a obtenção do menor erro quadrático, e gera uma série de valores ajustados prontos para serem utilizados na função de previsão.

A função de previsão utilizada foi a `forecast()`, fornecida pela biblioteca `forecast`, a mesma que fornece a função `auto.arima()`. Recebendo como parâmetro os resultados das funções `auto.arima()` e `HoltWinters()` (cada um por vez), a janela de previsão e o nível de confiança, são gerados os valores futuros, com seus máximos, mínimos e média previstos para o respectivo instante de tempo. A janela de previsão define o intervalo de tempo que vai ser previsto.

4.3 Métrica de desempenho

A métrica utilizada para comparar os métodos de previsão foi o **RMSE**. Para cada conjunto de valores previstos, foi calculado o respectivo índice, a fim de comparar seus valores e determinar qual método apresentou o melhor desempenho no experimento. Outros métodos considerados apresentam características indesejáveis para o cenário deste projeto. Por exemplo, o MAPE não é indicado para séries que apresentem valores reais iguais a zero, pois seu cálculo envolve uma operação de divisão onde o denominador é o valor real. Já o MAE, nem sempre é evidente o tamanho relativo do erro, dificultando o contraste entre grandes e pequenos erros.

4.4 Infraestrutura

A infraestrutura utilizada para executar os experimentos está descrita na Tabela 4. Durante a execução dos experimentos, foram coletados dados de consumo da máquina, como uso de CPU e uso de memória RAM. Esses dados foram utilizados para demonstrar o quanto de recursos da máquina é requisitado para o devido funcionamento do sistema desenvolvido.

CPU	i5 7200u 2.5GHz
RAM	8GB DDR4 2133MHz
ARMAZENAMENTO	1 TB HDD 5400 RPM 32 GB SSD cache
SISTEMA OPERACIONAL	Fedora Workstation 29 Kernel Linux 5.0.7
IDE	RStudio

Tabela 4 – Configuração da máquina que foi utilizada para executar os testes

5 Sistema ReCast

Neste capítulo é apresentado o protótipo do ReCast, sistema proposto para realizar previsões de demanda de consumo de energia elétrica residencial, com capacidade de processar um grande volume de dados. Na seção 5.1 são detalhados seu funcionamento, as funções desenvolvidas e bibliotecas utilizadas.

Além disso, na seção 5.2, é feita a análise dos resultados obtidos. Na seção 5.2.1 são abordados os resultados do cenário mais amplo, previsão de consumo da região, utilizando a métrica de erro RMSE, comparação de valores previstos e valores reais e gráficos de autocorrelação. Na seção 5.2.2, é abordado o cenário de previsão de consumo de cada residência, utilizando apenas a métrica de erro RMSE de ambos os métodos de previsão. Com os valores de RMSE, são criadas duas séries, cada uma contendo os valores do respectivo método de previsão, e em seguida, foi realizado um teste de hipótese entre as médias desses valores para averiguar a possibilidade de igualdade entre elas, além de analisar a média, mediana e a distribuição desses dados.

5.1 Sistema ReCast

O sistema de previsão de demanda de energia elétrica residencial proposto nesse trabalho recebeu o nome de **ReCast**, que realiza o tratamento nos dados a serem utilizados como parâmetros nas funções de previsão ARIMA e Holt-Winters, além da aplicação dos métodos em si. Foi desenvolvido utilizando a linguagem de programação R devido a sua competência reconhecida em projetos na área de análise de dados, oferecendo um ambiente ideal para manipulação de dataframes e de funções estatísticas sem o uso demasiado de bibliotecas externas.

O ReCast utiliza o Apache Spark para paralelizar o processamento durante toda sua execução, desde a leitura dos dados, filtragem e aplicação dos métodos de previsão. Sem o uso desse framework, não seria possível nem ler o arquivo original, dadas as características da linguagem R, que se limita a quantidade de memória RAM instalada na máquina em execução. O sistema ReCast realiza todo o procedimento de forma automática, até os resultados das previsões, sem que o usuário necessite intervir durante sua execução.

5.1.1 Preparação inicial e leitura dos dados

Inicialmente, são carregadas as bibliotecas a serem utilizadas, com destaque para a `sparklyr`, `dplyr` e `tseries`, além dos arquivos contendo as funções desenvolvidas utilizando as implementações de ambos os métodos de previsão. Em seguida, é estabelecida a conexão com o ambiente do Apache Spark, por meio da biblioteca `sparklyr`[1], permitindo dessa forma utilizar os recursos deste framework na aplicação. Além disso, é criada uma sessão `sparkR`, utilizando a biblioteca `SparkR`[2], um outro pacote que permite utilizar código em R com funções do Spark, a qual é utilizada em complemento a `sparklyr`, aplicando as funções desenvolvidas nos dados de forma distribuída por meio de recursos do tipo **MapReduce**.

Também é configurada a quantidade de memória que será compartilhada com o Spark e a de núcleos de processamento. Com a conexão estabelecida, é realizada a leitura do arquivo CSV original da base dados, disponível na memória de armazenamento na máquina em que o ReCast está sendo executado, utilizando os recursos do Spark e armazenando seu conteúdo em um dataframe. As colunas, a princípio, não apresentam nomes, mas é resolvido esse problema ao utilizar a função `names` para nomeá-las, a qual recebe uma lista de strings contendo os respectivos nomes e aplica no dataframe passado como parâmetro. O procedimento de configuração e conexão com o Spark é demonstrado no Quadro 5.1, e da leitura do arquivo da base de dados no Quadro 5.2.

Quadro 5.1 – Conexão com o Spark

```
######################################################################
### Iniciar conexao com o spark
######################################################################
config = spark_config()
config$`sparklyr.backend.threads` <- "4"
config$`sparklyr.shell.driver-memory` <- "4G"
config$`sparklyr.shell.executor-memory` <- "4G"
config$`spark.executor.memoryOverhead` <- "1g"
config$`sparklyr.shell.driver-java-options` <- paste0("-Djava.io.tmpdir="
   ↪ , "/home/***/.tmp")
## Conexao utilizando o sparklyr
sc <- sparklyr::spark_connect(master = "local",
                              spark_home = "/opt/spark",
                              config = config)
## Inicia o SparkR
```

[1] Biblioteca sparklyr - https://spark.rstudio.com/
[2] Biblioteca SparkR - https://spark.apache.org/docs/latest/sparkr.html

```
sparkR.session(sparkHome = "/opt/spark")
```

Quadro 5.2 – Leitura da base de dados

```
df <- spark_read_csv(sc, "file:///home/***/UFRPE/BSI/TCC/sorted.csv",
   ↪ header = FALSE)
```

5.1.2 Filtragem dos dados

São executados dois cenários. Um deles mais amplo, que considera os valores sem dividir por residências, prevendo o consumo de energia total da região que contempla todas as moradias contidas na base de dados. Outro mais específico, que divide o cálculo das previsões de consumo por cada residência, obtendo dessa forma os valores previstos para cada moradia individualmente.

O processo de tratamento dos dados para aplicação dos métodos de previsão é similar para ambos os cenários, sendo a principal diferença o início de cada um. Enquanto na região não é preciso filtrar a residência antes de começar as demais operações, pois deseja-se o consumo geral, no consumo por residência, é necessário filtrar a moradia antes de executar as demais instruções. No Quadro 5.3 é demonstrado a instrução que filtra o tipo de consumo desejado para gerar as previsões, que é o instantâneo (ou `load`).

Quadro 5.3 – Filtragem de consumo instantâneo

```
## Filtra o consumo instantâneo
df_all_present_consumption <- df %>% filter(work_or_load == 1)
```

Os dados filtrados são armazenados em um dataframe. Independente do cenário, é feito todo o tratamento necessário para extrair o consumo horário durante o período de tempo compreendido pela base de dados. O processo de tratamento utiliza a sintaxe e funções baseadas na biblioteca `dplyr`.

Na previsão por residência, os dados são agrupados de acordo com a data e hora, cômodo e tomada, nessa ordem. No caso da previsão por região, os dados são agrupados por data e hora, residência, cômodo e tomada, nessa ordem. Logo após, é criada uma nova coluna contendo a média de consumo da respectiva tomada naquele momento, denominada “consumo”, cujo valor é calculado utilizando os valores de consumo filtrados anteriormente. Para obter o consumo total da residência por hora, essas médias calculadas são somadas, obtendo como resultado um dataframe contendo data e hora e a média de consumo total da respectiva residência ou da região. Esse valor já é o consumo em Wh (Watt-hora), de acordo com a fórmula para calcular o consumo:

$$C = P * \Delta t$$

onde:

- C é o consumo em Watt-Hora
- P é a potência dissipada
- Δt é o intervalo de tempo

Como o intervalo de tempo estipulado para as previsões do sistema é de 1 hora, e o valor do consumo já está na unidade Watt, multiplicar o valor por 1 equivale ao mesmo valor, logo, esses valores já são o consumo em Wh. No Quadro 5.4 estão demonstradas as instruções de tratamento e extração do consumo horário para ambos os cenários.

Quadro 5.4 – Cálculo do consumo horário por residência e da região

```
####################
## Por residência
####################
## Formata o timestamp em data e hora
df_house_present_consumption <- df_house_present_consumption %>% mutate(
   ↪ hora = from_unixtime(ts, 'yyyy-MM-dd␣HH'))
## Cria uma coluna com a media do consumo de cada plug por comodo
df_house_hourly_mean_consumption <- df_house_present_consumption %>%
   ↪ group_by(hora, house_id, household_id, plug_id) %>% arrange(hora,
   ↪ household_id, plug_id) %>% summarise(consumo = mean(value))
## Consumo horário da residência
df_house_hourly_consumption <- df_house_hourly_mean_consumption %>% group
   ↪ _by(hora, house_id) %>% arrange(hora) %>% summarise(total = sum(
   ↪ consumo))
####################
## Da região
####################
## Formata o timestamp em data e hora
df_all_present_consumption <- df_all_present_consumption %>% mutate(hora
   ↪ = from_unixtime(ts, 'yyyy-MM-dd␣HH'))
## Cria uma coluna com a media do consumo de cada plug por comodo
df_all_hourly_mean_consumption <- df_all_present_consumption %>% group_by
   ↪ (hora, house_id, household_id, plug_id) %>% arrange(hora, household
   ↪ _id, plug_id) %>% summarise(consumo = mean(value))
```

```
## Consumo horário da região
df_all_hourly_consumption <- df_all_hourly_mean_consumption %>% group_by(
    ↪ hora) %>% arrange(hora) %>% summarise(total = sum(consumo))
```

5.1.3 Aplicação dos métodos de previsão

Cada dataframe, no final do processo de tratamento, é transferido para o ambiente R, a fim de poder aplicar código R nativo a esse dataframe gerado com auxílio do Spark. Esse dataframe é inserido em uma lista, criada exclusivamente para armazená-lo. Dataframes criados com o Spark, mesmo com as bibliotecas para o R, não são compatíveis com as funções padrões do R, necessitando utilizar funções dedicadas, oriundas das bibliotecas utilizadas. Durante o desenvolvimento desse projeto, não foram utilizadas bibliotecas desenvolvidas para o Spark e R abordando séries temporais devido a falta de suporte dos desenvolvedores e falta de documentação para auxiliar no desenvolvimento[3].

Para contornar esse obstáculo, é utilizada a função `spark.lapply()`, que recebe dois parâmetros, uma lista de dataframes e uma função. Durante sua execução, ela aplica a função em cada elemento da lista, função essa em que seu conteúdo é desenvolvido utilizando os objetos padrões do R. Contudo, a execução não é realizada no ambiente R convencional, ao invés disso todo o processamento é realizado de maneira distribuída e paralela no *framework* Spark. Na Quadro 5.5 é demonstrado as instruções dessa operações.

Quadro 5.5 – Código de aplicação dos métodos

```
#####################################################################
### Aplicacao dos metodos de previsao e Calculo das metricas
#####################################################################
## Coletar o dataframe
df_r <- sparklyr::collect(df_***_hourly_consumption)

## Cria a lista para adicionar os dataframes
list_df <- list()
list_df[[1]] <- df_r

## Aplica os métodos
spark.lapply(list_df, run_arima_df)
spark.lapply(list_df, run_hw_df)
```

[3] Biblioteca spark-ts - https://github.com/sryza/spark-timeseries

Foram elaboradas duas funções, uma para cada método de previsão, que são a `run_arima()` e `run_hw()`, recebendo como parâmetro, cada uma, um dataframe. Essas funções são passadas como parâmetro para a função `spark.lapply()`.

No conteúdo das funções, o tamanho das partes de "treino" e "teste" foi definido empiricamente, de acordo com a execução dos experimentos, onde se chegou a um valor de erro aceitável ao tentar prever uma janela de 72 horas. Para ambos os métodos, os ajustes necessários para tornar a série estacionária, caso seja preciso, serão realizados pelas funções dos respectivos métodos. Os outliers são removidos utilizando quartis e intervalo interquartil (IQR), removendo os valores maiores que $1,5 * IQR$ acima do terceiro quartil e $1,5 * IQR$ abaixo do primeiro quartil.

Dentro do corpo de cada função são carregadas as bibliotecas necessárias para executar as instruções contidas na própria função, com destaque para as bibliotecas `tseries`, `anytime` e `forecast`.

Quadro 5.6 – Principais trechos da função ARIMA

```
FORECAST_WINDOW <- 72
...
## Eliminar outliers
qnt <- quantile(df$total, probs=c(.25, .75), na.rm = TRUE)
caps <- quantile(df$total, probs=c(.05, .95), na.rm = TRUE)
H <- 1.5 * IQR(df$total, na.rm = TRUE)
df$total[df$total < (qnt[1] - H)] <- caps[1]
df$total[df$total > (qnt[2] + H)] <- caps[2]
coluna_diferenca <- df$total
...
## Estimativa dos parâmetros ARIMA
fit_power <- auto.arima(y = dados_train, stepwise = FALSE,
    ↪ approximation = FALSE, seasonal = TRUE, trace = FALSE, D = 1)
...
## Geração dos valores futuros
arima_forecast <- forecast(object = fit_power, h = FORECAST_WINDOW,
    ↪ level = 95)
...
## Métricas
indices <- accuracy(arima_forecast, x = dados_test)
```

A função `run_arima()` recebe um dataframe contendo o consumo de energia elétrica por hora como parâmetro. Primeiramente, cria uma série temporal com frequência horária, a partir da criação de um objeto do tipo "ts" com o parâmetro recebido na

função, oriundo da biblioteca `tseries`[4]. Divide os dados em "treino" e "teste", onde "teste" possui 72 valores. Em seguida, é escolhida a melhor configuração Arima pela função `auto.arima`, que define os valores dos parâmetros automaticamente de acordo com os valores de AIC, AICc ou BIC. Logo após, é realizado o cálculo dos valores previstos, utilizando a função `forecast`, e um nível de confiança de 95%. A métrica RMSE é calculada no final, por meio da função `accuracy`, fornecida pela biblioteca `forecast`. Na Quadro 5.6 estão demonstradas as principais instruções da função.

Quadro 5.7 – Principais trechos da função Holt-Winters

```
FORECAST_WINDOW <- 72
...
## Eliminar outliers
qnt <- quantile(df$total, probs=c(.25, .75), na.rm = TRUE)
caps <- quantile(df$total, probs=c(.05, .95), na.rm = TRUE)
H <- 1.5 * IQR(df$total, na.rm = TRUE)
df$total[df$total < (qnt[1] - H)] <- caps[1]
df$total[df$total > (qnt[2] + H)] <- caps[2]
coluna_diferenca <- df$total
...
## Coeficientes alpha, beta e gamma
ajuste_holt <- HoltWinters(dados_train)
...
## Previsao usando o Holt-Winters
holt_forecast <- forecast(ajuste_holt, h = FORECAST_WINDOW, level = 95)
...
## Calculo dos indices de erro
indices <- accuracy(holt_forecast, x = dados_test)
```

A função `run_hw()` recebe um dataframe contendo um consumo de energia elétrica por hora como parâmetro. Cria uma série temporal de frequência horária e divide as partes de "treino" e "teste" tal qual a função `run_arima()`. Os coeficientes α, β e γ, e os demais parâmetros, são calculados usando a função **HoltWinters**, da biblioteca `stats`. O cálculo dos valores previstos são realizados utilizando a função `forecast`, com um nível de confiança de 95%. No final, é calculado o RMSE com a função `accuracy`. No Quadro 5.7 estão os principais trechos da respectiva função.

[4] Biblioteca tseries - https://cran.r-project.org/package=tseries

5.1.4 Estacionariedade

Quadro 5.8 – Teste estacionariedade

```
############################
## Teste de estacionariedade
############################
stationarity_test <- kpss.test(ts(df_r_all$total))
```

Para determinar se a série original é ou não estacionária, foi utilizado o teste Kwiatkowski–Phillips–Schmidt–Shin (KPSS), cuja função é oriunda do pacote `tseries` e está demonstrado na Quadro 5.8.

5.2 Análise dos resultados

5.2.1 Análise dos valores previstos da região

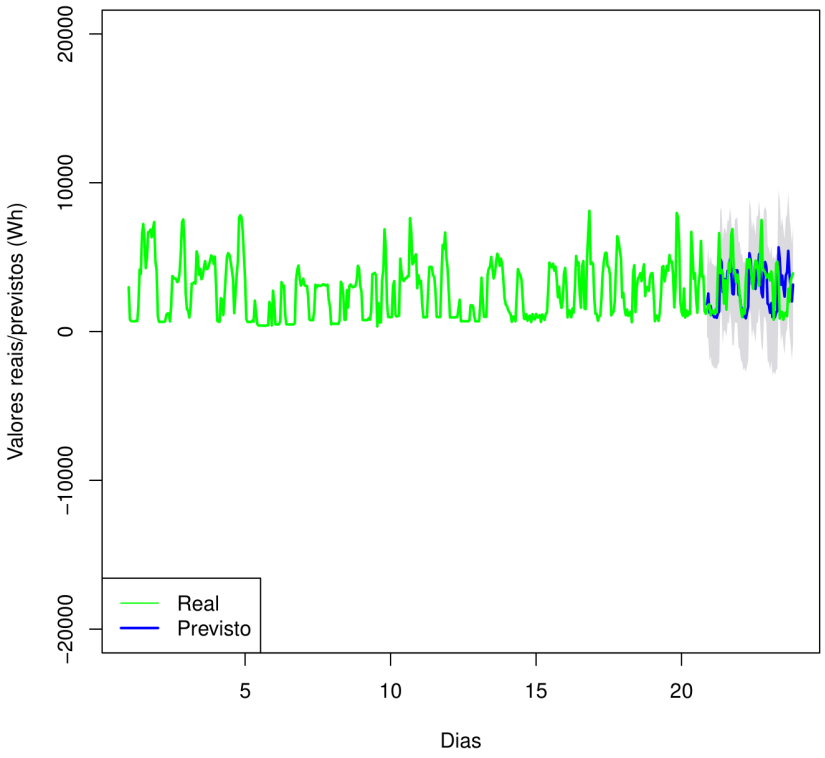
De acordo com os resultados obtidos na previsão da região, verificou-se que os valores previstos em ambos os métodos de previsão não diferem tanto dos valores reais. Utilizando como métrica o RMSE calculado para os valores previstos nos dois métodos, o ARIMA demonstrou uma perceptível vantagem em relação ao método Holt-Winters. Na Tabela 5 estão os valores encontrados do RMSE. Enquanto o ARIMA apresentou um erro de 1,89 kWh, o Holt-Winters apresentou 8,73 kWh, aproximadamente.

RMSE	
ARIMA	**Holt-Winters**
1894,46	8729,62

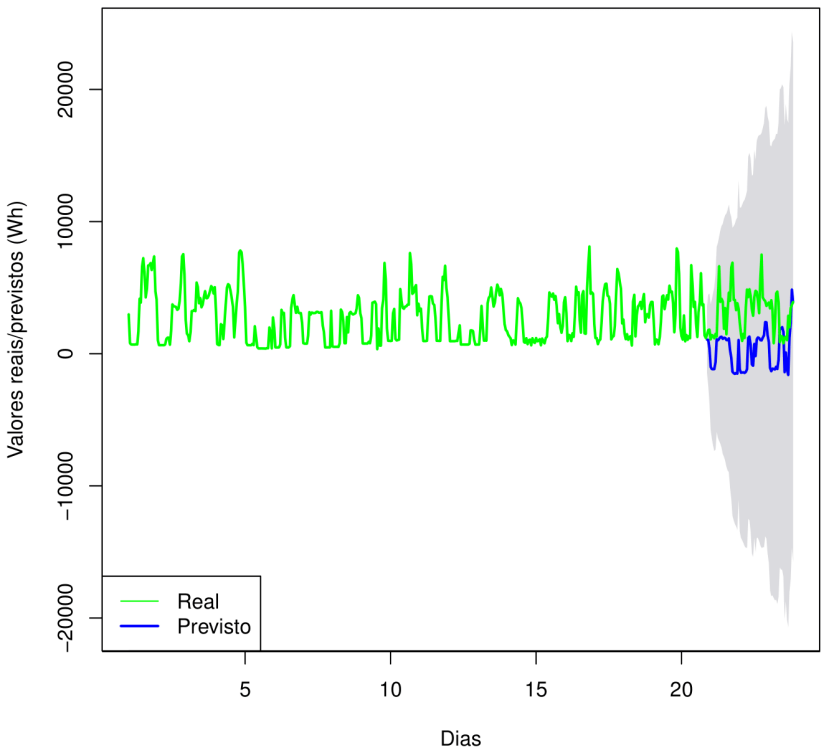
Tabela 5 – Valores da métrica RMSE (em Wh) dos métodos de previsão

De acordo com os gráficos dos valores previstos em comparação com os valores reais utilizando os métodos ARIMA e Holt-Winters, apresentados respectivamente nas Figuras 8a e 8b. É notório que, no início dos valores previstos, o Holt-Winters apresenta uma diferença bem maior que o ARIMA, além do intervalo de confiança ser maior, evidenciado pela área mais escura. Em ambos os métodos foi considerada a sazonalidade da série temporal, e por conta disso, os valores previstos utilizando o método Holt-Winters apresentou valores negativos, pois ele seguiu a tendência dos dados de treinamento.

Os gráficos da função de autocorrelação e autocorrelação parcial estão nas Figuras 9a e 9b. De acordo com os gráficos, pode-se suspeitar que a série em questão é estacionária. O teste KPSS apresentou um *p-value* de **0,1**, a um nível de 95% de

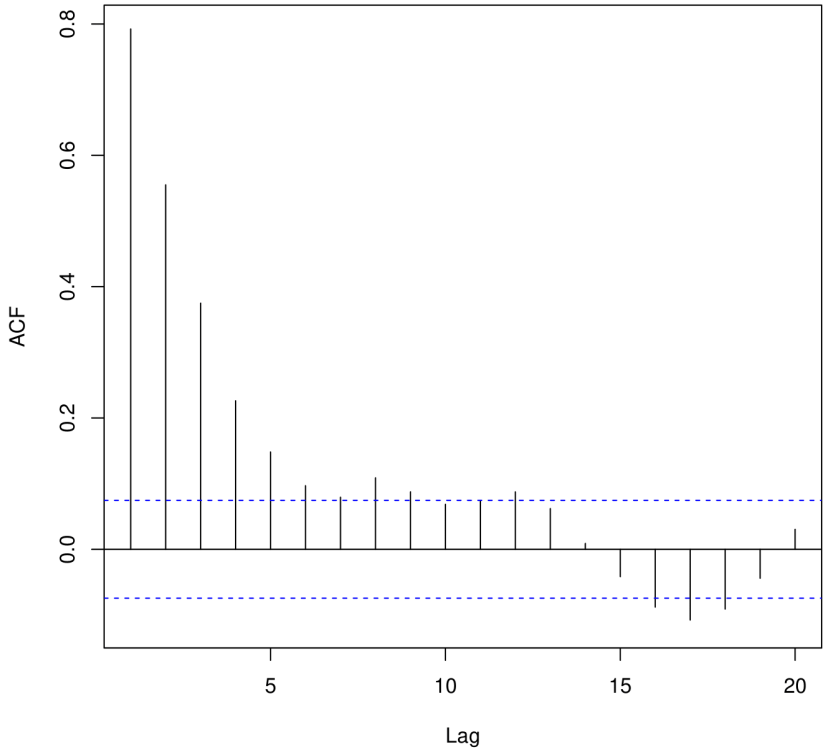


(a) ARIMA

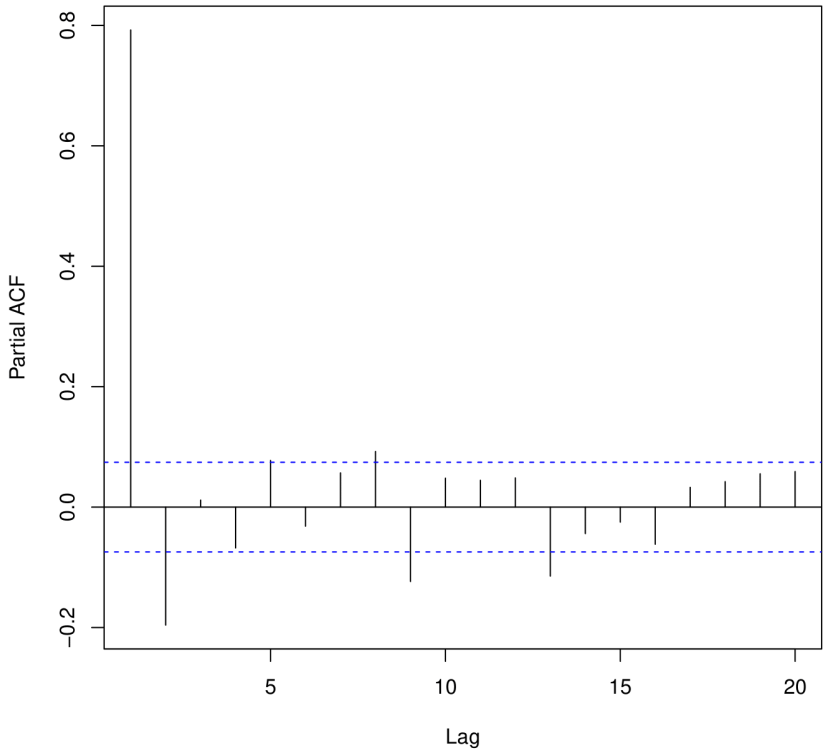


(b) Holt-Winters

Figura 8 – Gráfico dos valores reais e previstos dos métodos de previsão
fonte: próprio autor



(a) Autocorrelação



(b) Autocorrelação parcial

Figura 9 – Gráfico de Autocorrelação e Autocorrelação parcial
fonte: próprio autor

certeza, logo não se rejeita a hipótese nula de que a série em questão seja estacionária. Com ambos os resultados, é plausível considerar a série estacionária.

A função *auto.arima* decidiu por usar um modelo **SARIMA**, cujos parâmetros estão evidenciados na Tabela 6. O parâmetro “D” com valor “1” indica que a sazonalidade está sendo considerada para calcular os valores futuros. Ao analisar a Figura 9b, pode-se entender o porquê do parâmetro “p” e “P” ser igual a “2”, pois a *lag* 3 já está situada dentro da área delimitada pelas linhas tracejadas.

Com relação ao Holt-Winters, os parâmetros calculados para o modelo estão demonstrados na Tabela 7. O parâmetro α com valor próximo de “1” indica que valores passados estão com um peso maior no cálculo dos valores futuros. O parâmetro β com valor “0” indica que a tendência identificada nos dados mais recentes não estão sendo consideradas nos valores futuros. O parâmetro γ com valor igual a “1” indica que a sazonalidade identificada possui o maior peso no cálculo dos valores futuros.

Parâmetros ARIMA	
p	2
d	0
q	0
P	2
D	1
Q	0

Tabela 6 – Valores dos parâmetros SARIMA utilizados

Parâmetros Holt-Winters	
α	0.85
β	0
γ	1

Tabela 7 – Valores dos parâmetros Holt-Winters utilizados

Na Figura 10 os valores previstos de cada método foram plotados utilizando o formato boxplot. Dessa forma, fica evidente o melhor desempenho do ARIMA frente ao Holt-Winters. O valor máximo previsto pelo Holt-Winters, se aproximou do valor mínimo previsto pelo ARIMA, desconsiderando *outliers* em ambos os casos. Percebe-se também que os dados também não apresentam uma simetria perfeita, de acordo com suas medianas

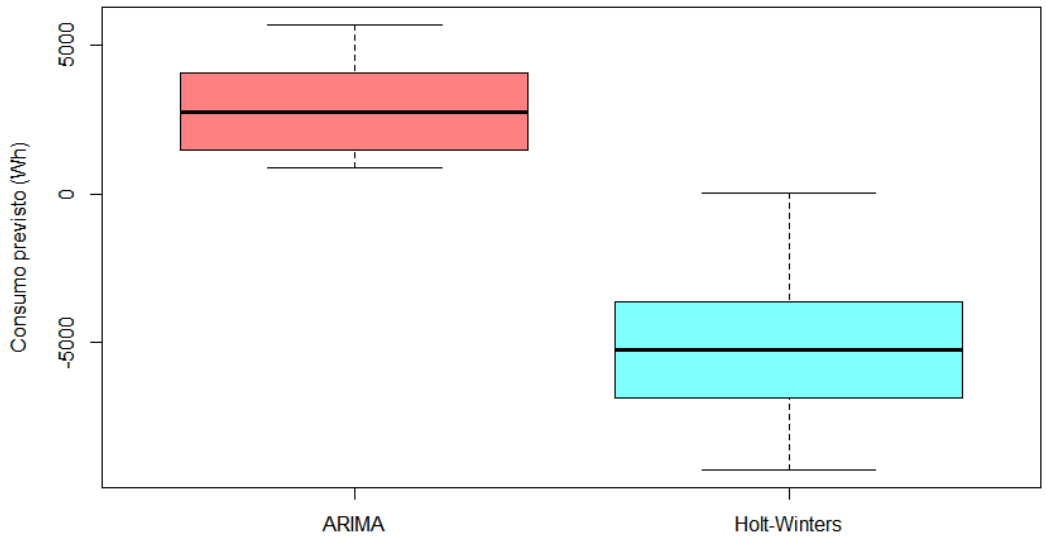


Figura 10 – Boxplots dos valores previstos com ARIMA e Holt-Winters
fonte: próprio autor

5.2.2 Análise dos valores previstos por residência

Foram realizadas as previsões de consumo de cada residência, e na Tabela 8 estão presentes as medidas estatísticas realizadas em cima dos valores do RMSE de cada previsão. A decisão de utilizar o RMSE para calcular as demais métricas (média, mediana, desvio padrão), em vez dos valores futuros, foi feita pensando na objetividade da análise, pois essa métrica sintetiza o quão eficaz foi a previsão em si. O ARIMA demonstrou um melhor resultado que o Holt-Winters, apresentando valores menores.

Foi feito também um teste de hipótese, do tipo **Wilcoxon-Matt-Whitney**, buscando verificar se as médias dos valores de RMSE dos dois métodos são equivalentes. Esse teste foi escolhido devido a sua capacidade de lidar com dados de distribuições não normais, característica da série utilizada nesse experimento. O *p-value* encontrado foi de **0,002337**, logo, por ser menor que 0,05, a um nível de confiança de 95%, rejeita-se a hipótese nula de que as médias são iguais. Então, é aceitável afirmar que as médias são estatisticamente diferentes.

	Média	Mediana	Desvio padrão	Teste de hipótese
ARIMA	713,86	556,76	483,21	Médias diferentes
Holt-Winters	1420,79	963,45	1195,45	

Tabela 8 – Valores estatísticos a respeito do RMSE das previsões das residências

De acordo com a Figura 11 pode-se visualizar que os valores decorrentes do método ARIMA, de um modo geral, são menores que os decorrentes do método Holt-Winters. O Holt-Winters, além de apresentar dois outliers, teve um valor máximo superior ao valor máximo do ARIMA em mais de 1000 unidades, enquanto o valor mínimo foi o mesmo para ambos. Em nenhum dos dois casos os dados apresentaram simetria, de acordo as medianas.

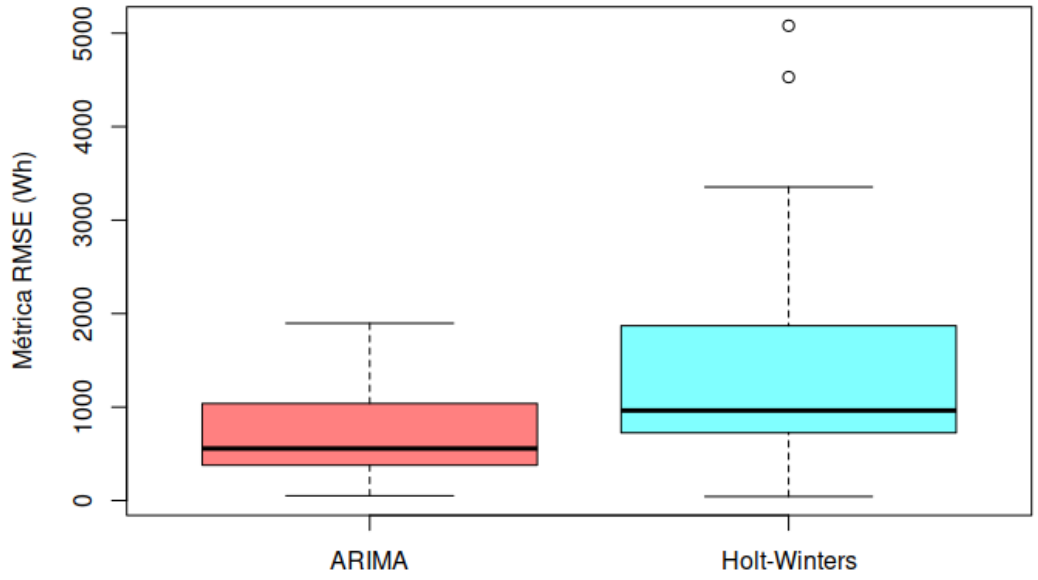
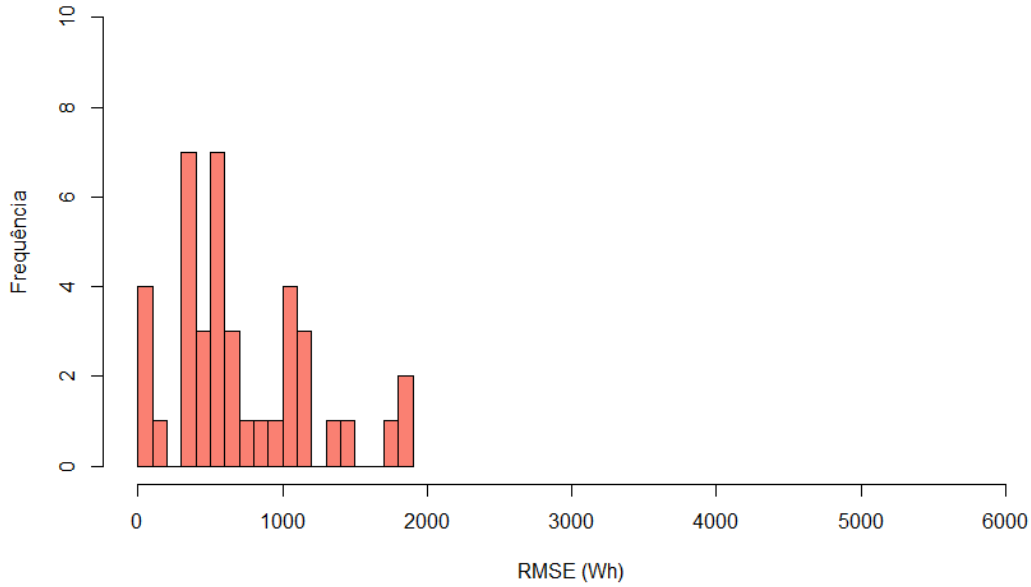


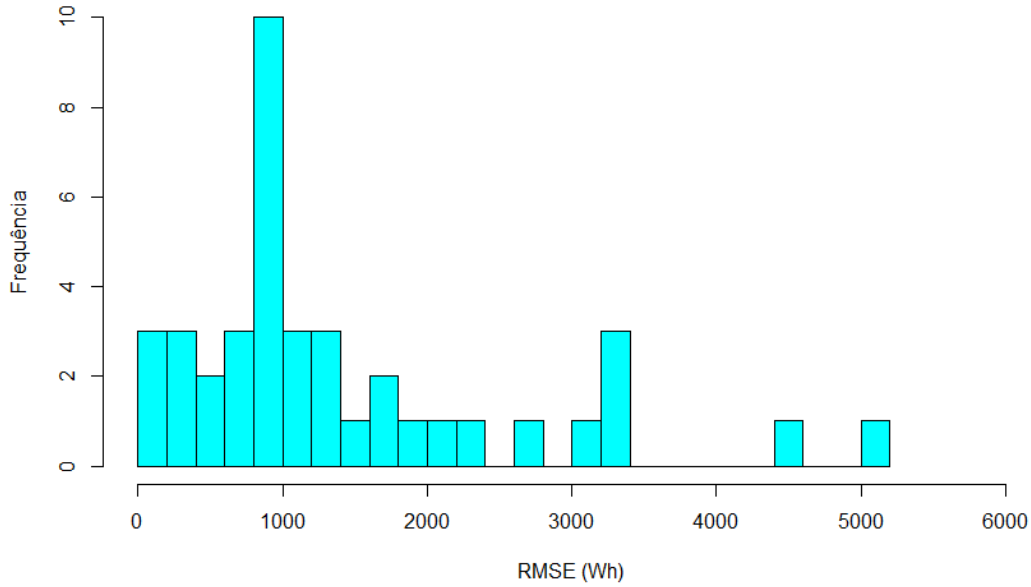
Figura 11 – Boxplots dos valores previstos com ARIMA e Holt-Winters para cada residência

fonte: próprio autor

Nas Figuras 12a e 12b estão os histogramas dos valores de RMSE das previsões das residências. O mais notório ao visualizá-los é o intervalo desses valores. No ARIMA, chega a um máximo de 2000 Wh, enquanto no Holt-Winters chega a quase 6000 Wh. Ambos os gráficos são assimétricos a direita, e o que auxilia nessa análise é o fato das médias serem maiores que as medianas, característica típica desse formato. Além disso, na Figura 12b é possível identificar outliers de valores acima de 4000 Wh.



(a) ARIMA



(b) Holt-Winters

Figura 12 – Histogramas dos valores de RMSE das previsões de cada residência

5.3 Consumo de recursos computacionais

Durante a execução do experimento, também foram coletados os dados de consumo dos recursos da máquina hospedeira. Esses dados foram coletados durante do início ao fim do cenário de previsão por residência partindo da conexão com o Spark, leitura da base de dados original, filtragem dos dados, execução dos métodos ARIMA e Holt-Winters, um após o outro, e cálculo das métricas. Esse cenário foi escolhido por apresentar um maior intervalo de tempo para conclusão, ou seja, o pior caso.

Com relação ao período de execução, a leitura da base de dados original apresentou a mesma duração em ambos os cenários, sendo 3 horas e 44 minutos. A filtragem dos dados e aplicação dos métodos de previsão para a região, incluindo cálculo do RMSE e teste de estacionariedade, apresentou um período de execução de 1 hora e 22 minutos, enquanto as mesmas instruções para a previsão por residência demonstrou um tempo total de execução de 2 dias, 5 horas e 25 minutos. Na Tabela 9 estão as métricas calculadas de acordo com os dados coletados. Os valores próximos entre a média e mediana demonstram que os dados não apresentam uma assimetria considerável, e a maior parte do consumo de RAM e CPU, na maior parte do tempo, foi devido ao processo do experimento.

	CPU (%)	**RAM (%)**
Média	75,03	97,50
Mediana	75,00	98,32
Desvio padrão	16,90	0,03

Tabela 9 – Métricas do consumo de recursos computacionais da máquina hospedeira

6 Conclusões

O respectivo trabalho buscou desenvolver um sistema que pudesse tratar e aplicar métodos de previsão estatística em um grande volume de dados, comparando os resultados obtidos em ambos os métodos para verificar, em circunstâncias mais gerais, qual apresentou o melhor desempenho. Os dados utilizados foram obtidos a partir de um evento internacional de Sistemas Distribuídos, o DEBS, no qual nele é realizada uma competição para análise desses dados. No caso, foi obtido o conjunto de dados da edição de 2014 da competição, que abordou a previsão de consumo elétrico residencial.

Com a necessidade de dar suporte a uma entrada massiva de dados, algo que a linguagem R por si só não consegue lidar, o uso do Apache Spark foi crucial para o desenvolvimento do sistema proposto, pois com ele, utilizando uma máquina doméstica, foi possível tratar todos os dados e extrair os resultados desejados. Houve algumas dificuldades, como documentação escassa e poucos exemplos encontrados nas pesquisas, o que demandou mais tempo de estudo para aplicar as devidas técnicas e conseguir que o sistema funcionasse como esperado. Porém o Spark foi o grande diferencial. O mesmo código desse experimento pode tanto ser utilizado em uma máquina local, como foi o caso devido a falta de acesso a uma maior infraestrutura, quanto em um *cluster* com várias máquinas, trazendo benefícios como ganho de desempenho e escalabilidade. Além disso, suporta também um tamanho maior de arquivo de dados que foi utilizado.

Os métodos de previsão tiveram resultados distintos, sendo perceptível a vantagem do ARIMA frente ao Holt-Winters no cenário utilizado. No teste para prever os dados de consumo da região, o Holt-Winters demonstrou um RMSE aproximadamente 4,6 vezes maior que o ARIMA, o qual teve um RMSE em torno de 1,8 kWh. No cenário de previsão individual de cada residência, a vantagem permaneceu com o ARIMA. Porém, a diferença dos valores de RMSE, em média, foi menor, onde o ARIMA apresentou um RMSE equivalente à metade do apresentado no Holt-Winters. Contudo, no contexto de consumo de energia elétrica, segundo verificado em (DEB et al., 2017), os erros evidenciados em ambos os métodos de previsão estão similares a outros trabalhos.

Referências

AHMAD, A. et al. A review on applications of ann and svm for building electrical energy consumption forecasting. *Renewable and Sustainable Energy Reviews*, v. 33, p. 102–109, May 2014. ISSN 13640321. Citado 2 vezes nas páginas 13 e 14.

ALLASSIGNMENTHELP. *Mind Blowing Information on Cross sectional data*. 2018. Acessado em 02/06/2019. Disponível em: <https://www.allassignmenthelp.com/blog/cross-sectional-data/>. Citado na página 20.

BRAGG, S. *Qualitative forecasting*. 2018. Acessado em 01/06/2019. Disponível em: <https://www.accountingtools.com/articles/what-is-qualitative-forecasting.html>. Citado na página 19.

CHUJAI, P.; KERDPRASOP, N.; KERDPRASOP, K. Time series analysis of household electric consumption with ARIMA and ARMA models. p. 6, 2013. Citado 2 vezes nas páginas 29 e 30.

CLEMENTS, A.; HURN, A.; LI, Z. Forecasting day-ahead electricity load using a multiple equation time series approach. v. 251, n. 2, p. 522–530, 2016. ISSN 03772217. Disponível em: <https://linkinghub.elsevier.com/retrieve/pii/S0377221715011698>. Citado 2 vezes nas páginas 28 e 30.

COCEL, A. *Calor ou frio intenso contribui para aumento no consumo de energia*. 2019. Acessado em 03/06/2019. Disponível em: <https://www.folhadecampolargo.com.br/noticias/geral/calor-ou-frio-intenso-contribui-para-aumento-no-consumo-de-energia-42222>. Citado na página 17.

DANTAS, T. M.; OLIVEIRA, F. L. C.; REPOLHO, H. M. V. Air transportation demand forecast through bagging holt winters methods. v. 59, p. 116–123, 2017. ISSN 09696997. Disponível em: <https://linkinghub.elsevier.com/retrieve/pii/S0969699716302265>. Citado na página 23.

DEAN, J.; GHEMAWAT, S. Mapreduce: simplified data processing on large clusters. *Communications of the ACM*, ACM, v. 51, n. 1, p. 107–113, 2008. Citado na página 26.

DEB, C. et al. A review on time series forecasting techniques for building energy consumption. v. 74, p. 902–924, 2017. ISSN 13640321. Disponível em: <https://linkinghub.elsevier.com/retrieve/pii/S1364032117303155>. Citado 2 vezes nas páginas 16 e 49.

DIAS, B. *PROGRAMAÇÃO DINÂMICA ESTOCÁSTICA E ALGORITMO DE FECHOS CONVEXOS NO PLANEJAMENTO DA OPERAÇÃO DE SISTEMAS HIDROTÉRMICOS*. Tese (Doutorado) — PUC-RIO - PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO, 2010. Certificação digital 0621325/CA. Citado na página 26.

DINO. *Consciência ambiental cresce, mas ainda enfrenta desafios*. 2017. Acessado em 07/05/2019. Disponível em: <https://www.terra.com.br/noticias/dino/consciencia-ambiental-cresce-mas-ainda-enfrenta-desafios,8ea8ba4032d5ff283ac3ad07bb0dea7cmdjy26jg.html>. Citado na página 31.

EMILIANO, P. C. et al. Critérios de informaçao de akaike versus bayesiano: análise comparativa. *19º Simpósio Nacional de Probabilidade e Estatística*, 2010. Citado na página 22.

EMPRESA DE PESQUISA ENERGÉTICA – EPE. *CONSUMO NACIONAL DE ENERGIA ELÉTRICA NA REDE POR CLASSE: 1995 - 2018*. [S.l.], 2018. Disponível em: <http://www.epe.gov.br/pt/publicacoes-dados-abertos/publicacoes/Consumo-Anual-de-Energia-Eletrica-por-classe-nacional>. Citado 2 vezes nas páginas 12 e 13.

FERONI, R. C.; ANDREÃO, W. L. Análise do modelo de holt-winters aplicado a uma série histórica de dados com tendência e sazonalidade. In: *Blucher Physics Proceedings*. Editora Blucher, 2017. p. 228–231. Disponível em: <http://www.proceedings.blucher.com.br/article-details/27773>. Citado na página 23.

FERREIRA, P. G. C. et al. *Análise de Séries Temporais em R: curso introdutório*. 1. ed. [S.l.]: Elsevier, 2018. ISBN 978-85-352-9087-5. Citado na página 16.

FIGURES, E. E. I. Statistical pocketbook 2018. *European Union*, 2018. ISSN 2363-247X. Disponível em: <https://publications.europa.eu/en/publication-detail/-/publication/99fc30eb-c06d-11e8-9893-01aa75ed71a1>. Citado 2 vezes nas páginas 13 e 14.

FLANDOLI, F. *ARIMA models*. 2011. Acessado em 12/07/2019. Disponível em: <http://users.dma.unipi.it/~flandoli/AUTCap4.pdf>. Citado na página 21.

GONÇALVES, L. *Características das séries temporais*. 2018. Acessado em 13/04/2019. Disponível em: <http://www.abgconsultoria.com.br/blog/caracteristicas-das-series-temporais/>. Citado 2 vezes nas páginas 16 e 18.

GONTIJO, T. S. et al. Consumo industrial de energia elétrica: um estudo comparativo entre métodos preditivos. *Brazilian Journal of Production Engineering-BJPE*, v. 3, n. 3, p. 31–45, 2017. Citado 3 vezes nas páginas 12, 28 e 30.

HYNDMAN, R. J. Measuring forecast accuracy. p. 9, 2014. Citado na página 24.

HYNDMAN, R. J.; ATHANASOPOULOS, G. *Forecasting: principles and practice*. [S.l.]: OTexts, 2018. Citado na página 19.

IBGE. *Estatísticas históricas do Brasil: séries econômicas, demográficas e sociais de 1550 a 1988*. [S.l.: s.n.], 1990. ISBN 85-240-0333-2. Citado na página 12.

JIANG, S. et al. ARIMA forecasting of china's coal consumption, price and investment by 2030. v. 13, n. 3, p. 190–195, 2018. ISSN 1556-7249, 1556-7257. Disponível em: <https://www.tandfonline.com/doi/full/10.1080/15567249.2017.1423413>. Citado na página 20.

KAYTEZ, F. et al. Forecasting electricity consumption: A comparison of regression analysis, neural networks and least squares support vector machines. *International Journal of Electrical Power & Energy Systems*, v. 67, p. 431–438, May 2015. ISSN 01420615. Citado na página 15.

MINITAB. *Função de autocorrelação (ACF)*. 2019. Acessado em 08/07/2019. Disponível em: <https://support.minitab.com/pt-br/minitab/18/help-and-how-to/modeling-statistics/time-series/how-to/autocorrelation/interpret-the-results/autocorrelation-function-acf/>. Citado na página 17.

PLANNING, I. of B. F. . *Quantitative Forecasting*. 2019. Acessado em 01/06/2019. Disponível em: <https://ibf.org/knowledge/glossary/quantitative-forecasting-222>. Citado na página 19.

SANTANA, E. F. Z. *Introdução ao Apache Spark*. 2016. Acessado em 20/06/2019. Disponível em: <https://www.devmedia.com.br/introducao-ao-apache-spark/34178>. Citado na página 27.

SHORO, A. G.; SOOMRO, T. R. Big data analysis: Ap spark perspective. p. 9, 2015. Citado na página 13.

SILVA, D. A. d.; SANTOS, M. E. d.; COSTA, D. F. A utilização do modelo holt-winters na elaboração de um orçamento de resultado de uma cooperativa de crédito rural. *Revista de Contabilidade do Mestrado em Ciências Contábeis da UERJ*, v. 21, n. 1, 2016. Citado na página 23.

SMARTEN. *What is the Holt-Winters Forecasting Algorithm and How Can it be Used for Enterprise Analysis?* 2018. Acessado em 28/04/2019. Disponível em: <https://www.smarten.com/blog/what-is-the-holt-winters-forecasting-algorithm-and-how-can-it-be-used-for-enterprise-analysis/>. Citado na página 23.

STEPHANIE. *KPSS Test: Definition and Interpretation*. 2016. Acessado em 10/07/2019. Disponível em: <https://www.statisticshowto.datasciencecentral.com/kpss-test/>. Citado na página 19.

TAYLOR, J. W.; MENEZES, L. M. de; McSharry, P. E. A comparison of univariate methods for forecasting electricity demand up to a day ahead. v. 22, n. 1, p. 1–16, 2006. ISSN 0169-2070. Citado 2 vezes nas páginas 28 e 30.

TRATAR, L. F.; STRMčNIK, E. The comparison of holt–winters method and multiple regression method: A case study. v. 109, p. 266–276, 2016. ISSN 03605442. Disponível em: <https://linkinghub.elsevier.com/retrieve/pii/S0360544216305357>. Citado 2 vezes nas páginas 29 e 30.

VEIGA, C. P. D.; TORTATO, U.; SILVA, W. V. D. Demand forecasting in food retail: a comparison between the holt- winters and ARIMA models. v. 11, p. 7, 2014. Citado 3 vezes nas páginas 20, 29 e 30.

WALTER, O. M. F. C. et al. Aplicação de um modelo sarima na previsão de vendas de motocicletas. *Exacta*, Universidade Nove de Julho, v. 11, n. 1, 2013. Citado na página 22.

WOODIE, A. *A Decade Later, Apache Spark Still Going Strong*. 2019. Acessado em 20/06/2019. Disponível em: <https://www.datanami.com/2019/03/08/a-decade-later-apache-spark-still-going-strong/>. Citado na página 27.

A Apêndice - Código-fonte do sistema Re-Cast

Quadro A.1 – Arquivo principal (recast.R)

```
library(sparklyr)
library(dplyr)
library(dbplyr)
library(tidyverse)
library(lubridate)
library(tseries)
library(urca)

## Funcoes ARIMA e Holt-Winters para o sistema
source("/home/***/UFRPE/BSI/TCC/tcc_bsi_ufrpe/software/arima-
    ↪ implementation.R")
source("/home/***/UFRPE/BSI/TCC/tcc_bsi_ufrpe/software/holt-winters-
    ↪ implementation.R")

######################################################################
### Iniciar conexao com o spark
######################################################################

config = spark_config()
config$`sparklyr.backend.threads` <- "4"
config$`sparklyr.shell.driver-memory` <- "4G"
config$`sparklyr.shell.executor-memory` <- "4G"
config$`spark.executor.memoryOverhead` <- "1g"

config$`sparklyr.shell.driver-java-options` <- paste0("-Djava.io.tmpdir="
    ↪ , "/home/***/.tmp")

sc <- sparklyr::spark_connect(master = "local",
                              spark_home = "/opt/spark",
                              config = config)

## Inicia o SparkR
```

```
sparkR.session(sparkHome = "/opt/spark")

######################################################################
### Leitura do CSV original
######################################################################

start_time <- Sys.time()
start_time

# df <- spark_read_csv(sc, "file:///home/***/UFRPE/BSI/TCC/sorted.csv",
    ↪ header = FALSE)
df <- spark_read_csv(sc, "file:///home/***/UFRPE/BSI/TCC/filteredsorted.
    ↪ csv", header = FALSE)

names(df) <- c("id", "ts", "value","work_or_load","plug_id","household_id
    ↪ ","house_id")

end_time <- Sys.time()
end_time

end_time - start_time

######################################################################
### Filtragem por residencia e calculo do consumo horario
######################################################################

start_time_calculo_consumo_por_casa <- Sys.time()
start_time_calculo_consumo_por_casa

## Inicia o SparkR
sparkR.session(sparkHome = "/opt/spark")

for (house in 0:39) {
  df_house <- filter(df, house_id == house)

  ## Filtra o consumo acumulado
  df_house_present_consumption <- df_house %>% filter(work_or_load == 1)

  ## Remove a coluna de ID do CSV original
```

```
  df_house_present_consumption <- select(df_house_present_consumption, -
    ↪ id)

  ## Formata o timestamp em data e hora
  df_house_present_consumption <- df_house_present_consumption %>% mutate
    ↪ (hora = from_unixtime(ts, 'yyyy-MM-dd␣HH'))

  ## Cria uma coluna com a media do consumo de cada plug por comodo
  df_house_hourly_mean_consumption <- df_house_present_consumption %>%
    ↪ group_by(hora, house_id, household_id, plug_id) %>% arrange(hora,
    ↪  household_id, plug_id) %>% summarise(consumo = mean(value))

  ## Soma o consumo de todos os plugs por hora
  df_house_hourly_consumption <- df_house_hourly_mean_consumption %>%
    ↪ group_by(hora, house_id) %>% arrange(hora) %>% summarise(total =
    ↪ sum(consumo))

  ######################################################################
  ### Aplicacao dos metodos de previsao e indices de acuracia
  ######################################################################

  ## Coletar o dataframe inteiro
  df_r <- sparklyr::collect(df_house_hourly_consumption)

  ## Cria a lista para adicionar os dataframes
  list_df <- list()
  list_df[[1]] <- df_r

  ## Aplica os métodos
  spark.lapply(list_df, run_arima_df)
  spark.lapply(list_df, run_hw_df)
}

end_time_calculo_consumo_por_casa <- Sys.time()
end_time_calculo_consumo_por_casa

end_time_calculo_consumo_por_casa - start_time_calculo_consumo_por_casa

######################################################################
```

```
### Consumo da regiao
#####################################################################

start_time_calculo_consumo_all <- Sys.time()
start_time_calculo_consumo_all

## Filtra o consumo acumulado
df_all_present_consumption <- df %>% filter(work_or_load == 1)

## Remove a coluna de ID do CSV original
df_all_present_consumption <- select(df_all_present_consumption, -id)

## Formata o timestamp em data e hora
df_all_present_consumption <- df_all_present_consumption %>% mutate(hora
   ↪ = from_unixtime(ts, 'yyyy-MM-dd␣HH'))

## Cria uma coluna com a media do consumo de cada plug por comodo
df_all_hourly_mean_consumption <- df_all_present_consumption %>% group_by
   ↪ (hora, house_id, household_id, plug_id) %>% arrange(hora, household
   ↪ _id, plug_id) %>% summarise(consumo = mean(value))

## Soma o consumo de todos os plugs por hora
df_all_hourly_consumption <- df_all_hourly_mean_consumption %>% group_by(
   ↪ hora) %>% arrange(hora) %>% summarise(total = sum(consumo))

#####################################################################
### Aplicacao dos metodos de previsao e indices de acuracia
#####################################################################

## Coletar o dataframe inteiro
df_r_all <- sparklyr::collect(df_all_hourly_consumption)

## Cria a lista para adicionar os dataframes
list_df_all <- list()
list_df_all[[1]] <- df_r_all

## Aplica os métodos
spark.lapply(list_df_all, run_arima)
spark.lapply(list_df_all, run_hw)
```

```
## Teste de estacionariedade
stationarity_test <- kpss.test(ts(df_r_all))

end_time_calculo_consumo_all <- Sys.time()
end_time_calculo_consumo_all

end_time_calculo_consumo_all - start_time_calculo_consumo_all
```

Quadro A.2 – Arquivo ARIMA (arima-implementation.R)

```
run_arima_df <- function(df) {
  ## Libs
  require(anytime)
  require(forecast)
  require(urca)
  require(tseries)
  require(MLmetrics)
  require(data.table)
  require(zoo)
  require(normtest)

  FORECAST_WINDOW <- 72

  ## HEADER_NAME PARA O CENÁRIO DE PREVISÃO
  ## POR RESIDÊNCIA
  HEADER_NAME <- paste(c("house:", df$house_id[1]),
                       sep = "␣",
                       collapse = "␣")

  ## CASO SEJA O CONSUMO DA REGIÃO
  ## HEADER_NAME <- "all"

  ## Eliminar outliers
  qnt <- quantile(df$total, probs=c(.25, .75), na.rm = TRUE)
  caps <- quantile(df$total, probs=c(.05, .95), na.rm = TRUE)
  H <- 1.5 * IQR(df$total, na.rm = TRUE)
  df$total[df$total < (qnt[1] - H)] <- caps[1]
  df$total[df$total > (qnt[2] + H)] <- caps[2]
  coluna_diferenca <- df$total
```

```
dados_ts <- ts(coluna_diferenca, frequency=24)

## Arquivo do gráfico
name_graphs <- paste("arima_graph", df$house_id[1],
                     sep = "-",
                     collapse = "␣")
name_pdf <- paste(name_graphs, "pdf", sep = ".")

pdf(name_pdf)

## Preparando o dataframe
dados_ts_na_removed <- na.approx(dados_ts)

## Retirando a parte final do dataframe que compromete o teste
dados_ts_na_removed <- head(dados_ts_na_removed, n = (length(dados_ts_
    ↪ na_removed) - 144))

## Dividindo treino e teste
dados_test <- tail(dados_ts_na_removed, n = FORECAST_WINDOW)

dados_train <- head(dados_ts_na_removed, n = (length(dados_ts_na_
    ↪ removed) - FORECAST_WINDOW))

fit_power <- auto.arima(y = dados_train,
                        stepwise = FALSE,
                        approximation = FALSE,
                        seasonal = TRUE,
                        trace = FALSE,
                        D = 1)

order_arima <- arimaorder(fit_power)

write.table(HEADER_NAME,
            file = "arima_order.txt",
            append = TRUE)
write.table(order_arima,
            file = "arima_order.txt",
            append = TRUE)
```

```
write.table("======",
          file = "arima_order.txt",
          append = TRUE)

arima_forecast <- forecast(object = fit_power,
                        h = FORECAST_WINDOW,
                        level = c(90, 95))

## Previsão
plot(arima_forecast, xlab = "Dias", ylab = "Valores␣reais/previstos␣(Wh
   ↪ )", main = "")
lines(dados_ts_na_removed, lwd = 2, col = 'green')
legend("bottomleft", c("Real", "Previsto"), lwd = c(1, 2),
      col = c("green", "blue"), bty = 'o')

dev.off()

write.table(HEADER_NAME,
          file = "arima_forecast.txt",
          append = TRUE)
write.table(arima_forecast,
          file = "arima_forecast.txt",
          append = TRUE)
write.table("======",
          file = "arima_forecast.txt",
          append = TRUE)

## Métricas
indices <- accuracy(arima_forecast, x = dados_test)

write.table(HEADER_NAME,
          file = "indices_arima.txt",
          append = TRUE)
write.table(indices,
          file = "indices_arima.txt",
          append = TRUE)
write.table("======",
          file = "indices_arima.txt",
          append = TRUE)
```

```
}
```

Quadro A.3 – Arquivo Holt-Winters (holt-winters-implementation.R)

```
run_hw_df <- function(df) {
  ## Libs
  require(forecast)
  require(urca)
  require(tseries)
  require(MLmetrics)
  require(data.table)
  require(zoo)
  require(stats)

  setwd("/home/***/UFRPE/BSI/TCC/")

  FORECAST_WINDOW <- 72

  ## HEADER_NAME PARA O CENÁRIO DE PREVISÃO
  ## POR RESIDÊNCIA
  HEADER_NAME <- paste(c("house:", df$house_id[1]),
                       sep = "␣",
                       collapse = "␣")

  ## CASO SEJA O CONSUMO DA REGIÃO
  ## HEADER_NAME <- "all"

  ## Eliminar outliers
  qnt <- quantile(df$total, probs=c(.25, .75), na.rm = TRUE)
  caps <- quantile(df$total, probs=c(.05, .95), na.rm = TRUE)
  H <- 1.5 * IQR(df$total, na.rm = TRUE)
  df$total[df$total < (qnt[1] - H)] <- caps[1]
  df$total[df$total > (qnt[2] + H)] <- caps[2]
  coluna_diferenca <- df$total


  ## Criando o objeto timeseries
  dados_ts <- ts(coluna_diferenca, frequency=24*7)
```

```
## Arquivo do gráfico
name_graphs <- paste("hw_graph", df$house_id[1],
                     sep = "-",
                     collapse = "␣")

name_pdf <- paste(name_graphs, "pdf", sep = ".")

Acf(dados_ts, lag.max = 20)
Pacf(dados_ts, lag.max = 20)

pdf(name_pdf)

## Tratamento necessário para realizar o forecast
dados_ts_na_removed <- na.approx(dados_ts)

## Retirando a parte final do dataframe que compromete o teste
dados_ts_na_removed <- head(dados_ts_na_removed, n = (length(dados_ts_
    ↪ na_removed) - 144))

## Dataframe de teste
dados_test <- tail(dados_ts_na_removed, n = FORECAST_WINDOW)

## Dataframe de treino
dados_train <- head(dados_ts_na_removed, n = (length(dados_ts_na_
    ↪ removed) - FORECAST_WINDOW))

ajuste_holt <- HoltWinters(dados_train)

plot(fitted(ajuste_holt))

## Coeficientes alpha, beta e gamma
write.table(HEADER_NAME,
            file = "ajuste_holt.txt",
            append = TRUE)

write.table(paste(c("alpha:", ajuste_holt$alpha),
                  sep = "␣",
                  collapse = "␣"),
            file = "ajuste_holt.txt",
```

```
          append = TRUE)

write.table(paste(c("beta:", ajuste_holt$beta),
                  sep = "␣",
                  collapse = "␣"),
          file = "ajuste_holt.txt",
          append = TRUE)

write.table(paste(c("gamma:", ajuste_holt$gamma),
                  sep = "␣",
                  collapse = "␣"),
          file = "ajuste_holt.txt",
          append = TRUE)

write.table("======",
          file = "ajuste_holt.txt",
          append = TRUE)

plot(dados_ts_na_removed, xlab = 'Dias', ylab = 'Valores␣reais/
   ↪ ajustados␣(Wh)', main = '')
lines(fitted(ajuste_holt)[,1], lwd = 2, col = 'red')
legend("topright", c("Consumo", "Ajuste"), lwd = c(1, 2), col = c("
   ↪ black", "red"), bty = 'o')

## Previsao usando o Holt-Winters
holt_forecast <- forecast(ajuste_holt, h = FORECAST_WINDOW, level = 95)

write.table(HEADER_NAME,
          file = "holt_forecast.txt",
          append = TRUE)
write.table(holt_forecast,
          file = "holt_forecast.txt",
          append = TRUE)
write.table("======",
          file = "holt_forecast.txt",
          append = TRUE)

## Plotagem comparando o treino com o teste
plot(holt_forecast, xlab = "Dias", ylab = "Valores␣reais/previstos␣(Wh)
```

```
    ↪ ", main = "")
  lines(dados_ts_na_removed, lwd = 2, col = 'green')
  legend("bottomleft", c("Real", "Previsto"), lwd = c(1, 2),
         col = c("green", "blue"), bty = 'o')

  dev.off()

  ## Calculo dos indices de erro
  indices <- accuracy(holt_forecast,
                      x = dados_test)

  write.table(HEADER_NAME,
              file = "indices_hw.txt",
              append = TRUE)
  write.table(indices,
              file = "indices_hw.txt",
              append = TRUE)
  write.table("======",
              file = "indices_hw.txt",
              append = TRUE)

}
```