



# Instituto Superior de Contabilidade e Administração

Politécnico de Coimbra



**Instituto Superior  
de Contabilidade  
e Administração**

Politécnico de Coimbra

COIMBRA BUSINESS SCHOOL  
ISCAC.pt

Filipe Nuno de Moura Rocha Rigueira

**Automatic Forecast Evaluation Tool**

Coimbra, outubro de 2021





**Instituto Superior  
de Contabilidade  
e Administração**

Politécnico de Coimbra

**COIMBRA BUSINESS SCHOOL**  
ISCAC.pt

Filipe Nuno de Moura Rocha Rigueira

## **Automatic Forecast Evaluation Tool**

Trabalho de projeto submetido ao Instituto Superior de Contabilidade e Administração de Coimbra para cumprimento dos requisitos necessários à obtenção do grau de **Mestre em Análise de Dados e Sistemas de Apoio à Decisão**, realizado sob a orientação da Professora Joana Jorge de Queiroz Leite e do Professor António Rui Trigo Ribeiro.

Coimbra, outubro de 2021

## **TERMO DE RESPONSABILIDADE**

Declaro ser o autor deste projeto, que constitui um trabalho original e inédito, que nunca foi submetido a outra Instituição de ensino superior para obtenção de um grau académico ou outra habilitação. Atesto ainda que todas as citações estão devidamente identificadas e que tenho consciência de que o plágio constitui uma grave falta de ética, que poderá resultar na anulação do presente projeto.

## **AGRADECIMENTOS**

Gostaria de agradecer à minha orientadora, Professora Joana Jorge de Queiroz Leite, pela disponibilidade que permitiu um acompanhamento semanal do projeto, pela sua orientação e coordenação que permitiram que este projeto fosse concluído dentro dos prazos, pelas suas críticas e partilha dos seus conhecimentos que resultaram nesta aplicação.

Gostaria também de agradecer ao meu coorientador, Professor António Rui Trigo Ribeiro, pela partilha dos seus conhecimentos na área da gestão de projetos e desenvolvimentos aplicativos.

Agradeço ainda aos dois a ajuda dada na realização deste relatório.

## RESUMO

As previsões devem ser parte integrante das atividades de tomada de decisão nas organizações modernas. A sua utilização está, no entanto, condicionada pela existência nas empresas de informação fiável e acessível, *know-how* na área das previsões, *software* e recursos económicos para a sua implementação e manutenção. Existem soluções *open-source* geralmente mais baratas do que as soluções comerciais, mas que obrigam a recursos técnicos igualmente dispendiosos e raramente ao alcance das organizações. Este projeto pretende dar resposta a este problema criando uma solução, baseada em *software open-source*, que não obrigue a avultados investimentos e que permita a utilizadores, sem grandes conhecimentos técnicos, efetuarem previsões quantitativas para séries temporais univariadas. Resulta do projeto uma aplicação *Web*, desenvolvida em *Shiny* e suportada por um conjunto de pacotes *software R*, vocacionados especificamente para os processos de previsão. Para além das atividades normais do processo, o carregamento, a transformação e exploração de dados, o treino, a avaliação de modelos, assim como à geração de previsões acrescenta-se ainda a capacidade de identificação de *outliers*, a imputação de valores em falta, a utilização da validação cruzada na avaliação dos modelos e a geração de previsões pontuais e intervalares. Automatiza uma parte significativa das tarefas e apresenta os resultados de uma forma simples e intuitiva.

*Palavras-chave: Séries temporais univariadas; Análise de dados; Validação cruzada; R-Shiny.*

## **ABSTRACT**

Forecasts should be part of modern organizations decision-making. Its dissemination is, however, conditioned by the existence of reliable and accessible information, know-how in forecasting, software, economic resources for its implementation and maintenance. There are open-source solutions that are generally cheaper than commercial ones but require technical resources that are equally expensive and rarely available to organizations. This project aims to solve this problem by creating a solution, based on open-source software, that does not require large investments and that allows users without great technical knowledge to make quantitative forecasts for univariate time series. The goal of the project is to build a Web application, developed in Shiny and supported by a set of R software packages specifically designed for forecasting. In addition to the normal activities of the process, loading, transforming, and exploring data, training, evaluating models, generating forecasts, it also adds outlier detection, missing values imputation, cross-validation for models' evaluation, point and interval forecasts. It automates a significant part of the tasks and presents the results in a simple and intuitive way. It automates a significant part of the tasks and presents the results in a simple and intuitive way.

*Keywords: Univariate Time series; Cross-validation; Data Analysis; R-Shiny*

## ÍNDICE GERAL

1	INTRODUÇÃO .....	1
1.1	Motivação .....	1
1.2	Objetivo .....	2
1.3	Organização .....	3
2	REVISÃO DE LITERATURA .....	5
2.1	Séries temporais univariadas e suas componentes.....	5
2.2	Processo de previsão .....	7
2.3	Preparação de dados.....	8
2.3.1	<i>Outliers</i> .....	8
2.3.2	Falhas de dados.....	9
2.4	Métodos de previsão de séries temporais .....	10
2.4.1	Métodos elementares .....	10
2.4.2	Métodos tradicionais .....	10
2.4.3	Métodos não tradicionais.....	11
2.5	Ambiente de teste.....	11
2.5.1	Hold-out.....	12
2.5.2	Validação cruzada.....	12
2.5.3	Métricas para avaliação da capacidade preditiva .....	15
2.6	Software .....	17
2.6.1	R .....	17
2.6.2	Shiny.....	18



---

2.6.3	Pacotes para previsão com funcionalidades múltiplas .....	19
3	METODOLOGIA .....	24
4	DESENVOLVIMENTO DA APLICAÇÃO.....	28
4.1	Análise de requisitos .....	28
4.1.1	1ª Etapa – Carregar dados (1. <i>Data Load</i> ).....	28
4.1.2	2ª Etapa – Transformar Dados (2. <i>Data Preparation</i> ).....	29
4.1.3	3ª Etapa – Explorar Dados (3. <i>Data Exploration</i> ).....	29
4.1.4	4ª Etapa – Treinar Modelos (4. <i>Modeling</i> ) .....	30
4.1.5	5ª Etapa – Gerar Previsões (5. <i>Forecasting</i> ).....	31
4.2	Tecnologias selecionadas.....	31
4.3	Ciclos de desenvolvimento .....	32
4.3.1	Iteração 1 .....	32
4.3.2	Iteração 2 .....	34
4.3.3	Iteração 3 .....	37
5	APLICAÇÃO .....	41
5.1	Apresentação da aplicação.....	41
5.1.1	<i>Data Load</i> - Carregamento de dados.....	43
5.1.2	<i>Data Preparation</i> - Transformação de dados .....	44
5.1.3	<i>Data Exploration</i> – Exploração de Dados.....	45
5.1.4	<i>Decomposition</i> – Decomposição STL .....	45
5.1.5	<i>Cross Validation</i> – Treinar Modelos .....	46
5.1.6	<i>Forecasting</i> – Gerar Previsões .....	47

5.2	Demonstração .....	49
5.2.1	Imputação de dados .....	49
5.2.2	Validações cruzadas .....	54
5.2.3	Avaliação de modelos e geração de previsões.....	59
6	CONCLUSÃO .....	68
	REFERÊNCIAS .....	69
7	APÊNDICES .....	73
7.1	Código R - Missing Data (secção 5.2.1.1).....	73
7.2	Imputação de dados – 20% (secção 5.2.1.2).....	74

## ÍNDICE DE FIGURAS

Figura 2.1. Decomposição.....	6
Figura 2.2. Etapas processo de previsão.....	7
Figura 2.3. Prequential Blocks. ....	14
Figura 2.4. Prequential Sliding Blocks. ....	15
Figura 3.1. Metodologia Desenvolvimento Rápido de Aplicação.....	26
Figura 4.1. Workflow da aplicação .....	28
Figura 4.2. Pacotes R usados na aplicação.....	32
Figura 5.1. Aplicação – Ecrã inicial .....	41
Figura 5.2. Inputs da aplicação.....	42
Figura 5.3. Outputs da aplicação .....	42
Figura 5.4. Data Load - Carregamento de Dados .....	43
Figura 5.5. Data Preparation - Transformação do Dados .....	44
Figura 5.6. Data Exploration - Exploração de Dados.....	45
Figura 5.7. Decomposition - Decomposição STL .....	45
Figura 5.8. Cross Validation - Treinar Modelos (1).....	46
Figura 5.9. Cross Validation -Treinar Modelos (2).....	47
Figura 5.10. Forecasting - Gerar Previsões Pontuais .....	47
Figura 5.11. Forecasting - Gerar Previsões Intervalares .....	48
Figura 5.12. Série – Retiradas 20% das observações .....	50
Figura 5.13. Imputação - IQR factor 3 - Modelos M01/M05/M06/M07/M09.....	53
Figura 5.14. Imputação - IQR factor 3 - Modelos M04 .....	53

---

Figura 5.15. Imputação - IQR factor 3 - Modelos M08 .....	53
Figura 5.16. CV - OSS – Configuração.....	55
Figura 5.17. CV - Fixed Window – Configuração .....	56
Figura 5.18. CV - Rolling Window – Configuração .....	57
Figura 5.19. CV - Submenu - Error analysis .....	57
Figura 5.20. CV - Submenu - Error analysis Plots .....	58
Figura 5.21. CV - Submenu - Error Plots .....	59
Figura 5.22. Data Load .....	60
Figura 5.23. Data Preparation.....	60
Figura 5.24. Sazonalidade – Periodicidade Mensal.....	61
Figura 5.25. Sazonalidade - Periodicidade Trimestral .....	61
Figura 5.26. Subséries Sazonais .....	61
Figura 5.27. Evolução Anual.....	62
Figura 5.28. Decomposição da série.....	62
Figura 5.29. Validação cruzada – Configuração .....	63
Figura 5.30. Validação cruzada - Resumo Modelos.....	63
Figura 5.31. Validação cruzada - Avaliação dos Modelos .....	64
Figura 5.32. AirPassenger - Previsão Pontual – Global .....	64
Figura 5.33. Previsão Pontual - Melhores Modelos .....	65
Figura 5.34. Previsão Intervalar - Auto Arima.....	65
Figura 5.35. Previsão Intervalar - ETS MAM.....	66
Figura 5.36. Previsão Intervalar - ETS ZZZ.....	66

Figura 5.37. Previsão Intervalar - Holt Winters .....	67
Figura 5.38. Previsão Intervalar – Snaive .....	67

## ÍNDICE DE TABELAS

Tabela 2.1. Métricas de avaliação .....	16
Tabela 2.2. Pacotes R – Resumo .....	22
Tabela 2.3. Pacotes R – Âmbito .....	22
Tabela 4.1 - Evolução das funcionalidades entre versões .....	40
Tabela 5.1 - Resultados – Métricas MAE+MAPE .....	50
Tabela 5.2- Resultados – IQR Factor(1,2,3).....	51
Tabela 7.1 - Imputação20% - Valores Imputados .....	74
Tabela 7.2 - Imputação20% - Erros de imputação .....	75
Tabela 7.3 - Imputação20% - Erro Absoluto.....	76
Tabela 7.4 - Imputação20% - Percentagem Erro Absoluto .....	77

## **Lista de abreviaturas, acrónimos e siglas**

ACF - Autocorrelação

ARIMA - Processo autorregressivo e de médias móveis integrado

BATS - Box-Cox transform, ARMA errors, Trend, and Seasonal components

CSS - Cascading Style Sheets

CSV – Comma-separated values

CV - Validação cruzada

HTML - HyperText Markup Language

ETS - Error-Trend-Seasonal

IQR - Amplitude interquartil

MAE - Erro absoluto médio

MAPE - Percentagem do erro absoluto médio

MAR - Falha aleatória de dados

MCAR - Falha completamente aleatória de dados

ME - Erro médio

MNAR - Falha condicionada de dados

MPE – Erro percentual médio

MSE - Erro quadrático médio

NNAR – neural network autoregression

PACF - partial autocorrelation function

RAD - Desenvolvimento Rápido de Aplicação

RW - Random Walk width drift

RMSE - Raiz do erro quadrático médio

SARIMA - ARIMA Sazonal

STL - Seasonal and Trend decomposition using Loess



# 1 INTRODUÇÃO

## 1.1 Motivação

O tema das previsões é um assunto recorrente dentro das organizações. São necessárias previsões para tomar boas decisões e são utilizadas em áreas como: gestão operacional, prevendo vendas e utilizando essa informação na programação da produção, no controlo de stocks, na gestão da cadeia de abastecimento e na estimativa de necessidades de recursos; o marketing, utilizando a previsão de vendas para planear campanhas e definir políticas de preços; a gestão de risco, para prever o retorno dos investimentos; a financeira, para prever as principais variáveis económicas, a fim de apoiar as decisões de planeamento estratégico (Montgomery et al. 2015).

Segundo Hyndman e Athanasopoulos (2021, capítulo 1), o entendimento dos fatores que contribuem para um evento, a quantidade e qualidade dos dados disponíveis e a possibilidade de as previsões poderem afetar a realidade, condicionam o processo de previsão. São estes aspetos que vão determinar a metodologia de previsão, concretamente se esta deve ser qualitativa ou quantitativa, como se detalha abaixo, seguindo de perto estes autores.

A previsão qualitativa resulta da opinião de especialistas e é muito comum nas organizações e, na ausência de dados históricos, esta poderá ser mesmo a única solução. No entanto, é importante reconhecer, que resultam de avaliações subjetivas potencialmente condicionadas por preconceitos e fatores psicológicos. Estas limitações podem, no entanto, ser minimizadas pela implementação de abordagens bem estruturadas e sistemáticas que permitam melhorias na sua qualidade.

A previsão quantitativa, como refere Montgomery et al. (2015), faz uso de modelos de previsão que, com base em dados históricos, identificam padrões e relações estatísticas, para projetar os padrões de dados no futuro. A existência de padrões consistentes, tendências significativas, sazonalidades relevantes, evidência da presença de ciclos de negócios, existência de dados discrepantes e a intensidade das relações entre as variáveis disponíveis, são alguns dos fatores que podem condicionar a qualidade das previsões.

Segundo Hyndman e Athanasopoulos (2021, capítulo 1), boas previsões capturam os

padrões e relacionamentos genuínos que tenham ocorrido no passado e não replicam eventos passados que não ocorrerão novamente. No entanto, como referem Taylor e Letham (2018), a produção de previsões de alta qualidade não é algo facilmente atingível, quer seja por máquinas ou especialistas. Segundo estes autores, os processos de previsão são difíceis de afinar e os analistas experientes e especializados são difíceis de encontrar. É, porém, importante referir que a utilização de algoritmos matemáticos complexos não é, por si só, uma garantia de qualidade. A disseminação das previsões nas organizações está, portanto, condicionada por fatores económicos, pela necessidade de *software* específico, e por fatores humanos que resultam da necessidade de conhecimento técnico específico. No entanto, como referem Svetunkov e Yusupova (2021), existem inúmeras ferramentas *open-source*<sup>1</sup> que, apesar de exigirem conhecimentos especializados, permitem a implementação de modelos de previsões a custos mais reduzidos. De entre as várias linguagens de programação *open-source*, o R é, atualmente, uma referência na área das previsões em séries temporais (Fildes et al. 2020). Trata-se de uma linguagem que se baseia na utilização de funcionalidades disponibilizadas em pacotes temáticos e que tem um conjunto alargado de funcionalidades dedicadas ao processo de previsão. Apesar de existirem funcionalidades que cobrem todas as etapas do processo de previsão, nenhuma o faz de forma integrada, o que permitiria uma utilização mais fácil e interativa.

## 1.2 Objetivo

Este projeto visa criar uma aplicação que permita, a utilizadores sem proficiência na linguagem R, efetuarem previsões com base em séries temporais univariadas<sup>2</sup>. Para além

---

<sup>1</sup> *Open source* é a abreviatura do termo original *software open source* (OSS) e que classifica o código de software que é desenvolvido para ser utilizado de forma livre e sem custos. O software open source é desenvolvido de forma descentralizada e colaborativa. Costuma ser mais barato, mais flexível e mais duradouro do que as opções proprietárias já que é desenvolvido por comunidades de especialistas. Este tipo de código pode ser visto, analisado, modificado e distribuído conforme as necessidades de cada um. (<https://www.redhat.com/pt-br/topics/open-source/what-is-open-source>)

<sup>2</sup> Uma série temporal univariada é um conjunto de observações únicas registadas em pontos sucessivos no tempo (Moritz et al. 2015).

da capacidade de geração de previsões a aplicação deverá também disponibilizar funcionalidades que permitam efetuar avaliações comparativas entre diferentes modelos de previsão. Sendo uma aplicação destinada a utilizadores não especialistas, esta deverá: automatizar uma parte significativa das tarefas, apresentar os resultados de uma forma simples e intuitiva, ajudar na identificação dos modelos que melhor se ajustam aos dados em análise. Em resumo a aplicação deverá disponibilizar o seguinte conjunto de funcionalidades:

- Seleção e carregamento dados,
- Análise da qualidade dos dados,
- Tratamento dos dados,
- Análise exploratória de dados,
- Seleção de métricas de avaliação de modelos,
- Avaliação comparativa de resultados,
- Previsões pontuais e intervalares.

### **1.3 Organização**

No capítulo 2 é feita uma revisão de literatura dos temas: séries temporais univariadas, processo de previsão, identificação de *outliers*, imputação de dados, classificação e descrição dos métodos de previsão, metodologias de treino e avaliação de modelos e métricas de avaliação de erros de previsão. São também descritas as potencialidades base da linguagem R na área das previsões e das capacidades interativas disponibilizadas pelo Shiny. No final do capítulo é feita uma avaliação de diferentes pacotes R recentemente criados no âmbito do projeto.

No capítulo 3 é descrita a metodologia de desenvolvimento de projetos informáticos utilizada.

No capítulo 4 são descritos os requisitos funcionais da aplicação, as tecnologias utilizadas e o ciclo de desenvolvimento do projeto com a descrição da evolução funcional de cada protótipo criado.

No capítulo 5 é feita uma apresentação da aplicação descrevendo as respetivas funcionalidades, é feita uma demonstração pratica da imputação de dados, uma demonstração da validação cruzada na avaliação de modelos e uma demonstração detalhada de um exemplo em que é utilizada a aplicação na identificação dos melhores modelos para a geração de previsões pontuais e intervalares para uma série temporal.

As conclusões do projeto são apresentadas no capítulo 6.

## 2 REVISÃO DE LITERATURA

Este capítulo está organizado em dois subtemas principais. O primeiro, que compreende as secções 2.1 a 2.5, está relacionado com conceitos base e o processo de previsão de séries temporais, concretamente, séries temporais univariadas, *outliers*, imputação de valores em falta, métodos de previsão de séries temporais, validação cruzada, métricas de avaliação de modelos de previsão. O segundo, com a secção 2.6, trata das questões diretamente ligadas ao *software*, nomeadamente, a avaliação do Shiny e de alguns dos pacotes R recentes criados na área de previsão em séries temporais.

### 2.1 Séries temporais univariadas e suas componentes

Define-se como série temporal todo o conjunto de observações, associadas a determinado fenómeno aleatório, efetuadas em períodos de tempo sucessivos (Brockwell & Davis, 2002). A sua análise passa pela utilização de abordagens sistemáticas que permitem, não só a análise sumária dos dados, como também a compreensão da forma como variam ao longo do seu período temporal. A análise sumária descreve as características quantitativas da série sem que se tenha em consideração a componente do tempo, no entanto, para compreender a forma como variam implica que a avaliação seja feita tendo em atenção a sequencialidade dos momentos em que ocorrem.

As séries temporais podem exibir uma variedade de padrões, que podem ser separados, por decomposição (Dagum & Bianconcini, 2016, capítulo 2), nas seguintes componentes:

- Ciclo – Define um padrão de flutuações recorrentes que ocorrem obrigatoriamente com periodicidade longa e não necessariamente fixa,
- Tendência – Define um padrão que se verifica durante um longo período de tempo. Corresponde à forma como a série evolui a longo prazo e normalmente resulta numa linha, não necessariamente linear, de subida ou descida,
- Sazonalidade – Define um padrão ou padrões de variação que ocorrem obrigatoriamente com periodicidade curta e que se repetem consecutivamente no mesmo período temporal,

- Resíduos ou componente irregular – Identifica a componente aleatória e corresponde aos valores que não podem ser explicados pelas restantes componentes. Resulta normalmente numa variação irregular quer seja em termos de amplitude quer seja em termos de periodicidade. Idealmente, deverá ser totalmente aleatória e a sua distribuição deverá ser normal.

Existem vários métodos de decomposição: o clássico, os usados pelas agências de estatísticas oficiais (X11, X11/X12-ARIMA, TRAMO-SEATS, X13-ARIMA-SEATS, analisados em Dagum e Bianconcini, 2016) e o *Seasonal and Trend decomposition using Loess* (STL). Este último foi proposto por Cleveland et al. (1990) e, de acordo com Hyndman e Athanasopoulos (2021, capítulo 3), apresenta vantagens interessantes relativamente aos restantes, tais como, a sua versatilidade, a sua robustez a decompor séries temporais, a capacidade de tratar qualquer tipo de sazonalidade, o permitir controlar a taxa de mudança da variação da componente sazonal ao longo do tempo, ao permitir controlar a suavidade do ciclo de tendência, ao poder ser configurado de forma a que os *outliers* não afetem as estimativas do ciclo de tendência e componentes sazonais. A Figura 2.1 ilustra a decomposição STL de uma série de dados mensais que exhibe três das quatro componentes referidas.

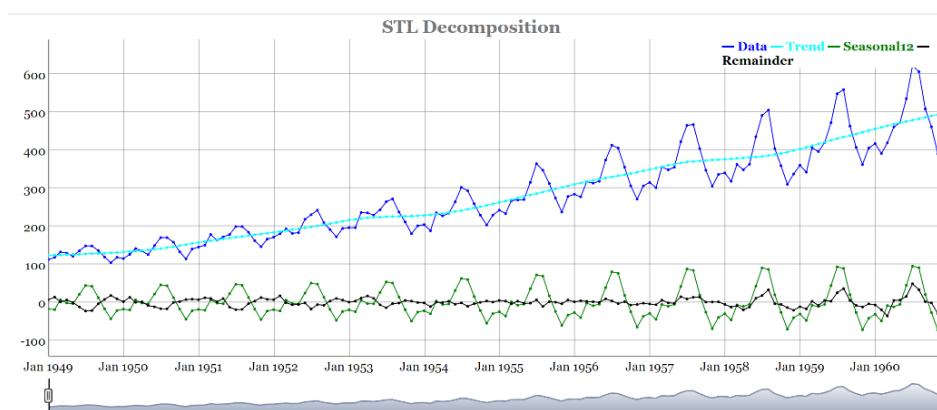


Figura 2.1. Decomposição: série original (Data, azul-escuro) e as três componentes obtidas por decomposição STL, tendência (Trend, azul-claro), sazonalidade (Seasonal, verde) e resíduos (Remainder, preto).

## 2.2 Processo de previsão

O processo de previsão é composto por uma série de atividades conectadas que transformam uma ou mais entradas em uma ou mais saídas (Montgomery et al. 2015). As etapas devem ser percorridas sequencialmente permitindo, no entanto, e sempre que necessário, repetir tarefas anteriores, tal como se mostra na Figura 2.2.

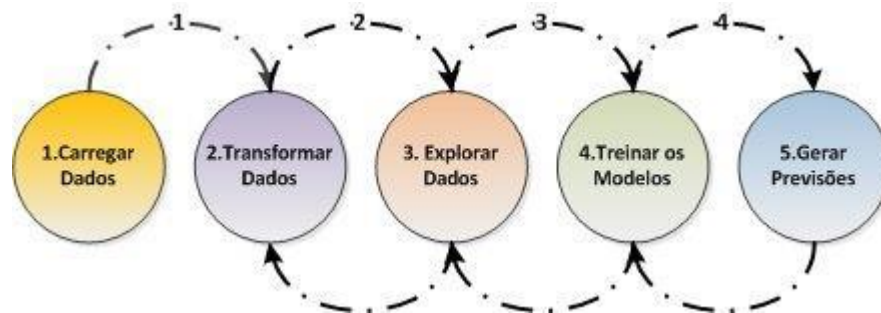


Figura 2.2. Etapas processo de previsão

1ª Etapa – Carregar dados. A série de dados é carregada e deve ser efetuada uma pré-avaliação do seu conteúdo. Esta etapa corresponde à primeira abordagem aos dados e o objetivo principal é confirmar que estes têm a quantidade e qualidade necessárias para dar seguimento ao processo. Nesta etapa deve ser identificada a necessidade ou não de transformações.

2ª Etapa – Transformar dados. Nesta fase devem ser efetuadas todas as transformações aos dados necessárias para permitir que estes possam ser utilizados pelos modelos de previsão. Esta etapa apenas será executada caso sejam identificadas necessidades de transformação de dados.

3ª Etapa – Explorar dados. Nesta fase é feita uma exploração dos dados de forma a identificar as suas características mais relevantes. Desta etapa poderá resultar a necessidade de novas transformações fazendo com que o processo seja retornado à 2ª etapa.

4ª Etapa – Treinar os modelos. Nesta fase os diferentes modelos são treinados e testados. Os resultados obtidos são utilizados para avaliar os modelos e identificar e selecionar aqueles que melhor se ajustam à série e aos objetivos pretendidos. Desta etapa poderá

resultar a necessidade de novas transformações fazendo com que o processo seja retornado à 2ª etapa.

5ª Etapa – Gerar previsões. Nesta etapa utilizam-se o(s) melhor(es) método(s) para gerar previsões que podem ser do tipo pontual ou intervalar.

## 2.3 Preparação de dados

A qualidade dos dados é fundamental para o processo de previsão pelo que qualquer análise deverá ser antecedida por uma etapa em que se identifiquem e corrijam as anomalias existentes.

### 2.3.1 *Outliers*

Os valores discrepantes denominam-se *outliers* e correspondem a valores que se desviam da norma o suficiente para serem considerados exceções (Reunanen et al. 2020). A sua identificação é feita com base na suposição de que esses valores discrepantes, ou exceções, se destacam do restante conjunto de dados. A classificação de um *outlier* é um desafio, porque, na maioria dos casos, a sua definição precisa é bastante ambígua. Segundo Volodymyr (2021), a resolução dessa ambiguidade passa pela quantificação da sua intensidade.

A identificação de *outliers* em séries temporais começa pela sua decomposição de forma a identificar os seus de padrões e, de seguida, valor a valor, identificar aqueles que se afastam significativamente do padrão esperado. Por exemplo, Vallis *et al.* (2014) sugerem a utilização de uma decomposição STL para que, com base em apenas na componente residual, se realize a identificação dos *outliers*. Tal como referem Hyndman e Athanasopoulos (2021, capítulo 13), nessa identificação pode ser utilizada uma metodologia baseada na amplitude interquartil (IQR, i.e., a diferença entre o 3.º quartil (Q3) e o 1.º quartil (Q1)), que identifica os intervalos de valores que estabelecem a separação entre os valores considerados “normais” dos *outliers* de acordo com os seguintes limites:

- Limite Inferior =  $Q1 - IQR * IQR\ factor$
- Limite Superior =  $Q3 + IQR * IQR\ factor$



Um *outlier* corresponderá a todo e qualquer valor que se posicione fora destes limites. O parâmetro *IQR factor* é um fator que permite ajustar a sensibilidade do método, sendo, por omissão, igual a 3 (i.e., o valor habitual para a deteção dos *outliers* severos). Quanto menor for o valor de *IQR factor*, maior será a sensibilidade do método, aumentando a probabilidade de identificar *outliers*.

### 2.3.2 Falhas de dados

No caso específico das séries temporais, as falhas de dados são um problema grave porque quebram a sua sequencialidade temporal, o que, segundo Moritz et al. (2015), condiciona a performance dos modelos de previsão.

De uma forma geral, a resolução deste problema passa por uma das seguintes três opções: a eliminação da variável onde existem falhas; eliminação dos registos onde ocorrem as falhas; a substituição das falhas por valores que possam ser considerados válidos. No caso específico de uma série temporal univariada, resulta que a primeira opção não é possível, por apenas existir uma variável, e a segunda ser dificilmente aplicável, por provocar a sua descontinuidade temporal. A solução nestes casos passa pela sua substituição por um processo denominado de imputação e que corresponde a um processo de correção, baseado numa estimativa, que os substitui por valores prováveis calculados com base nas informações disponíveis (Montgomery et al. 2015). Ainda segundo Moritz et al. (2015), a substituição de valores em falta por valores razoáveis passa necessariamente pela utilização de metodologias que tenham em consideração a componente temporal, as características da série (tendência, sazonalidade, irregularidade, autocorrelação) assim como o tipo de aleatoriedade das falhas (MCAR<sup>3</sup>, MAR<sup>4</sup>, MNAR<sup>5</sup>)

---

<sup>3</sup> MCAR - Falha de informação completamente ao acaso. Não existe nenhuma relação entre os dados em falta e os restantes dados e não há um padrão para os valores reais dos próprios dados ausentes.

<sup>4</sup> MAR - Falha aleatória de dados. Existe uma relação entre as falhas e as outras variáveis no conjunto de dados. No entanto, os valores reais que faltam são aleatórios.

<sup>5</sup> MNAR - Falha de dados não é aleatória. O padrão de falta está relacionado a outras variáveis no conjunto de dados, mas, além disso, os valores dos dados perdidos não são aleatórios.

## 2.4 Métodos de previsão de séries temporais

### 2.4.1 Métodos elementares

Correspondem a um conjunto de métodos básicos que normalmente são utilizados como referência para os restantes métodos. Destes os mais utilizados são o Naïve que corresponde a assumir como previsão o último valor conhecido, o Naïve Sazonal, utilizado para dados onde existe sazonalidade e em que se assume como previsão o valor observado no mesmo período anterior, o *Random Walk with drift (RW)* que é equivalente a traçar uma linha entre a primeira e a última observação e extrapolar para o futuro (Hyndman & Athanasopoulos, 2021, capítulo 5).

### 2.4.2 Métodos tradicionais

Os métodos de alisamento exponencial e os métodos *AutoRegressive Integrated Moving Average (ARIMA)* são as duas abordagens mais utilizadas nas previsões em séries temporais. Os modelos de alisamento exponencial baseiam-se numa interpretação da tendência e da sazonalidade nos dados, enquanto que os modelos ARIMA visam descrever as autocorrelações nos dados (Hyndman & Athanasopoulos, 2021, capítulo 8).

Os métodos baseados no alisamento exponencial utilizam médias ponderadas das observações anteriores em que o peso atribuído a cada observação vai diminuindo exponencialmente ao longo do tempo. A sua popularidade pode ser atribuída à sua simplicidade e à sua razoável precisão (Ostertagova & Ostertag, 2011). Os mais simples são: o alisamento exponencial simples para séries sem tendência nem sazonalidade, o método de Holt (Holt, 2004) para séries com tendência mas sem sazonalidade. O método Holt inclui mais um parâmetro de alisamento do que o alisamento exponencial simples e que corresponde à informação da tendência (Ferber Tratar & Strmčnik, 2016). O método de Holt-Winters por sua vez acrescenta ao método Hold a componente da sazonalidade. Atualmente, estes métodos recebem a designação genérica de *Error-Trend-Seasonal (ETS)*, acrónimo que pode também ser lido como *Exponential Smoothing*. Permitindo vários tipos de tendência, de sazonalidade e dois tipos de erro estes métodos têm ainda algumas variantes como sejam: o Theta que corresponde a um modelo de alisamento exponencial simples com *drift* em que o respetivo parâmetro de *drift* corresponde a

metade da inclinação da tendência linear ajustada aos dados (Hyndman & Billah, 2003), o BATS corresponde a um modelo em que é utilizada uma transformação Box-Cox o que lhe permite uma maior eficácia em séries em que existam múltiplas sazonalidades (de Livera et al. 2011).

O método ARIMA, por sua vez, corresponde a um modelo em que é feita uma diferenciação da série e de seguida se conjuga um modelo auto regressivo com um modelo de média móvel (Makridakis & Hibon, 1997). A classe dos ARIMA inclui ainda os modelos ARIMA Sazonais (SARIMA) capazes de modelar séries com sazonalidade.

Para além das metodologias anteriores existe um conjunto, denominado por modelos estruturais, que utilizam as componentes mais relevantes das séries e podem ser facilmente estendidos para lidar com qualquer frequência de observação (semanal, diária, horária) e características específicas das séries que são difíceis de lidar pelos métodos ARIMA como sejam a heterocedasticidade, não linearidade, não gaussianidade (Proietti, 1991). Segundo (Harvey & Koopman, 1996) são baseados na decomposição da série e nada mais são do que modelos de regressão nos quais as variáveis explicativas são funções do tempo e os parâmetros variam com o tempo.

### 2.4.3 Métodos não tradicionais

Nesta classificação cabem os métodos recentes, alguns deles baseados em *machine learning* e outros que resultam da junção de diferentes métodos. São exemplo destes modelos o NNAR que utiliza uma rede neural do tipo *feed-forward* em conjugação com um modelo de autoregressão linear (Hyndman & Athanasopoulos, 2021, capítulo 11) e o modelo Prophet que corresponde a um modelo aditivo em que tendências não lineares são ajustadas de acordo com as sazonalidades existentes e que permite incorporar os efeitos de datas especiais como sejam, por exemplo, os feriados (Taylor & Letham, 2018).

## 2.5 Ambiente de teste

Segundo Hyndman e Athanasopoulos (2021, capítulo 5), o desempenho de um modelo só pode ser avaliado adequadamente confrontando as suas previsões com dados que não foram usados no seu próprio treino. Song et al. (2021), por sua vez, referem que a

“generabilidade de um modelo”<sup>6</sup> descreve o quão bem os resultados da sua calibração, com base numa amostra, são aplicáveis a outras amostras da população. Segundo estes autores, as calibrações dos modelos com base numa única amostra tendem a sofrer do problema de sobre ajustamento. O efeito contrário, segundo os mesmos autores, também poderá ocorrer quer seja pela variação da amostragem, quer seja pela complexidade do modelo. A solução que propõem passa pela utilização de várias amostras de calibração.

Segundo Bergmeir et al. (2018) a validação cruzada *K-fold* é um dos procedimentos mais utilizados na avaliação de modelo de previsão. No entanto, referem estes autores, que quando se trata de séries temporais, por causa da autocorrelação inerente e pelo potencial de não estacionaridade dos dados, a sua aplicabilidade não é direta e é muitas vezes substituída pela metodologia *hold-out*.

### 2.5.1 Hold-out

Segundo Cerqueira et al. (2020) esta metodologia é utilizada em situações em que os dados têm algum grau de dependência temporal como é o caso das séries temporais. A metodologia separa a série em dois blocos, o primeiro bloco utilizado para treinar o modelo e o segundo bloco utilizado para testar a qualidade das previsões. De acordo com os autores, esta metodologia, no caso das séries temporais, deve preservar a ordem temporal das observações, permitindo assim lidar com as correlações temporais existentes entre observações sucessivas.

### 2.5.2 Validação cruzada

Para Song et al. (2021) a essência da validação cruzada passa por, a partir de um único conjunto de dados, gerar diferentes conjuntos de treino e teste e, a partir destes, treinar e testar repetidamente o modelo. Esta metodologia resulta num método versátil para a calibração dos modelos e para a avaliação da sua generalização. Os autores repartem o processo em cinco etapas:

---

<sup>6</sup> Os autores usam a expressão “Model generalizability describes how well the findings from a sample are applicable to other samples in the population.”.

- 1) Dividir o conjunto de dados num conjunto de treino e num conjunto de teste (de acordo com o método de validação cruzada escolhido),
- 2) Treinar o modelo com os dados do conjunto de treino,
- 3) Avaliar a precisão da previsão do modelo ajustado com base nos dados de teste,
- 4) Repetir as etapas 1 a 3,
- 5) Agregar todos os resultados de precisão de previsão da Etapa 4.

Como referem Cerqueira et al. (2020) a validação cruzada permite a otimização da utilização dos dados, sendo que, para séries temporais, estes autores propõem duas abordagens distintas de repartição dos dados:

- 1) *Prequential Blocks*<sup>7</sup> em que os dados são repartidos em séries de blocos de dimensão crescente, como se ilustra na Figura 2.3;

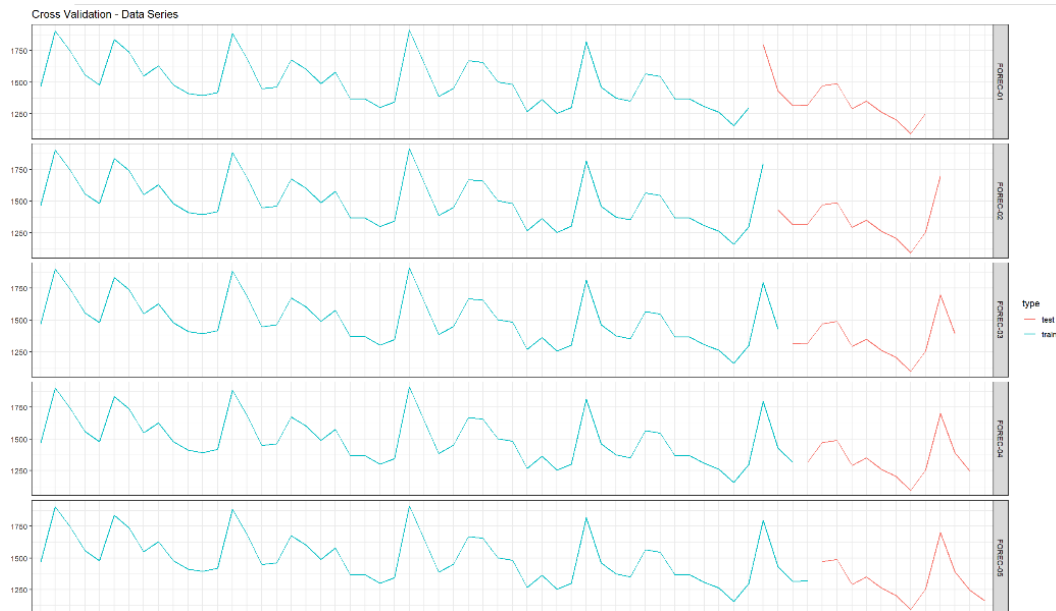


Figura 2.3. *Prequential Blocks* – A cada gráfico corresponde uma série da validação cruzada, sendo que a azul são representas as observações usadas no treino (Train) e a vermelho as usadas no teste (Test).

- 2) *Prequential Sliding Blocks*<sup>8</sup> em que as séries têm blocos de dimensão fixa, mas que deslizam ao longo da série, como se exemplifica na Figura 2.4.

---

<sup>7</sup> *Prequential Blocks* - Fixa apenas o tamanho do bloco de teste e a posição inicial do bloco de treino. Em cada nova série acrescenta uma nova observação ao bloco de treino, o que provoca arrastamento correspondente do bloco de teste. A criação das séries termina assim que a série seguinte já não possua dados suficientes para completar ambos os blocos.

<sup>8</sup> *Prequential Sliding Blocks* - Utiliza blocos de treino e de teste de dimensões fixas, em que as séries diferem entre si por terem posições iniciais obrigatoriamente diferentes. Tipicamente, a primeira série começa na posição inicial da série original e as seguintes vão avançando, até que a série seguinte já não possua dados suficientes para completar ambos os blocos.

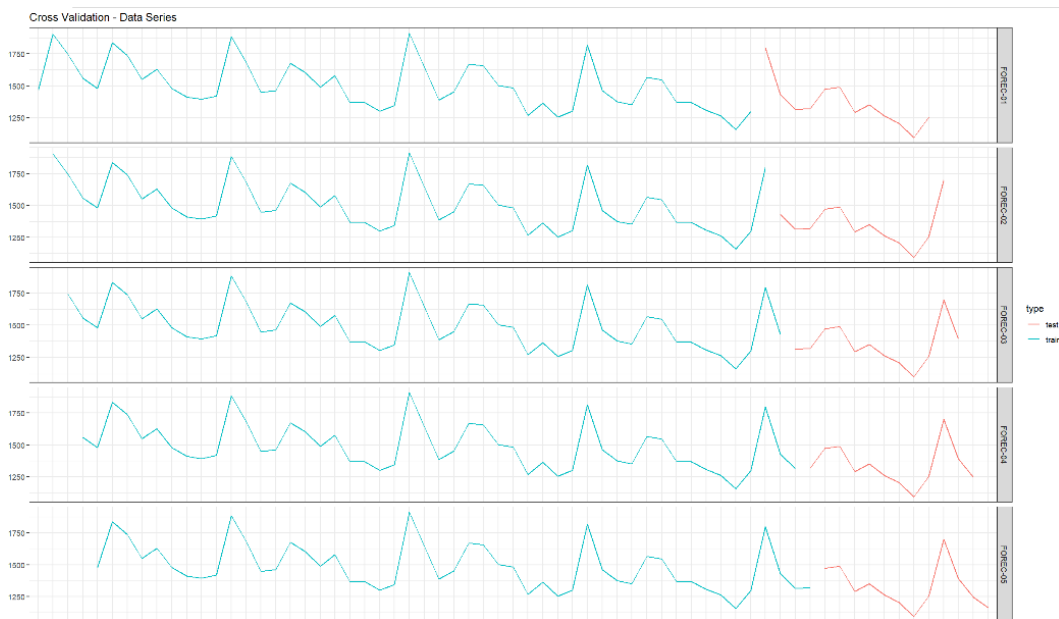


Figura 2.4. Prequential Sliding Blocks – A cada gráfico corresponde uma série da validação cruzada, sendo que a azul são representas as observações usadas no treino (Train) e a vermelho as usadas no teste (Test).

Tanto num caso como no outro, as séries têm que ter obrigatoriamente uma dimensão que, por um lado garanta as condições de treino e de teste, mas por outro, tenham que ser suficientemente pequenas para maximizar o número de séries possíveis. Estas duas abordagens propostas pelos autores têm a vantagem de garantir a sequencialidade dos dados, algo que o *standard* da validação cruzada, habitualmente usada com dados seccionais, não garante, por funcionar numa lógica de circularidade.

É importante referir que Molinaro et al. (2005), citado em Song et al. (2021) afirmam que, aumentar o número de repetições, no caso da validação cruzada resulta no aumento da estabilidade das estimativas (ou seja, diminui a variância), sem aumentar o seu viés.

### 2.5.3 Métricas para avaliação da capacidade preditiva

A avaliação da capacidade preditiva dos modelos, feita no(s) conjunto(s) de teste, passa pela identificação dos respetivos erros de previsão. Esta avaliação obriga a simular previsões para períodos em que os valores reais são conhecidos, o que permite obter os erros de previsão, já que estes resultam da diferença entre os valores observados e os

valores previstos, independentemente no modelo de previsão (Hyndman & Koehler, 2005).

Koutsandreas et al. (2021) referem que continua a existir muita controvérsia em torno da escolha da medida de erro mais apropriada na avaliação do desempenho dos métodos de previsão. Segundo estes autores, os estatísticos defendem o uso de medidas com boas propriedades estatísticas e os profissionais preferem medidas que sejam fáceis de comunicar e compreender. Como referem Mehdiyev et al. (2016), diferentes métricas de avaliação têm sido adotadas em diferentes estudos e os resultados sugerem não existir uma métrica que universalmente possa ser identificada como sendo a “melhor”. Para Chen et al. (2017) uma boa medida de exatidão deve fornecer um resumo informativo e claro da distribuição do erro e, na sua escolha, devem ser considerados critérios como: a sua confiabilidade, complexidade computacional, tratamento de *outliers*, independência de escala, sensibilidade a mudanças e interpretabilidade. Estes autores referem, no entanto, que, para muitos investigadores, nenhuma medida isolada pode ser considerada superior as restantes.

Tabela 2.1. Métricas de avaliação

Métrica	Fórmula
Raiz do erro quadrático médio (RMSE)	$\sqrt{[\Sigma(\text{Observação} - \text{Previsão})^2 / n]}$
Erro Absoluto Médio (MAE)	$(1/n) * \Sigma   \text{Observação} - \text{Previsão}  $
Erro Percentual Absoluto Médio (MAPE)	$(1/n) * \Sigma (  \text{Observação} - \text{Previsão}   /   \text{Real}  ) * 100$
Erro quadrático médio (MSE)	$(1/n) * \Sigma (\text{Observação} - \text{Previsão})^2$
Erro Médio (ME)	$(1/n) * \Sigma (\text{Observação} - \text{Previsão})$

Das métricas principais, apresentadas na Tabela 2.1, algumas são dependentes de escala (ME / MAE / MSE / RMSE) o que as torna inadequadas para comparar a exatidão de um método utilizando séries diferentes (Hyndman & Koehler, 2005). Para avaliar a exatidão de um método que utilize uma única série, os mesmos autores consideram a métrica MAE como sendo uma boa opção, por ser fácil de calcular e de compreender. Outras possibilidades passam pela utilização de métricas baseadas em erros percentuais, como por exemplo a MAPE. Estas têm a vantagem de serem independentes de escala e, por isso, frequentemente utilizadas na comparação do desempenho de previsão entre



diferentes séries de dados. No entanto, Hyndman e Koehler (2005) identificam duas desvantagens na utilização desta métrica: não poder ser utilizado se as séries tiverem valores zero e, apesar de considerar o valor absoluto do erro percentual, penalizar mais os erros positivos do que os negativos. Na existência destes constrangimentos, ao invés da utilização da MAPE, estes autores propõem a utilização de métricas baseadas em valores absolutos, como sejam a MAE e MAPE, ou valores quadrados, como sejam a MSE e RMSE, para avaliar a exatidão das previsões.

Chen et al. (2017) por seu lado, referem que a evolução da utilização das medidas de exatidão pode ser avaliada pela forma como estas têm sido utilizadas nos principais estudos comparativos de métodos de previsão. Tendo por base estas competições, estes autores, identificam as métricas RMSE e MAPE como sendo as mais antigas e mais populares.

## 2.6 Software

### 2.6.1 R

O R é uma linguagem de programação que se baseia na utilização de funcionalidades disponibilizadas em pacotes. O seu desenvolvimento é da responsabilidade do R Core Team (2021) e todas as restantes equipas de desenvolvimento contribuem com funcionalidades que são depois disponibilizadas em repositórios de pacotes. O principal repositório dos pacotes é o *CRAN* existindo, no entanto, outros sobretudo vocacionados para áreas científicas muito específicas (Theußl et al. 2011). Segundo estes autores a criação e desenvolvimento de *software* de forma descentralizada é promovida, chegando mesmo a ser submetidos mais de 300 pacotes por dia. Para conseguir a sua publicação é obrigatório garantir o cumprimento de um conjunto de regras *standard* ao nível: da programação, da forma como é feita a documentação do código, da forma como as respetivas funcionalidades são documentadas, na obrigatoriedade de fornecimento de dados e de cenários que permitem a sua avaliação e teste.

De entre das várias linguagens de programação *open-source*, o R é, atualmente, uma referência na área das previsões em séries temporais (Fildes et al. 2020) e disponibiliza inúmeras funcionalidades dedicadas ao processo de previsão. Apesar de existirem

funcionalidades que cobrem todas as etapas do processo de previsão, nenhuma o faz de forma integrada, o que permitiria uma utilização mais fácil e interativa.

### 2.6.2 Shiny

O R disponibiliza um conjunto de ferramentas altamente flexíveis e poderosas para analisar e visualizar dados. O Shiny (V1.5.0; Chang et al. 2020), por seu lado, permite agrupá-las num aplicativo *web* para que qualquer pessoa as possa usar (Satyahadewi & Perdana, 2021). Segundo Wickham (2020), o Shiny torna significativamente mais fácil a criação de aplicações *Web* porque fornece um conjunto de funções que geram o HTML, CSS e JavaScript de uma forma automática. Ainda de acordo com este autor, o Shiny esta a ser utilizado: na criação de painéis de controlo onde são apresentados indicadores de desempenho de alto nível, permitindo também a sua exploração detalhada na comunicação de modelos complexos, a um público não técnico com a utilização de visualizações informativas e análises de sensibilidade interativas; na indústria farmacêuticas para acelerar a colaboração entre cientistas e analistas durante o desenvolvimento de medicamentos; nas empresas tecnológicas de Silicon Valey para apresentar painéis de controlo em tempo real que incorporam análises avançadas e a nível da academia como uma ferramenta de ensino para conceitos estatísticos. Uma pesquisa no Google Académico permite confirmar a existência de inúmeras referências a trabalhos científicos em que o Shiny é citado.

Jahanshiri e Shariff (2014) referem que a sua utilização permite criar aplicações *web* que permitem a análise de dados em tempo real algo que anteriormente era difícil de conseguir e, Satyahadewi e Perdana (2021), classificam-no como o “companheiro” perfeito para o R, tornando rápido e simples partilhar análises gráficas com as quais os utilizadores podem facilmente interagir e consultar através de tablets ou smartphones. O Shiny não se limita a criar aplicativos triviais ou pré-fabricados, os seus componentes de interface podem ser facilmente customizados ou estendidos (Wickham, 2020). Resumidamente, oferece a capacidade de expor os superpoderes R a qualquer pessoa que use um navegador da *web*.

O Shiny é composto por 3 componentes (Satyahadewi & Perdana, 2021):

- Componente interativa – tem o layout da aplicação e disponibiliza o acesso aos componentes interativos;
- Componente do servidor – recebe os inputs da componente interativa, efetua os processamentos necessários, produz os resultados de saída que são enviados para a componente interativa;
- Função ShinyApp – corresponde ao próprio aplicativo e que combina as duas componentes.

A sua programação reativa, rastreia automaticamente as dependências entre os componentes interativos e o respetivo código, o que permite que sempre que alguma entrada seja alterada, o Shiny despolete, em automático, os processamentos que lhe estão ligados o que, por sua vez, provoca a atualização de todas as saídas relacionadas (Satyahadewi & Perdana, 2021).

### **2.6.3 Pacotes para previsão com funcionalidades múltiplas**

Recentemente surgiram vários pacotes que disponibilizam funcionalidades interessantes para projetos na área das previsões baseadas em séries temporais. Seguidamente, é feita uma apresentação sumária de um grupo desses pacotes que, de forma completa, ou pelo menos parcial, cobrem um dos seguintes tópicos: previsão de modelos; otimização de previsão; avaliação do modelo de previsão; comparação de modelos ou automatização de processos.

#### **predtoolsTS: Time Series Prediction Tools**

O pacote predtoolsTS (v0.1.1; Moreno, Rivas & Godoy, 2018) permite efetuar previsões de séries temporais utilizando métodos estatísticos ou métodos de *machine learning*. De acordo com Charte *et al.* (2019), o pacote está atualmente limitado aos métodos disponibilizados pelos pacotes forecast e caret e funciona numa lógica de encadeamento de tarefas que garantem quatro fases da geração de uma previsão: processamento e preparação de dados; treino e teste de modelos; geração da previsão; pós processamento dos dados (caso tenha existido transformação de dados na fase de preparação). Apesar de permitir uma avaliação comparativa, esta está limitada a cinco modelos.

### **ForecastTB: Test Bench for the Comparison of Forecast Methods**

O pacote ForecastTB (v1.0.1; Neeraj, Dhanraj & Bokde, 2020) tem como objetivo final a comparação da performance de diferentes modelos de previsão em séries temporais. Segundo Bokde et al. (2020), este pacote permite a seleção das métricas de avaliação e utiliza o método de Monte Carlo para efetuar as respetivas otimizações. Não está limitado em termos de métodos, permitindo inclusivamente adicionar outros métodos, quer sejam programados em R ou noutras linguagens, como, por exemplo, em MatLab. Disponibiliza uma forma normalizada de avaliação de modelos, permite a automatização de processos, garante uma avaliação verdadeira e imparcial, permite reproduzir os resultados e os testes efetuados, permite facilitar o processo aos utilizadores que não sejam proficientes na linguagem R.

### **ForecastComb: Forecast Combination Methods**

Segundo Weiss et al. (2019), ainda que empiricamente, já se encontra bem estabelecido que, a combinação diferentes métodos de previsão, pode resultar em melhores previsões. De acordo com a sua avaliação, este pacote forecastComb (v 1.3.1; Weiss, Roetzer & Raviv 2018), fornece um conjunto de ferramentas que ajudam os utilizadores ao longo de todo o processo de combinação dos diferentes modelos de previsão. As melhores técnicas de combinação podem ser identificadas: automaticamente, computando todos os métodos de combinação disponíveis com relação a um critério de perda; manualmente testando independentemente cada uma das quinze técnicas de combinação de pesos diferentes suportadas. Obriga a ter pelo menos dois modelos de previsão e, uma vez que o pacote não fornece nenhuma ferramenta para criá-los, têm de utilizar modelos disponibilizados por outros pacotes. A otimização pode ser efetuada manualmente, atribuindo a cada modelo o respetivo peso, automaticamente deixando a ferramenta a responsabilidade da sua otimização. A precisão dos modelos pode ser calculada com base numa das seguintes métricas: ME, RMSE, MAE, MPE, MAPE.

### **forecastHybrid: Convenient Functions for Ensemble Time Series Forecasts**

O objetivo do pacote forecastHybrid (v5.0.19; Shaub & Ellis, 2020) é construir modelos combinados. As previsões podem ser geradas combinando métodos de previsão, como

ARIMA, ETS, THETAF, NNAR, STL *Decomposition*, BATS e *Seasonal Naïve* do pacote *forecast*. Três técnicas de combinação diferentes são disponibilizadas: pesos iguais, pesos baseados em erros na amostra, pesos baseados em erros na validação cruzada. A avaliação da precisão do modelo também pode ser realizada aplicando técnicas de validação cruzada. A construção dos modelos pode ser feita de forma automática ou manual com a indicação de um número mínimo de parametrizações. A geração dos modelos é intuitiva e permite ainda visualizar os resultados e respetiva precisão. A otimização dos modelos pode ser feita selecionando apenas a lista de modelos, o método de combinação de pesos e a métrica de precisão do modelo. Os ajustes do modelo podem ser avaliados visualmente pela comparação dos valores ajustados e reais. Também permite a utilização da validação cruzada para a avaliação do desempenho dos modelos.

### **TSstudio: Functions for Time Series Analysis and Forecasting**

O pacote TSstudio (v0.1.6; Rami & Krispin, 2020) disponibiliza um conjunto de ferramentas para a análise de dados e de modelos de previsões. Krispin (2020) indica que, ao nível da exploração de dados, disponibiliza um conjunto muito completo de representações gráficas que permitem evidenciar as características das séries de dados tais como: sazonalidade; autocorrelação; decomposição; análise de *lags*. A modelação é feita de forma incremental, adicionando, passo a passo, cada um de seus componentes: série de dados, métodos de previsão, métodos de treino, horizonte de previsão, intervalo de confiança, e a avaliação e comparação de modelos é feita com base nos respetivos resíduos. Está limitado à previsão de apenas um modelo de cada vez, mas permite a simulação de previsões. Está limitado aos modelos disponibilizados pelo pacote *forecast*.

Na Tabela 2.2 apresenta-se um resumo das vantagens e desvantagens de cada um dos pacotes apresentados anteriormente e a

Tabela 2.3 sintetiza o âmbito das funcionalidades presentes em cada um deles.

*Tabela 2.2. Pacotes R – Resumo*

<b>Pacote</b>	<b>Vantagens</b>	<b>Desvantagens</b>
predtoolsTS	Métodos estatísticos e de <i>machine learning</i> . Número de métodos de previsão disponíveis. Transformação automática do conjunto de dados. Abrange todas as etapas do processo de previsão.	Ferramentas para comparar os resultados do modelo de previsão. Número limitado de métodos que podem ser comparados.
ForecastTB	Novos modelos e métricas podem ser adicionados ao processo. Fornece uma comparação verdadeira e imparcial. Os resultados podem ser facilmente replicados. Estratégia de Monte Carlo para identificar a previsão mais adequada.	Número de métodos de previsão disponíveis Recursos gráficos limitados
ForecastComb	O processo de combinação automática. Gama de métodos de otimização. Variedade de métodos de previsão disponíveis. Possibilidade de combinar métodos estatísticos e de previsão de <i>machine learning</i> .	Não permite a previsão intervalar
forecastHybrid	Método de combinação automática. Fácil de configurar. Recursos de validação cruzada para avaliar a precisão do modelo.	Número de métodos de previsão disponíveis
TSstudio	Boas ferramentas de análise de dados. Capacidades gráficas. Facilidade na criação, avaliação e comparação dos modelos.	Número de métodos de previsão disponíveis.

*Tabela 2.3. Pacotes R – Âmbito*

<b>Pacote</b>	<b>Exploração de dados</b>	<b>Preparação dos dados</b>	<b>Geração de modelos</b>	<b>Avaliação de modelos</b>	<b>Comparação dos modelos</b>
predtoolsTS	Não	Sim	Sim	Sim	Sim
ForecastTB	Não	Não	Sim	Sim	Sim
ForecastComb	Não	Não	Não	Sim	Sim
forecastHybrid	Não	Não	Sim	Sim	Não
TSstudio	Sim	Não	Sim	Sim	Não

### 3 METODOLOGIA

Os projetos de desenvolvimento de *software* são normalmente difíceis de gerir por terem frequentes alterações dos seus requisitos e consequentemente do seu planeamento. Estas alterações podem resultar de uma má definição inicial dos requisitos funcionais, do seu mau entendimento, do aparecimento de novas oportunidades de negócio, da necessidade de inclusão de novos requisitos, da falta de capacidade técnica das equipas de desenvolvimento, entre outros fatores (Laranjeira et al. 2019; Trigo & Varajão, 2020). A sua difícil gestão está associada à imprevisibilidade, que resulta dessas frequentes alterações, daí a necessidade de se utilizar uma metodologia que sirva de guia e estabeleça regras que suportem a sua gestão.

Para a escolha da metodologia foi utilizada como referência a sistematização de Despa (2014). Segundo ela a gestão de projetos de *software* segue as mesmas regras básicas dos restantes projetos, mas tem a preocupação de se ajustar às suas especificidades. Estes projetos têm tipicamente as seguintes fases:

- Pesquisa: definição dos requisitos e dos objetivos;
- Desenho: definição do layout da aplicação;
- Planeamento: definição do fluxo geral do projeto e identificação dos entregáveis;
- Desenvolvimento: programação;
- Testes: avaliação e testes da aplicação face aos requisitos definidos;
- Aprovação: aceitação por parte do destinatário da aplicação;
- Setup: disponibilização da aplicação em produção;
- Manutenção: gestão corrente da aplicação quando esta se encontra em utilização.

Segundo Despa (2014), existem dois grandes grupos de metodologias:

- Metodologias do tipo *High Weight* adequadas a projetos em que os requisitos são claramente definidos desde o início e em que não são expectáveis alterações significativas no decurso do projeto. Este tipo de metodologia não prevê alterações na sequência das etapas e obrigam a que a etapa seguinte apenas seja



iniciada depois de terminada a anterior. São modelos claramente menos flexíveis, mas mais previsíveis e, portanto, mais fáceis de controlar e de gerir.

- Metodologias do tipo *Low Weight* por sua vez seguem a filosofia *Agile* e são ajustadas a projetos em que os requisitos, ou não são claros, ou em que existe uma grande probabilidade de serem alterados no decurso do projeto. São metodologias flexíveis, porque preveem múltiplas interações e permitem a redefinição dos requisitos e revisão do planeamento. Funcionam muito na base dos desenvolvimentos incrementais, em que os projetos são repartidos em módulos tratando-os como se de um subprojecto se tratasse, e em que, no final de cada módulo, normalmente resulta num protótipo. Adaptam-se bem a projetos tecnológicos, porque acomodam as alterações de requisitos e lidam bem com alterações do enquadramento em que se desenvolve o projeto.

A escolha da melhor metodologia de gestão por sua vez deve ter em consideração os seguintes fatores e preocupações:

- Tipo de projeto;
- Requisitos;
- Complexidade;
- Objetivo;
- Entregáveis,
- Necessidades em termos de documentação;
- Enquadramento,
- Informação disponível;
- Duração;
- Utilização futura;
- Constrangimentos orçamentais.

Considerando os constrangimentos temporais, os ciclos de desenvolvimento previstos, resulta que uma metodologia do tipo *Low Weigth* se ajusta melhor a este projeto. A metodologia Desenvolvimento Rápido de Aplicação (RAD), cujo fluxo é esquematizado na Figura 3.1, utiliza uma lógica de desenvolvimento incremental e iterativa em que no final de cada ciclo resulta um produto final, ainda que incompleto, denominado por protótipo. É uma metodologia flexível, que permite acomodar revisões de requisitos e que prevê as revisões do planeamento face aos riscos e à realidade do momento.

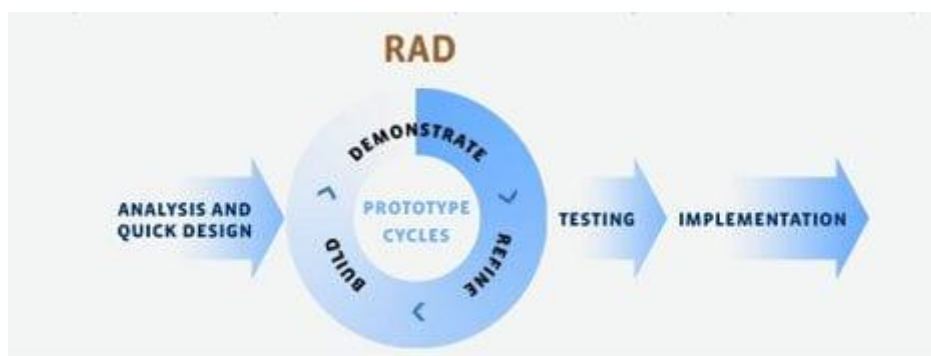


Figura 3.1. Metodologia Desenvolvimento Rápido de Aplicação

Beynon-Davies et al. (1999) classificam-na como uma metodologia completa para o desenvolvimento de sistemas de informação, por cobrir todo o ciclo de vida, desde o início até à sua entrega. Segundo os autores, o criador da metodologia tinha como objetivos: produzir sistemas de alta qualidade, rapidez no desenvolvimento e entrega e também baixos custos. Trata-se de uma metodologia extremamente popular para o desenvolvimento de sistemas iterativos e é caracterizada pela prototipagem incremental, pelo foco no produto, por garantir as entregas e por um grande envolvimento dos utilizadores (Beynon-Davies & Holmes, 2002). Segundo estes autores, a maioria destes projetos têm pequena dimensão, curta duração e possuem um grupo de utilizadores claramente definido. Tipicamente estes projetos são desenvolvidos por fases organizadas em função das entregas previstas (Beynon-Davies & Holmes, 2002).

A metodologia resulta na implementação de um conjunto sequencial de etapas:

1. Análise de requisitos em que as partes interessadas discutem e concordam sobre as necessidades do negócio, o âmbito do projeto, as restrições e os requisitos do sistema;

2. Design em que utilizadores e analistas de sistemas desenham os modelos e protótipos que representam os processos, os *inputs* e *outputs* do sistema;
3. Construção que corresponde á etapa em que são desenvolvidos os protótipos de acordo com o definido nas etapas anteriores;
4. *Cutover* corresponde à etapa em que é testada protótipo, é feita a sua avaliação face aos requisitos e design definidos, é preparada a sua implementação (Cunha et al. 2015).

Este ciclo de avaliação-discussão-alteração é tipicamente repetido 3 vezes e tem por objetivo refinar continuamente o protótipo até que o utilizador esteja satisfeito com o resultado final (Beynon-Davies et al. 1999).

## 4 DESENVOLVIMENTO DA APLICAÇÃO

Neste capítulo começa-se por descrever os requisitos de cada uma das tarefas da aplicação e por apresentar as tecnologias utilizadas na sua implementação. Na subsecção final descrevem-se as três iterações efetuadas assim como a evolução funcional de cada protótipo até á sua versão final.

### 4.1 Análise de requisitos

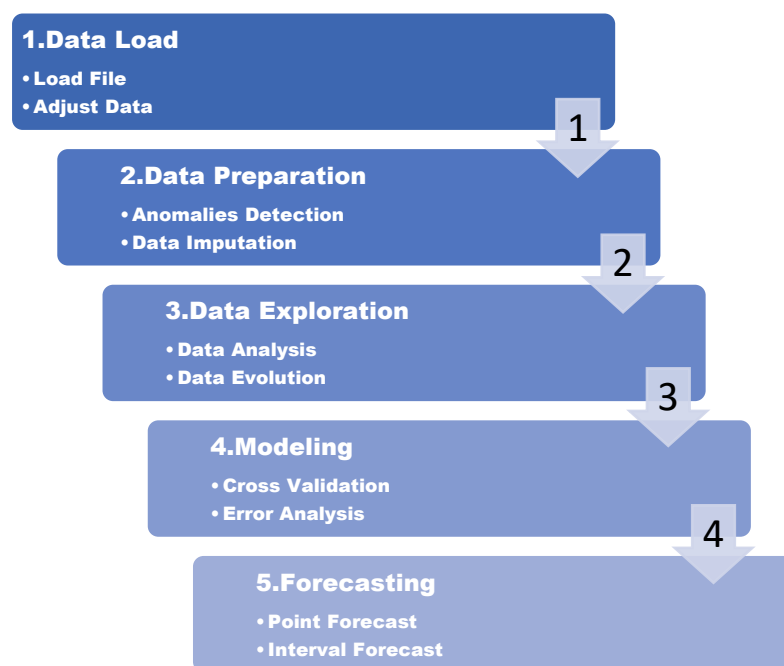


Figura 4.1. Workflow da aplicação

A aplicação deverá automatizar ao máximo os processos, no entanto, deve permitir ao utilizador, o controlo sobre as respetivas configurações. Deverá cobrir todas as etapas do processo de previsão e deverá ser organizada de acordo com o fluxo de tarefas esquematizado na Figura 4.1 e que se descrevem nas subsecções seguintes.

#### 4.1.1 1ª Etapa – Carregar dados (1. Data Load).

Implementar uma funcionalidade que permita selecionar e carregar os dados de um ficheiro CSV. Criar uma tabela em que são apresentados os cinco primeiros e os cinco últimos registos. Criar a possibilidade de os utilizadores indicarem: as colunas das datas, a coluna das observações, a periodicidade das observações (Diária/Mensal/Anual), o

formato da data. Criar uma funcionalidade que gere a série temporal. Criar um gráfico que apresente a série.

#### **4.1.2 2ª Etapa – Transformar Dados (2. Data Preparation)**

Implementar uma funcionalidade que permita detetar e tratar as anomalias.

Para a deteção de *outliers* deve implementar a funcionalidade descrita na subsecção 2.3.1 e que permite ao utilizador ajustar a respetiva sensibilidade através do *IQR factor*.

Para o tratamento das falhas de dados deverá implementar a funcionalidade descrita na subsecção 2.3.1 disponibilizando o seguinte conjunto de modelos de imputação: *Multiple imputation using Fully Conditional Specification* (FCS) do package MICE (van Buuren & Groothuis-Oudshoorn, 2011); e, do package *imputeTS* (Moritz & Bartz-Beielstein, 2017), Linear Interpolation, Spline Interpolation, Stineman Interpolation, Kalman StructTS, Kalman StructTS Smooth, Kalman Space Models, Seasonally Decomposed Interpolation, Seasonally Decomposed Mean, Seasonally Decomposed Kalman Filter. Os dados imputados devem ser apresentados num gráfico. O número de *outliers* gerados na imputação de cada uns dos métodos implementados devem ser resumidos e apresentados numa tabela. Para tornar visível os *outliers* gerados estes devem também ser apresentados num gráfico.

Devem ser implementados dois botões: 1) *Test* que faz a simulação da imputação dos dados utilizando o modelo selecionado, 2) *Apply* um botão que aplica na série a imputação selecionada.

#### **4.1.3 3ª Etapa – Explorar Dados (3. Data Exploration)**

Criar um conjunto alargado de representações gráficas da série. Os gráficos devem ser ajustados de acordo com a periodicidade dos dados (Diária/Mensal/Anual) e devem permitir: explorar a série temporal, apresentar a tendência e sazonalidade, apresentar vários níveis de granularidade dos dados, sempre que possível devem ser interativos. Devem estar organizados em submenus. Para a sua implementação utilizar as funcionalidades dos pacotes: *dygraphs* (V1.1.1.6; Vanderkam et al. 2018), *forecast*

(V8.12; Hyndman et al. 2020), ggplot2 (V3.3.2; Wickham et al. 2020), plotly (V4.9.2.1; Sievert et al. 2020).

Implementar também a possibilidade de os utilizadores efetuarem uma decomposição STL. Esta deve permitir ao utilizador a seleção das componentes que pretende visualizar e deve ser utilizada a função `mstl`<sup>9</sup> do pacote `forecast` (V8.12; Hyndman et al., 2020). Complementar a informação com um histograma dos valores residuais e com um gráfico que, indique as intensidades da tendência e sazonalidade da série, calculada pela função `feat_stl` do pacote `feasts` (V0.1.4; O'Hara-Wildet et al. 2020).

#### 4.1.4 4ª Etapa – Treinar Modelos (4. Modeling)

Implementar quatro formas distintas de treinar e modelos: 1) Fixed Window, 2) Random Fixed Window, 3) Rolling Window, 4) Random Rolling Window de acordo com o descrito na subsecção 2.3.2. Adicionar a possibilidade de identificar os seguintes parâmetros: 1) a posição inicial da série; 2) o número de observações para treino; 3) o número de observações para testes; 4) o salto de posições entre séries; 5) o número máximo de séries; e 6) o tipo de validação a utilizar. Sempre que uma das parametrizações é alterada a aplicação deverá reconstruir as séries e atualizar os gráficos respetivos. Deverá ser disponibilizada a seguinte lista de modelos de previsão: BATS, ETS, NNAR, Auto.Arima, Naïve, Seasonal Naïve, RWF do pacote `forecast` (V8.12; Hyndman et al. 2020), `StructTS`, `HoltWinters` do pacote `stats` (V4.0.3; R Core Team 2020), `Prophet` do pacote `prophet` (V0.6.1; Taylor & Letham, 2020). A multiselecção de modelos deverá ser possível para permitir a comparação dos respetivos resultados. Disponibilizar um botão *Calculate* para desencadear o treino dos respetivos modelos.

A avaliação dos modelos deverá ser feita de acordo com a métrica de avaliação descritas na subsecção 2.5.3. Os resultados deverão ser apresentados da seguinte forma:

- Uma tabela com o *ranking* dos três melhores modelos para cada horizonte das previsões (para a métrica de erro selecionada);

---

<sup>9</sup> Esta função `mstl` tem a vantagem, relativamente à função `stl` do package `stats`, de ser completamente automática.

- Uma tabela com a informação detalhada dos erros médios por modelo e horizonte de previsão;
- Um gráfico *boxplot* com a informação anterior;
- Um gráfico comparativo com a evolução dos erros de cada modelo para cada um dos horizontes de revisão;
- Um gráfico *boxplot* para cada modelo com todos os erros por horizontes temporais;
- Gráfico *boxplot* dos erros para cada uma das métricas de avaliação e para cada modelo;
- Para todas as métricas de avaliação um gráfico comparativo com a evolução dos erros de cada modelo para cada um dos horizontes de revisão.

#### **4.1.5 5ª Etapa – Gerar Previsões (5. Forecasting)**

Deverão ser disponibilizadas duas possibilidades de previsão:

- 1) Pontual, terá os mesmos modelos utilizados na etapa anterior, o que permitirá a multiselecção de modelos e que permitirá indicar o respetivo horizonte temporal,
- 2) Intervalar, que estará limitado à utilização de apenas um modelo.

Os resultados da previsão pontual deverão ser apresentados numa tabela com os valores da previsão, modelo a modelo, e para cada um dos horizontes temporais indicados. Deverá também ser criado um gráfico em que a série original é apresentada em conjunto com os valores da previsão pontual de cada um dos modelos.

A previsão intervalar deverá estar limitada a intervalos de previsão de 80% e a 95% e os resultados devem ser apresentados numa tabela e num gráfico em que a série original é apresentada em conjunto com os valores da previsão intervalar.

## **4.2 Tecnologias selecionadas**

O R foi a linguagem de programação escolhida fazendo-se uso de todos os pacotes necessários para a implementação das respetivas Figura 4.2. Para a integração das

diferentes funcionalidades numa aplicação web única será utilizado o pacote Shiny descrito na subsecção 2.6.2.

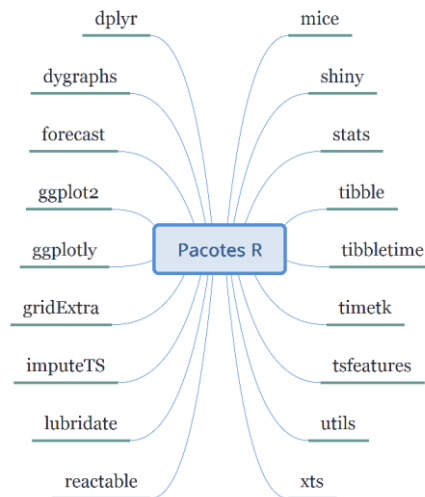


Figura 4.2. Pacotes R usados na aplicação.

### 4.3 Ciclos de desenvolvimento

A aplicação foi desenvolvida ao longo de 8 meses tendo sido criados dois protótipos, que correspondem às iterações 1 e 2, e a aplicação final, descrita e apresentada no capítulo 5.1, que resultou da 3ª e última iteração. O desenvolvimento foi efetuado apenas pelo autor do relatório e os testes foram efetuados em conjunto com a coordenadora do projeto. Em cada iteração foi efetuada uma avaliação crítica das funcionalidades implementadas, tendo ocorrido frequentemente a necessidade de revisão dos requisitos face aos resultados dos testes e à identificação de novas necessidades. Os protótipos gerados e as respetivas evoluções são descritas nas subsecções seguintes.

#### 4.3.1 Iteração 1

Nesta interação foram implementadas as seguintes funcionalidades:

- 1) Carregamento de dados: a leitura de ficheiros, tabela com dados carregados;
- 2) Exploração de dados: estatísticas dos dados, a deteção de anomalias, gráficos com a evolução da série, gráficos de ACF / PACF, gráfico com representação da série;



- 3) Análise de dados: gráficos de decomposição da série, gráficos de sazonalidade mensal, gráficos de evolução mensal;
- 4) Ajustamento: treino dos modelos de previsão, análise gráfica dos resultados, análise gráfica dos erros de previsão;
- 5) Avaliação dos modelos: tabela com avaliação dos resultados das diferentes simulações.

A título de exemplo apresenta-se, de seguida, o código utilizado na implementação da funcionalidade da imputação de dados em falta (c.f. Código R 4-1).

```

# -----
# Identifica e aplica método selecionado
# -----
observeEvent(input$imp_Button2,
{
  showModal(modalDialog("Running process", footer=NULL))
  rv$plotData<-NULL
  rv$imputedData<-NULL
  if (input$imputationMethod=="mice" & !is.null(rv$dadosFicheiro))
    rv$imputedData<-miceData(rv$dadosFicheiro,input$dateType)
  if (input$imputationMethod=="linear.interp" & !is.null(rv$dadosFicheiro))
    rv$imputedData<-naInterpData(rv$dadosFicheiro,"linear",rv$Frequency)
  if (input$imputationMethod=="spline.interp" & !is.null(rv$dadosFicheiro))
    rv$imputedData<-naInterpData(rv$dadosFicheiro,"spline",rv$Frequency)
  if (input$imputationMethod=="stine.interp" & !is.null(rv$dadosFicheiro))
    rv$imputedData<-naInterpData(rv$dadosFicheiro,"stine",rv$Frequency)
  if (input$imputationMethod=="na.struct" & !is.null(rv$dadosFicheiro))
    rv$imputedData<-naStructTSDData(rv$dadosFicheiro,rv$Frequency)
  if (input$imputationMethod=="kalman.1" & !is.null(rv$dadosFicheiro))
    rv$imputedData<-naKalmanData(rv$dadosFicheiro,'StructTS','FALSE',rv$Frequency)
  if (input$imputationMethod=="kalman.2" & !is.null(rv$dadosFicheiro))
    rv$imputedData<-naKalmanData(rv$dadosFicheiro,'StructTS','TRUE',rv$Frequency)
  if (input$imputationMethod=="kalman.3" & !is.null(rv$dadosFicheiro))
    rv$imputedData<-naKalmanData(rv$dadosFicheiro,'SpaceModels','TRUE',rv$Frequency)
  if (input$imputationMethod=="seadec.1" & !is.null(rv$dadosFicheiro))
    rv$imputedData<-naSeadecData(rv$dadosFicheiro,'interpolation',rv$Frequency)
  if (input$imputationMethod=="seadec.2" & !is.null(rv$dadosFicheiro))
    rv$imputedData<-naSeadecData(rv$dadosFicheiro,'mean',rv$Frequency)
  if (input$imputationMethod=="seadec.3" & !is.null(rv$dadosFicheiro))
    rv$imputedData<-naSeadecData(rv$dadosFicheiro,'kalman',rv$Frequency)

  rv$plotData<-cbind(rv$dadosFicheiro,rv$imputedData)
  colnames(rv$plotData)<-c('date','actual','imputed')
  #View(rv$plotData[3])
  removeModal()
})

# -----
# Completa dados em falta mice
# -----
miceData<-function(df_in,miceType){
  if (is.null(df_in) | ncol(df_in)!=2)
    return (NULL)

  date<-lubridate::as_date(df_in[,1])
  value<-as.numeric(df_in[,2])
  year<-lubridate::year(date)
  month<-lubridate::month(date)
  quarter<-ceiling(month/3)
  semester<-ceiling(month/6)
  df_mice<-cbind.data.frame(date,year,month,quarter,semester,value)

  preds_mice <-mice::mice(df_mice)
  pred_values <- mice::complete(preds_mice)

  df_out<-pred_values[,6]

```

```

    return(df_out)
}

# -----
# Completa dados em falta - imputeTS::na_interpolation ####
# -----
naInterpData<-function(df_in,interpType="linear",pFrequency=12){
  if (is.null(df_in) | ncol(df_in)!=2)
    return (NULL)

  df_ts<-xts::xts(as.numeric(df_in[,2]),order.by=as.Date(df_in[,1]),frequency=pFrequency)
  df_out<-as.numeric(imputeTS::na_interpolation(df_ts,option=interpType))

  return(df_out)
}

# -----
# Completa dados em falta - imputeTS::na_kalman ####
# -----
naKalmanData<-function(df_in,pModel="StructTS",pSmooth="TRUE",pFrequency=12)
{
  if (is.null(df_in) | ncol(df_in)!=2)
    return (NULL)

  df_ts<-xts::xts(as.numeric(df_in[,2]),order.by=as.Date(df_in[,1]),frequency=pFrequency)
  if (pModel=="StructTS" & pSmooth=="TRUE")
    df_out<-as.numeric(imputeTS::na_kalman(df_ts,model=pModel,smooth =TRUE))
  if (pModel=="StructTS" & pSmooth=="FALSE")
    df_out<-as.numeric(imputeTS::na_kalman(df_ts,model=pModel,smooth =FALSE))
  if (pModel=="SpaceModels" & pSmooth=="TRUE")
    df_out<-as.numeric(imputeTS::na_kalman(df_ts,smooth =TRUE))

  return(df_out)
}

# -----
# Completa dados em falta - imputeTS::na_seadec ####
# -----
naSeadecData<-function(df_in,pModel="interpolation",pFrequency=12){
  if (is.null(df_in) | ncol(df_in)!=2)
    return (NULL)

  df_ts<-xts::xts(as.numeric(df_in[,2]),order.by=as.Date(df_in[,1]),frequency=pFrequency)
  df_out<-as.numeric(imputeTS::na_seadec(df_ts,algorithm =pModel))

  return(df_out)
}

```

#### Código R 4-1 – Implementação da imputação dos dados

Após a implementação das alterações e da adição de novas funcionalidades realizou-se uma nova bateria de testes da qual resultou:

- 1) A necessidade de revisão das estatísticas dos dados para permitir a interatividade e simplificar os gráficos;
- 2) A necessidade de revisão das funcionalidades de Ajustamento e Avaliação de modelos para utilizar modelos individuais e modelos híbridos.

### 4.3.2 Iteração 2

Na segunda interação a aplicação foi ajustada de acordo com o resultado do *feedback* da versão anterior e foram adicionadas as seguintes funcionalidades:

- 1) Carregamento de dados: gráfico interativo da série de dados;
- 2) Exploração de dados: gráficos com valores acumulados por períodos, gráficos com análise da série por períodos, gráficos com evolução mensal e anual da série;
- 3) Preparação de dados: adicionadas funcionalidades para deteção de falhas e *outliers*, e de imputação de dados em falta;
- 4) Validação cruzada: implementadas funcionalidades para a utilização da validação cruzada no treino dos modelos, tabela e gráficos comparativos dos erros de previsão;
- 5) Decomposição: gráfico com a decomposição STL, gráfico com a força da tendência, gráfico com a força da sazonalidade, gráficos com análise valores residuais;
- 6) Ajustamento de modelos individuais: treino de modelos, tabela de resultados, tabela de comparação entre modelos, avaliação gráfica dos resultados de cada modelo;
- 7) Ajustamento de modelos híbridos: treino de modelos, tabela de resultados, tabela de avaliação gráfica dos resultados de cada modelo, análise gráfica dos erros de previsão.

A título de exemplo, apresenta-se abaixo o código utilizado na geração de séries de dados da validação cruzada (*c.f.* Código R 4-2).

```
# -----  
# Data Series  
# df_data<-cvData(df,.h,.Window,.initial,.step,.max)  
# -----  
cvData <- function(df,.forecast,.train,.initial,.step,.max,.type)  
{  
  if (is.null(df))  
    return(NULL)  
  
  # Adjust initial parameters  
  .total<-nrow(df)  
  
  df_result<-NULL  
  
  try({  
    if(.type=="Fixed Window")  
    {  
      .initial<-ifelse((.initial+.train+.forecast)<.total,.initial,.total-.train-.forecast)  
      .train<-ifelse(.train>=.forecast,.train,.forecast)  
      .s<-.initial  
      .e<-.total-.train-.forecast  
      .seq<-seq(from=.s,to=.e,by=.step)  
      n<-0  
      for (i in .seq)
```

```

{
  n<-n+1
  if (n>.max)
    break

  .start<-i
  .end<- .start+.train+.forecast
  .mid<- .end-.forecast

  df_train<-cbind(cv=paste0("FOREC-",stringr::str_pad(n, 2, pad = "0")), type="train"
,df[.start:.mid,])
  df_test<-cbind(cv=paste0("FOREC-",stringr::str_pad(n, 2, pad = "0")), type="test",
df[(.mid+1):.end,])
  df_result<-rbind(df_result,df_train,df_test)
}
}}, silent = TRUE)

try({
  if(.type=="Random Fixed Window")
  {
    .initial<-ifelse((.initial+.train+.forecast)<.total,.initial,.total-.train-.forecast)
    .train<-ifelse(.train>=.forecast,.train,.forecast)
    .step<-1
    .s<- .initial
    .e<- .total-.train-.forecast
    .seq<-seq(from=.s,to=.e,by=1)
    .seq<-sample(x=.seq, size=.max, replace=FALSE)

    n<-0
    for (i in .seq)
    {
      n<-n+1
      if (n>.max)
        break

      .start<-i
      .end<- .start+.train+.forecast
      .mid<- .end-.forecast
      df_train<-cbind(cv=paste0("FOREC-",stringr::str_pad(n, 2, pad = "0")),type="train",
df[.start:.mid,])
      df_test<-cbind(cv=paste0("FOREC-",stringr::str_pad(n, 2, pad = "0")),type="test",
df[(.mid+1):.end,])
      df_result<-rbind(df_result,df_train,df_test)
    }
  }}, silent = TRUE)
  try({
    if(.type=="Rolling")
    {
      .initial<-ifelse((.initial+.train+.forecast)<.total,.initial,.total-.train-.forecast)
      .train<-ifelse(.train>=.forecast,.train,.forecast)
      .step<-1
      .s<- .initial
      .e<- .total-.train-.forecast
      .seq<-seq(from=.s,to=.e,by=1)

      n<-0
      for (i in .seq)
      {
        n<-n+1
        if (n>.max)
          break

        .start<- .s
        .end<-i+.train+.forecast
        .mid<- .end-.forecast

        df_train<-cbind(cv=paste0("FOREC-",stringr::str_pad(n, 2, pad = "0")), type="train",
df[.start:.mid,])
        df_test<-cbind(cv=paste0("FOREC-",stringr::str_pad(n, 2, pad = "0")), type="test",
df[(.mid+1):.end,])
        df_result<-rbind(df_result,df_train,df_test)
      }
    }}, silent = TRUE)

    try({
      if(.type=="Random Rolling")
      {
        .initial<-ifelse((.initial+.train+.forecast)<.total,.initial,.total-.train-.forecast)
        .train<-ifelse(.train>=.forecast,.train,.forecast)
        .step<-1

```

```
.s<-initial
.e<-total-train-forecast
.seq<-seq(from=.s,to=.e,by=1)
.seq<-sample(x=.seq, size=.max, replace=FALSE)

n<-0
for (i in .seq)
{
  n<-n+1
  if (n>.max)
    break

  .start<-s
  .end<-i+.train+.forecast
  .mid<-end-.forecast
  df_train<-cbind(cv=paste0("FOREC-",stringr::str_pad(n, 2, pad = "0")), type="train",
df[.start:.mid,])
  df_test<-cbind(cv=paste0("FOREC-",stringr::str_pad(n, 2, pad = "0")), type="test",
df[(.mid+1):.end,])
  df_result<-rbind(df_result,df_train,df_test)
}
}, silent = TRUE)

return(df_result)
}
```

#### Código R 4-2 - Validação cruzada - Geração de séries

Após a implementação das alterações foi realizada uma nova bateria de testes da qual resultou:

- 1) A decisão de retirar os modelos híbridos;
- 2) A necessidade de rever o ajustamento dos modelos;
- 3) A necessidade de rever as funcionalidades da decomposição para permitir a interatividade;
- 4) Rever os gráficos da força da tendência e da sazonalidade;
- 5) Rever as regras e configurações da validação cruzada.

### 4.3.3 Iteração 3

Na terceira e última interação a aplicação foi ajustada de acordo com o resultado do *feedback* da versão anterior e foram adicionadas as seguintes funcionalidades:

- 1) Validação cruzada: tabela com *ranking* dos modelos;
- 2) Previsão pontual: geração de previsões, tabela com os valores previstos, gráfico com as previsões;
- 3) Previsão intervalar: geração de previsões, tabela com os valores previstos, gráfico com as previsões.

A título de exemplo apresenta-se, de seguida, o código utilizado na geração da previsão intervalar (c.f. Código R 4-3).

```
# -----
# Interval Forecasting event ####
# -----
observeEvent(input$calcIntervalForecast,
  {
    #Clean Forecast Values
    showModal(modalDialog("Please wait while forecasting process runs.", footer=NULL, fade=TRUE))
    rv$IntervalForecast<-NULL
    rv$IntervalForecastValues<-NULL
    df<-data.frame(rv$dataOriginal)
    rv$IntervalForecast<-
IntervalforecastTabValues(df,input$intervalforecastModels,input$nIntFor,input$dateType)
    forec<-tail(rv$IntervalForecast,n=input$nIntFor)
    .nc<-ncol(forec)
    rv$IntervalForecastValues<-forec[,c(1,3:.nc)]
    if (is.null(nrow(rv$IntervalForecastValues)))
    {rv$IntervalForecastValues[1]<-
as.character.Date(lubridate::as_date(as.numeric(rv$IntervalForecastValues[1])))}
    else
    {rv$IntervalForecastValues[,1]<-
as.character.Date(lubridate::as_date(as.numeric(rv$IntervalForecastValues[,1])))}

    removeModal()
  })

IntervalforecastTabValues <- function(df,
                                     models,
                                     h = 1,
                                     dataType="month")
{
  if (is.null(df) | is.null(models))
    return (NULL)

  startDate<-as.Date(max(df[,1]))
  dates <- seq(startDate, by = dataType, length.out = h+1)
  dates <- dates[2:(h+1)]

  .cols<-c('date','actual','forecast','upper80','upper95','lower80','lower95')
  .vals<-rep(0,times=length(df[,2]))
  .actuals <- as.matrix(cbind(.vals,.vals,.vals,.vals))
  .dates <- cbind(date=dates,actual=0)
  .results<-cbind(date=lubridate::as_date(as.numeric(df[,1])),actual=as.numeric(df[,2]))
  .results<-rbind(.results,.dates)

  dt_ts<-convertDataReceivedTS(df,dataType,2)

  if ('ETS ANN' %in% models){
    etsANNFcast <- IntervalforecastModelTS(dt_ts, mets, h, vrs = "ANN")
    if (!is.null(etsANNFcast)){
      etsANNFcast<-round(etsANNFcast,3)
      .etsANN <- rbind(.actuals, as.matrix(etsANNFcast))
      .results <- cbind(.results, etsANN = .etsANN)
    }
  }

  if ('ETS MAM' %in% models){
    etsMAMFcast <- IntervalforecastModelTS(dt_ts, mets, h, vrs = "MAM")
    if (!is.null(etsMAMFcast)){
      etsMAMFcast<-round(etsMAMFcast,3)
      .etsMAM <- rbind(.actuals, as.matrix(etsMAMFcast))
      .results <- cbind(.results, etsMAM = .etsMAM)
    }
  }

  if ('ETS ZZZ' %in% models){
    etsZZZFcast <- IntervalforecastModelTS(dt_ts, mets, h, vrs = "ZZZ")
    if (!is.null(etsZZZFcast)){
      etsZZZFcast<-round(etsZZZFcast,3)
      .etsZZZ <- rbind(.actuals, as.matrix(etsZZZFcast))
      .results <- cbind(.results, etsZZZ = .etsZZZ)
    }
  }
}
```

```

if ('BATS' %in% models) {
  batsForecast <- IntervalforecastModel(df, mbats, h)
  if (!is.null(batsForecast)) {
    batsForecast<-round(batsForecast,3)
    .bats <- rbind(.actuals, as.matrix(batsForecast))
    .results <- cbind(.results, bats = .bats)
  }
}

if ('Auto Arima' %in% models) {
  autoarimaForecast <- IntervalforecastModelTS(dt_ts, mautoarima, h)
  if (!is.null(autoarimaForecast)) {
    autoarimaForecast<-round(autoarimaForecast,3)
    .autoarima <- rbind(.actuals, as.matrix(autoarimaForecast))
    .results <- cbind(.results, bats = .autoarima)
  }
}

if ('Struct' %in% models) {
  structForecast <- IntervalforecastModelTS(dt_ts, mstruct, h)
  if (!is.null(structForecast)) {
    structForecast<-round(structForecast,3)
    .struct <- rbind(.actuals, as.matrix(structForecast))
    .results <- cbind(.results, struct = .struct)
  }
}

if ('HW' %in% models) {
  hwForecast <- IntervalforecastModelTS(dt_ts, mhw, h)
  if (!is.null(hwForecast)) {
    hwForecast<-round(hwForecast,3)
    .hw <- rbind(.actuals, as.matrix(hwForecast))
    .results <- cbind(.results, struct = .hw)
  }
}

if ('NAIVE' %in% models) {
  naiveForecast <- IntervalforecastModelTS(dt_ts, mnaive, h)
  if (!is.null(naiveForecast)) {
    naiveForecast<-round(naiveForecast,3)
    .naive <- rbind(.actuals, as.matrix(naiveForecast))
    .results <- cbind(.results, struct = .naive)
  }
}

if ('SNAIVE' %in% models) {
  snaiveForecast <- IntervalforecastModelTS(dt_ts, msnaive, h)
  if (!is.null(snaiveForecast)) {
    snaiveForecast<-round(snaiveForecast,3)
    .snaive <- rbind(.actuals, as.matrix(snaiveForecast))
    .results <- cbind(.results, struct = .snaive)
  }
}

if ('RWF' %in% models) {
  rwfForecast <- IntervalforecastModelTS(dt_ts, mrwf, h)
  if (!is.null(rwfForecast)) {
    rwfForecast<-round(rwfForecast,3)
    .rwf <- rbind(.actuals, as.matrix(rwfForecast))
    .results <- cbind(.results, struct = .rwf)
  }
}

if ('THETAF' %in% models) {
  thetafForecast <- IntervalforecastModel(df, mthetaf, h)
  if (!is.null(thetafForecast)) {
    thetafForecast<-round(thetafForecast,3)
    .thetaf <- rbind(.actuals, as.matrix(thetafForecast))
    .results <- cbind(.results, struct = .thetaf)
  }
}

colnames(.results)<- .cols

return(.results)
}

```

*Código R 4-3 - Implementação da previsão intervalar*

Após a implementação das alterações e da adição de novas funcionalidades realizou-se uma nova bateria de testes da qual resultou a necessidade de correção de alguns erros detetados e de melhorias ao nível da usabilidade tendo ficado definida a estrutura final das funcionalidades da aplicação.

A Tabela 4.1 resume a evolução das funcionalidades ao longo três versões.

*Tabela 4.1 - Evolução das funcionalidades entre versões*

<b>Protótipo-1</b>	<b>Protótipo-2</b>	<b>Aplicação Final</b>
Data Load	Data Load	Data Load
Load File	Load File	Load File
Adjust Data	Adjust Data	Adjust Data
Series Plot	Series Plot	Series Plot
Data Preparation	Data Preparation	Data Preparation
Imputation	Imputation	Imputation
Outliers	Outliers	Outliers
Data Exploration	Data Exploration	Data Exploration
Series Plots	Series Plots	Series Plots
Data Analysis	Seasonality Plots	Seasonality Plots
Decomposition	Evolution Plots	Evolution Plots
Monthly Analysis Plots	STL Decomposition	STL Decomposition
Seasonality Plots	Cross Validation	Cross Validation
Modeling	Series generation	Series generation
Training Plots	Model Evaluation	Model Evaluation
Residuals Plots	Error Analysis	Error Analysis
Model Evaluation	Residuals Analysis	Residuals Analysis
Model Results Table	Error evaluation	Error evaluation
	Decomposition	Forecasting
	STL Decomposition Plots	Point Forecast
	Residuals Plots	Interval Forecast
	Individual Modeling	
	Error Analysis Table	
	Model Comparison table	
	Model Results Plots	
	Combination Modeling	
	Evaluation Table	
	Training Plots	
	Residuals Plots	
	Training Models	
	Results	
	Training Plots	
	Forecast Plots	
	Accuracy Table	



## 5 APLICAÇÃO

Neste capítulo faz-se a apresentação da aplicação e a descrição das suas funcionalidades (secção 5.1). No final utiliza-se a aplicação para exemplificar o processo de imputação de dados (subsecção 5.2.1), o processo da validação cruzada (subsecção 5.2.2), a utilização da aplicação para realizar a totalidade do processo de previsão para uma série de dados (subsecção 5.2.3).

### 5.1 Apresentação da aplicação

A aplicação é composta por três áreas distintas: 1) Menu onde se encontram todas as funcionalidades da aplicação e que estão agrupadas em seis separadores, 2) Definições onde são disponibilizadas aos utilizadores as configurações, 3) Zona onde são apresentados os resultados na forma de tabelas ou de gráficos. A Figura 5.1 mostra estas três áreas no ecrã inicial.

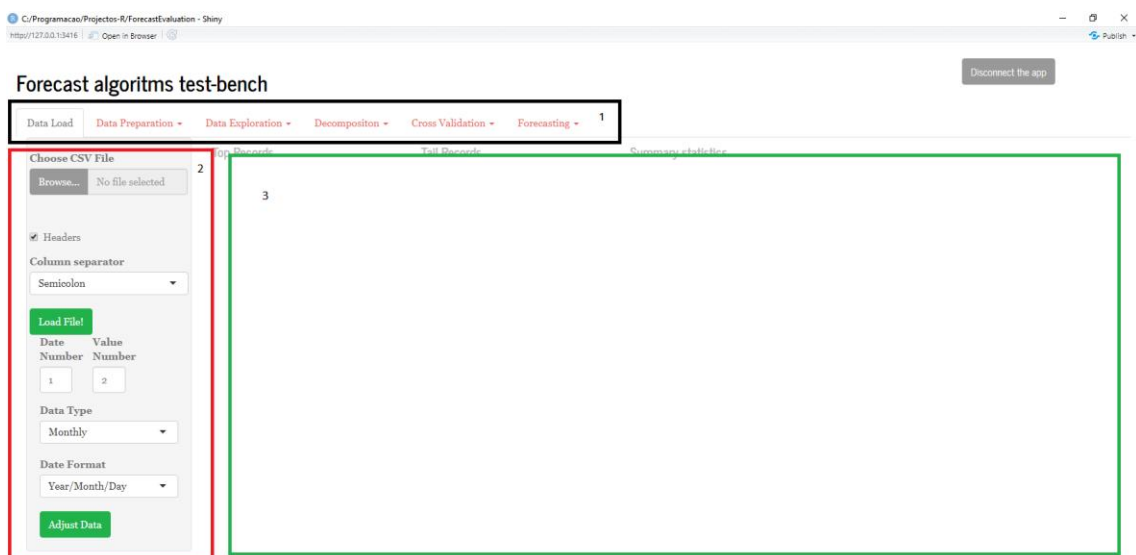


Figura 5.1. Aplicação – Ecrã inicial

As funcionalidades estão organizadas em seis separadores, que correspondem às etapas do processo de previsão definidas na subsecção 2.2. O primeiro menu, *Data Load*, contém as funcionalidades necessárias na etapa do Carregamento dos Dados (descrita na subsecção 4.1.1); o segundo, *Data Preparation*, contém as funcionalidades necessárias na etapa da Transformação dos Dados (descrita na subsecção 4.1.2); o terceiro, *Data Exploration*, e o

quarto, *Decomposition*, correspondem às funcionalidades da etapa da Exploração dos Dados (descrita na subseção 4.1.3); o quinto, *Cross Validation*, disponibiliza as funcionalidades que permitem Treinar e Avaliar os Modelos (descrita na subseção 4.1.4); o sexto e último, *Forecasting*, disponibiliza as funcionalidades da Geração de Previsões (descrita na subseção 4.1.5).

Na Figura 5.2. *Inputs*, são listadas as interações permitidas aos utilizadores em cada uma das funcionalidades e, na Figura 5.3. *Outputs* da aplicação, os *outputs* gerados pela aplicação.

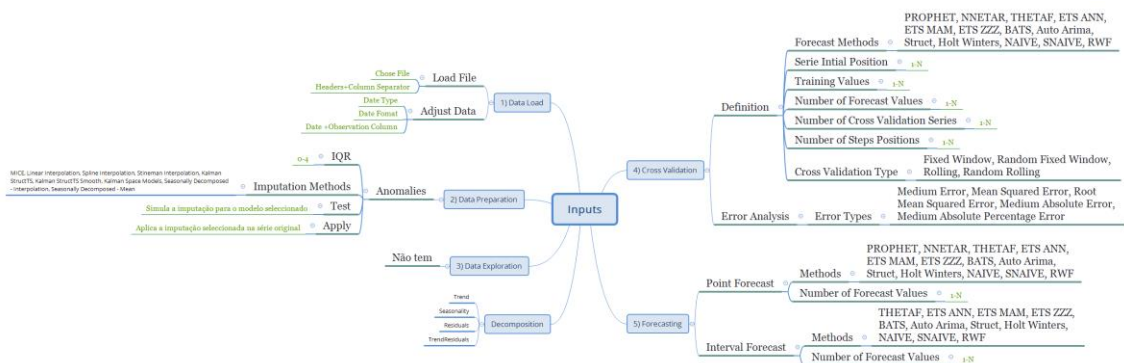


Figura 5.2. Inputs da aplicação

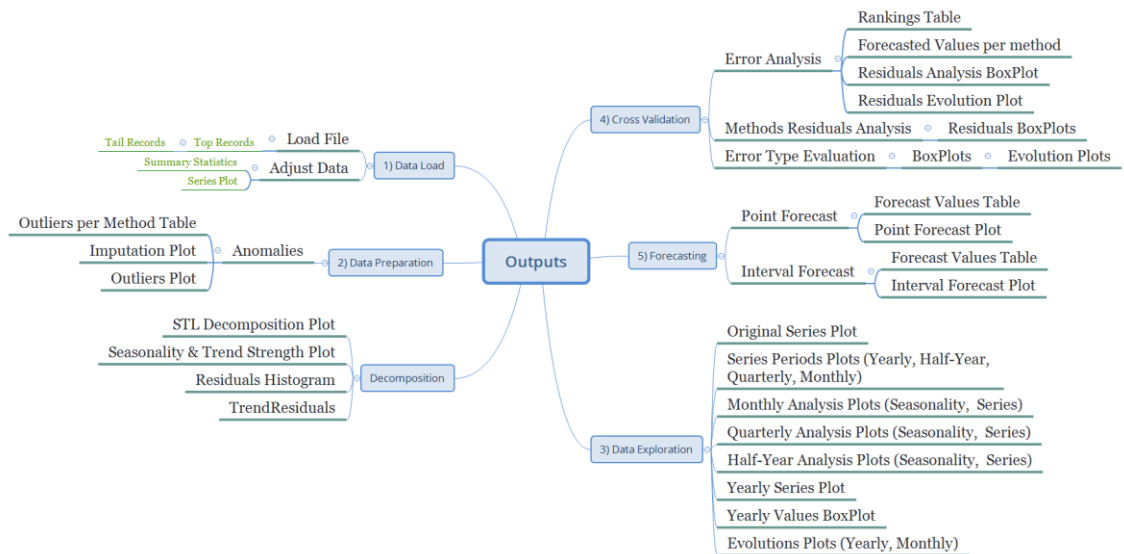


Figura 5.3. Outputs da aplicação

### 5.1.1 Data Load - Carregamento de dados

A primeira tarefa a realizar na ferramenta corresponde ao carregamento dos dados. A Figura 5.1 mostra o ecrã inicial que é também o ecrã *Data Load* antes do carregamento dos dados e a Figura 5.4 apresenta o ecrã *Data Load* com o carregamento dos dados já efetuado.

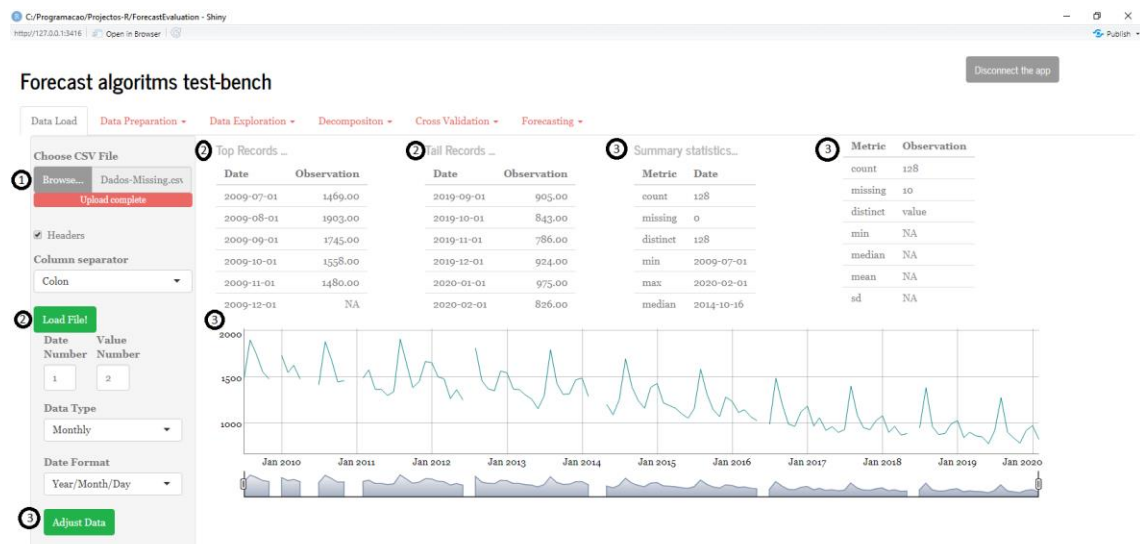


Figura 5.4. Data Load - Carregamento de Dados

Segundo a legenda da Figura 5.4:

- 1) Botão *Browse* dá acesso ao *Explorer* do *Windows*, cabendo ao utilizador a seleção do ficheiro que contem a série de dados. Aplicação apenas permite a leitura de ficheiros do tipo CSV.
- 2) Antes de carregar os dados o utilizador tem que indicar se o ficheiro tem cabeçalhos e também o tipo de separador de colunas. O botão *Load File* carrega os dados que são depois apresentados em duas tabelas com os seis primeiros e últimos registos.
- 3) Para que a aplicação funcione corretamente é necessário manualmente identificar as colunas das datas e das observações, o tipo de periodicidade dos dados e ainda o formato da data. O botão *Adjust Data* converte os dados recebidos numa série temporal e são, então, apresentadas duas tabelas com as estatísticas de cada uma das colunas assim como um gráfico interativo com os dados da série.

### 5.1.2 Data Preparation - Transformação de dados

Esta etapa apenas será necessária efetuar caso existam falhas de dados na série original.

A Figura 5.5 mostra o ecrã do *Data Preparation* e as funcionalidades são descritas seguidamente de acordo com a respetiva legenda.

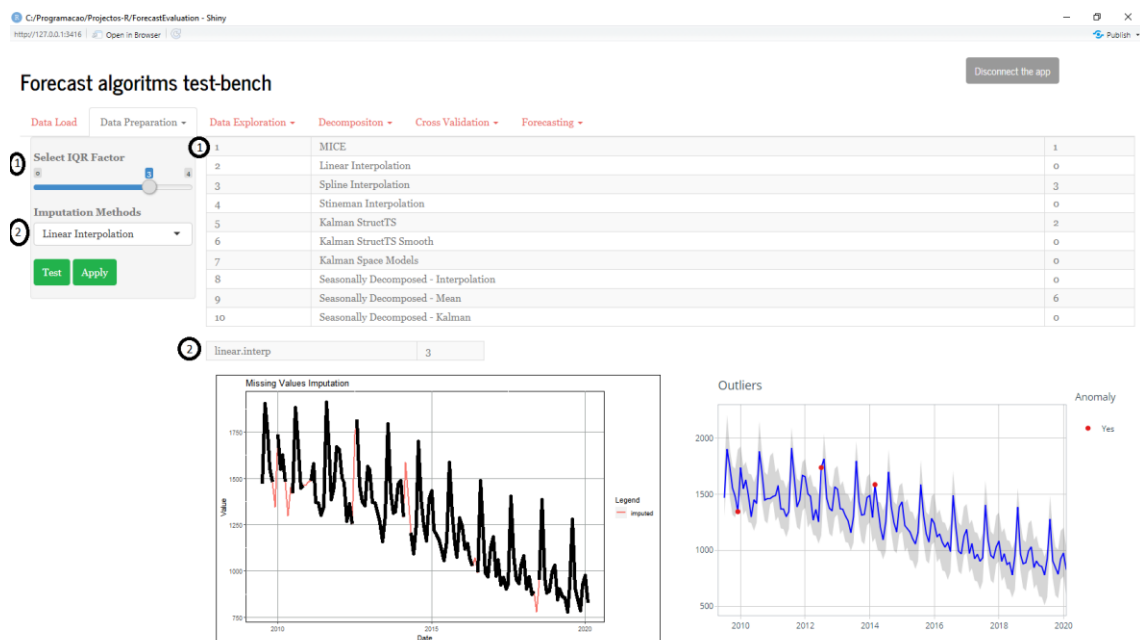


Figura 5.5. Data Preparation - Transformação do Dados

- 1) Por defeito a aplicação utiliza um IQR *factor* de 3 na identificação de outliers, mas o utilizador pode ajustar o valor de acordo com a escala permitida. Os valores são recalculados em automático para os 10 métodos existentes.
- 2) Para simular a imputação de valores o utilizador tem que selecionar um dos modelos e selecionar o botão *Test*. Os dois gráficos são atualizados de acordo. Para aplicar na série original os valores da simulação o utilizador terá que selecionar o botão *Apply*. Uma vez imputados os valores estes passam a fazer parte da série original Figura 5.5.

### 5.1.3 Data Exploration – Exploração de Dados

Este menu apenas apresenta um conjunto de representações gráficas da série Figura 5.6. As separações são ajustadas, de forma automática, de acordo com a periodicidade da série e o utilizador não tem a necessidade de realizar nenhum tipo de configuração.



Figura 5.6. Data Exploration - Exploração de Dados

### 5.1.4 Decomposition – Decomposição STL

Esta funcionalidade da decomposição, apesar de estar separada da anterior, faz parte da etapa *Data Exploration*. A Figura 5.7 mostra o ecrã da decomposição e as funcionalidades são descritas de acordo com a respetiva legenda.



Figura 5.7. Decomposition - Decomposição STL

- 1) O utilizador pode seleccionar as componentes que pretende analisar, sendo que o gráfico interativo é atualizado automaticamente sempre que são alteradas as configurações.

- 2) O gráfico representa a intensidade da tendência e da sazonalidade.
- 3) O histograma representa a distribuição dos valores residuais.

### 5.1.5 Cross Validation – Treinar Modelos

Este menu permite treinar os modelos de previsão com base numa das 4 metodologias de validação cruzada descritas na subsecção 2.5.2. A Figura 5.8 corresponde ao écran inicial e as funcionalidades são descritas de acordo com a legenda.

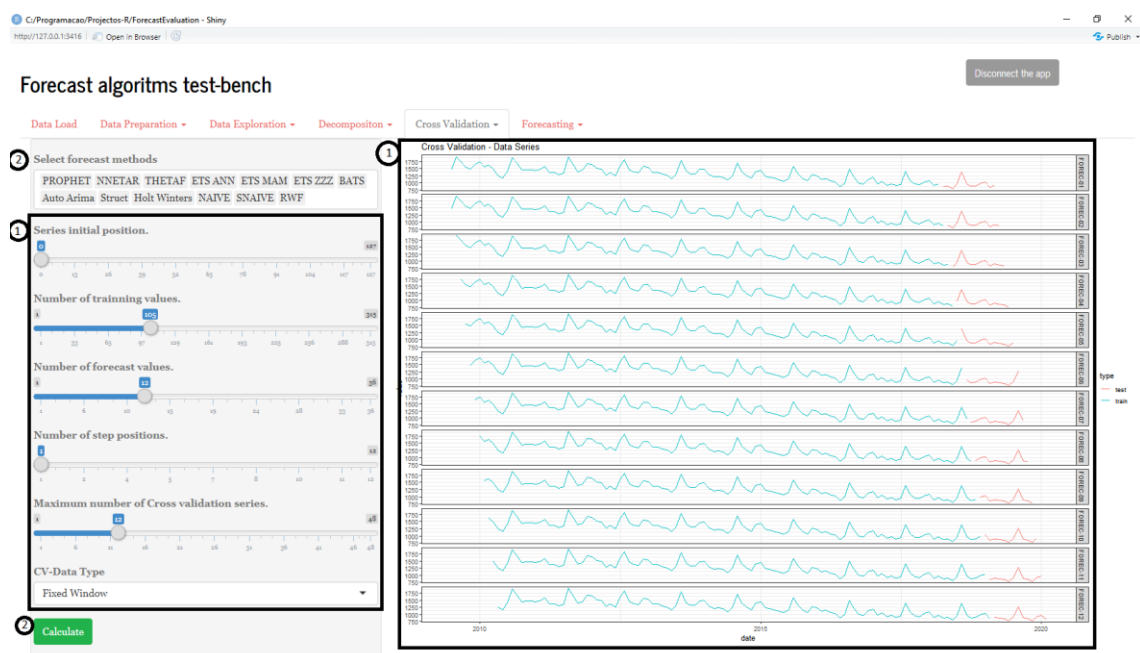


Figura 5.8. Cross Validation - Treinar Modelos (1)

- 1) O utilizador tem que configurar a validação cruzada indicando o seu tipo e um conjunto de parâmetros. A aplicação de forma automática recria as séries e apresenta os respetivos gráficos.
- 2) Para treinar os modelos o utilizador seleciona-os e força o cálculo com o botão *Calculate*. Para visualizar os resultados, o utilizador tem que selecionar a métrica de referência e o sistema apresenta os resultados em formato de tabela e graficamente, como mostra a Figura 5.9.



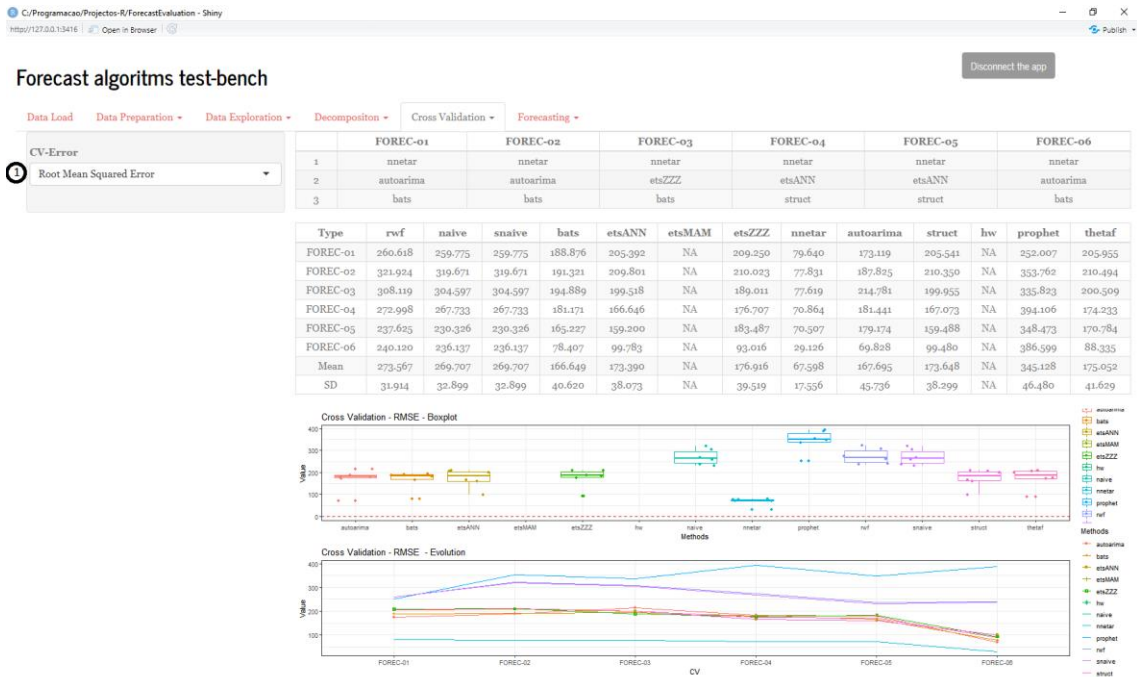


Figura 5.9. Cross Validation -Treinar Modelos (2)

### 5.1.6 Forecasting – Gerar Previsões

Este menu permite duas possibilidades de geração de previsões: A) Previsões pontuais para um conjunto de modelos; B) Previsões Intervalares.

A Figura 5.10 corresponde ao ecrã da Previsão Pontual e as funcionalidades são descritas de acordo com a legenda.

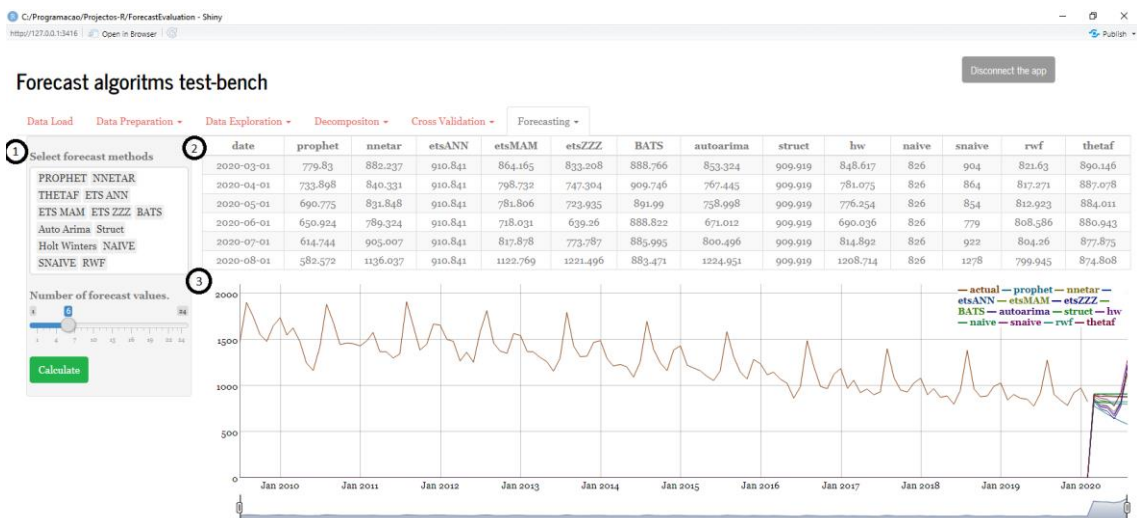


Figura 5.10. Forecasting - Gerar Previsões Pontuais

- 1) O utilizador tem que seleccionar os modelos e o horizonte temporal da previsão. A multiselecção é permitida e para as previsões serem geradas tem que se seleccionar o botão *Calculate*.
- 2) Os resultados das previsões são apresentados numa tabela para cada um dos modelos e para cada período temporal.
- 3) A série original e as previsões dos modelos são apresentadas num gráfico interativo.
- 4) A Figura 5.11 corresponde ao écran da Previsão intervalar e as funcionalidades são descritas de acordo com a legenda abaixo.

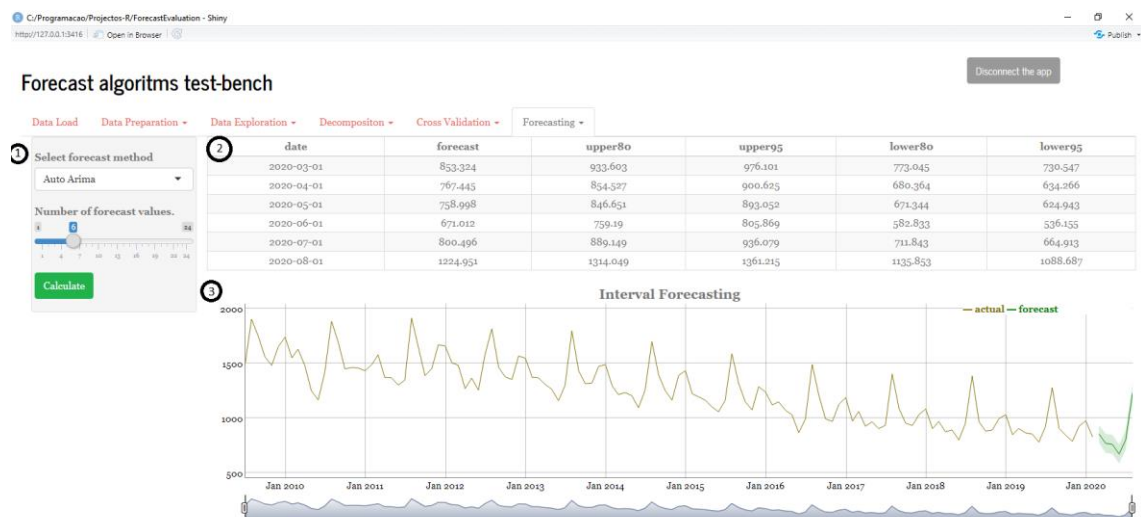


Figura 5.11. Forecasting - Gerar Previsões Intervalares

- 1) O utilizador tem que seleccionar um modelo e o horizonte temporal da previsão. A previsão é gerada após a seleção do botão *Calculate*.
- 2) Os diferentes valores da previsão intervalar (pontual, limites intervalo 80%, limites intervalo 95%) são apresentados numa tabela para cada período temporal.
- 3) A série original, a previsão temporal e os intervalos de 80% e 95% são apresentadas num gráfico interativo.



## 5.2 Demonstração

Nesta secção é feita uma apresentação detalhada das potencialidades das principais funcionalidades da aplicação, concretamente:

- Deteção de *outliers* e imputação de dados em falta (descrita na subsecção 5.2.1),
- Utilização da validação cruzada no treino e avaliação dos modelos de previsão (descrita na subsecção 5.2.2),
- Utilização da aplicação para identificar o melhor modelo de previsão para uma série específica e realizar as respetivas previsões (descrita na secção 5.1).

### 5.2.1 Imputação de dados

No menu *Data Preparation* são disponibilizados 10 modelos para a imputação de dados em falta (descritos na secção 2.3.2). Para ajudar na avaliação e escolha é disponibilizada uma funcionalidade que permite fazer a contabilização dos *outliers* que resultam da aplicação de cada um dos modelos (descrita na secção 2.3.1).

Para fazer a demonstração da combinação imputação-*outlier*, foi utilizada uma série de dados, criadas a partir da série *AirPassengers*, disponibilizada pelo *package datasets*, à qual foram retiradas aleatoriamente 20% das observações<sup>10</sup>. A escolha desta série de dados resulta do facto de originalmente não ter falhas e de apresentar clara tendência e sazonalidade. Para a demonstração foram utilizados todos os modelos de imputação e para a comparação dos erros de imputação<sup>11</sup> foram utilizadas as métricas MAE e MAPE. Os *outliers* foram contabilizados para três IQR *factors* distintos (1, 2 e 3).

Para ajudar na avaliação dos modelos os seus resultados foram ordenados de acordo com os respetivos erros e números de *outliers* e foram também utilizadas as representações gráficas de cada modelo para complementar a análise quantitativa com uma análise qualitativa.

---

<sup>10</sup> Para reprodutibilidade, o código que permite obter estas três séries é apresentado no APÊNDICE-7.1

<sup>11</sup> Erro de imputação = (Valor Real – Valor Imputado)

### 5.2.1.1 Série

A série utilizada na demonstração apresenta 28 falhas algumas das quais sucessivas (ver Figura 5.12).

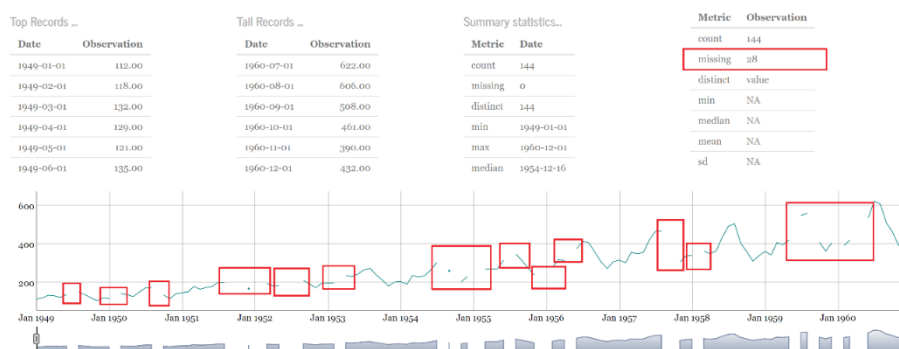


Figura 5.12. Série – Retiradas 20% das observações

### 5.2.1.2 Imputação

Depois de aplicados os modelos, os seus resultados foram compilados em duas tabelas. Tabela 5.1 contém os dados dos erros de imputação de cada modelo calculados com base nas métricas MAE e MAPE e a Tabela 5.2 contém os *outliers* contabilizados em cada modelo para os três IQR *factor* considerados (os resultados detalhados podem ser consultados no apêndice 7.2).

Tabela 5.1 - Resultados – Métricas MAE+MAPE

ID	Modelo	Valores		Rankings	
		MAE	MAPE	MAE	MAPE
M01	Linear Interpolation	17,012	6,614	2	2
M02	Spline Interpolation	16,530	6,262	1	1
M03	Stineman Interpolation	17,418	6,737	7	7
M04	Kalman StructTS	35,479	12,972	8	8
M05	Kalman StructTS Smooth	17,012	6,614	2	2
M06	Kalman Space Models	17,012	6,614	2	2
M07	Seasonally Decomposed - Interpolation	17,012	6,614	2	2
M08	Seasonally Decomposed - Mean	91,179	38,138	10	10
M09	Seasonally Decomposed - Kalman	17,012	6,614	2	2
M10	MICE	41,107	14,929	9	9

Considerando os erros de imputação apresentados na Tabela 5.1 verifica-se que:

- O modelo M02 corresponde ao modelo que tem menos erros independentemente da métrica que seja considerada,
- Os modelos M01/M05/M06/M07/M09 apresentam os mesmos resultados,
- O modelo M03, apesar de classificado na 7 posição, apresenta valores muito próximos dos melhores modelos,
- Com a exceção dos modelos M04/M08/M10 todos têm um comportamento muito semelhante.

Tabela 5.2- Resultados – IQR Factor(1,2,3)

ID	Modelo	Outliers – IQR Factor			Rankings		
		1	2	3	1	2	3
M01	Linear Interpolation	19	9	5	1	1	1
M02	Spline Interpolation	23	12	8	8	6	8
M03	Stineman Interpolation	22	12	6	7	6	7
M04	Kalman StructTS	24	16	13	9	8	9
M05	Kalman StructTS Smooth	19	9	5	1	1	1
M06	Kalman Space Models	19	9	5	1	1	1
M07	Seasonally Decomposed - Interpolation	19	9	5	1	1	1
M08	Seasonally Decomposed - Mean	28	21	14	10	10	10
M09	Seasonally Decomposed - Kalman	19	9	5	1	1	1
M10	MICE	20	16	5	6	8	1

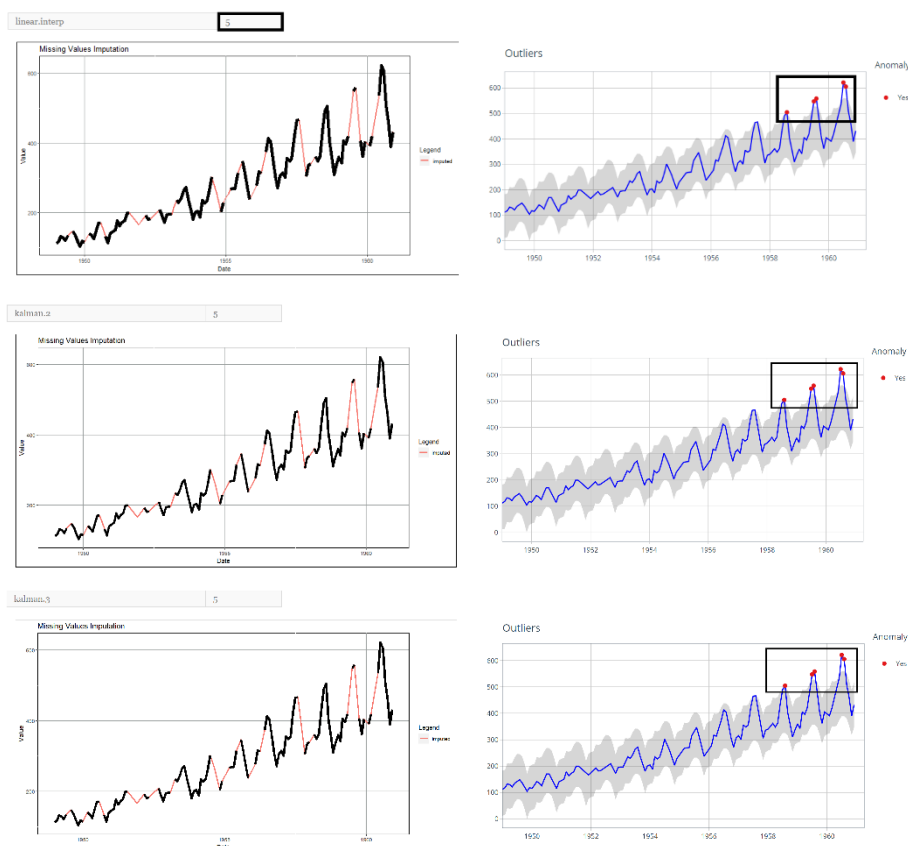
Considerando os *outliers* apresentados na Tabela 5.2 verifica-se que:

- Os modelos M01/M05/M06/M07/M09 apresentam o mesmo número de *outliers* independentemente do IQR *factor* utilizado,
- Os modelos M01/M05/M06/M07/M09 apresentam sempre o menor número de *outliers*,
- O modelo M10 para um IQR *factor* de 3 também consegue a 1ª posição em conjunto com os modelos M01/M05/M06/M07/M09,

- Os modelos M02/M03/M10 são os modelos em que a alteração do IQR *factor* tem maior impacto na sua classificação.

Fazendo a análise dos gráficos da imputação, apenas para um IQR *factor* 3 verifica-se que:

- A imputação efetuada pelos modelos M01/M05/M06/M07/M09 é muito semelhante e parece seguir o padrão de evolução da série Figura 5.13,
- O modelo M04 apesar de apresentar um elevado número de *outliers* no entanto parece seguir o padrão de evolução da série Figura 5.14.
- O modelo M08 é claramente aquele que pior identifica o padrão de evolução da série Figura 5.15.



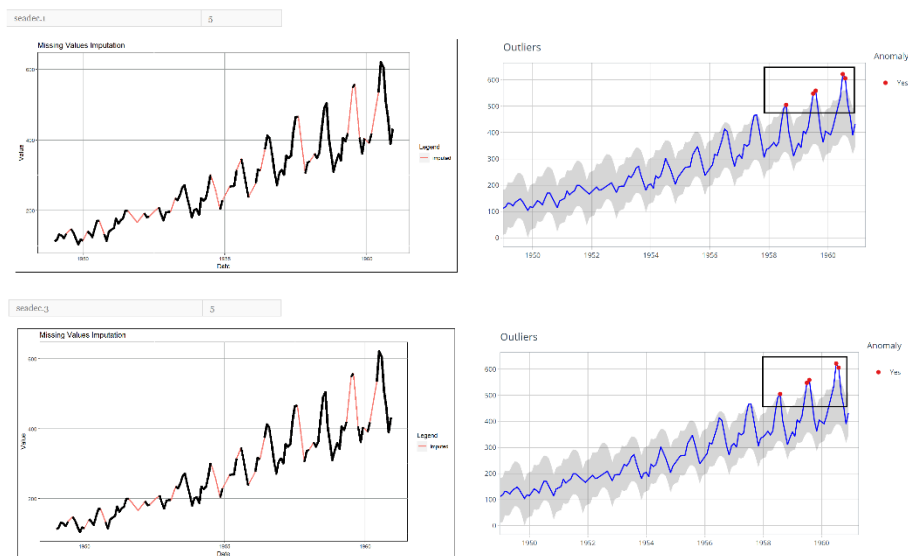


Figura 5.13. Imputação - IQR factor 3 - Modelos M01/M05/M06/M07/M09

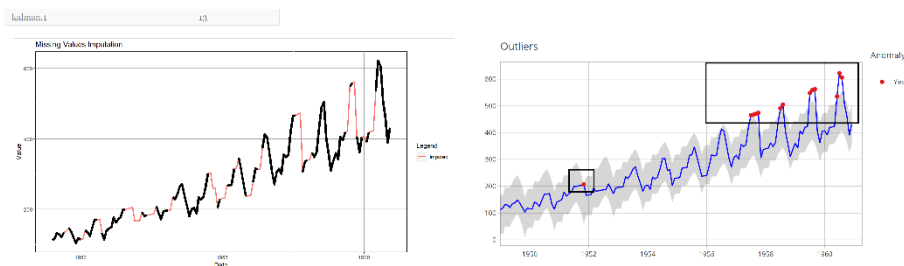


Figura 5.14. Imputação - IQR factor 3 - Modelos M04

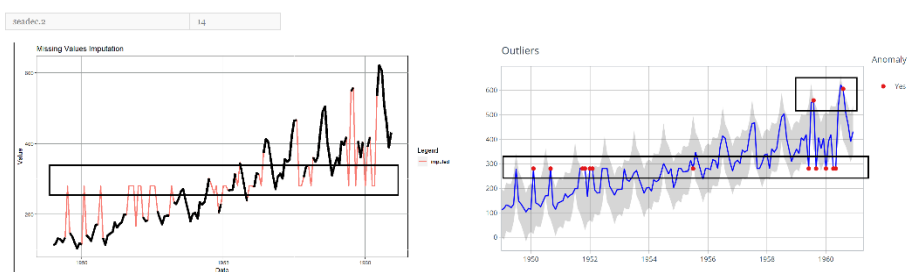


Figura 5.15. Imputação - IQR factor 3 - Modelos M08

### 5.2.1.3 Conclusão

Com base nesta demonstração verifica-se não existir uma correspondência entre um maior número de *outliers* e uma maior percentagem de erros pelo que, na identificação da melhor metodologia, esta informação deve ser complementada com a análise gráfica de cada modelo.

## 5.2.2 Validações cruzadas

Para efetuar a demonstração da validação cruzada foram utilizados os dados da série *AirPassengers*, os modelos: **rwf (M1) / naïve (M2) / snaive (M3) / bats (M4) / etsANN (M5) / etsZZZ (M6) / nnetar (M7) / autoarima (M8) / struct (M9) / prophet (M10) / thetaf (M11)**. Foram efetuadas previsões para horizontes temporais a 12 meses (Mês+1/.../Mês+12) e contabilizados os erros utilizando a métrica RMSE (descritos na subsecção 2.5.3).

### 5.2.2.1 Hold-out

A aplicação permite configurar a validação cruzada de forma a ser possível a realização de avaliação com base na metodologia *hold-out* (descrita na subsecção 2.5.1) A título de exemplo apresenta-se, de seguida, a configuração necessária para a realização de avaliação com base nas últimas 60 observações da série em que se pretende que as primeiras 48 sejam utilizadas no treino e as restantes 12 para teste.

Critérios (Figura 5.16, quadro-1):

- Series initial position = 85 (série tem um total de 144 registos),
- Number of training values=48,
- Number of forecast values=12,
- Number of steps positions=1,
- Maximum number of Cross validation series=1,
- CV- Data Type=Fixed Window.

A série é apresentada e atualizada automaticamente sempre que são efetuadas alterações às configurações (Figura 5.16, quadro-2).

Forecast algorithms test-bench

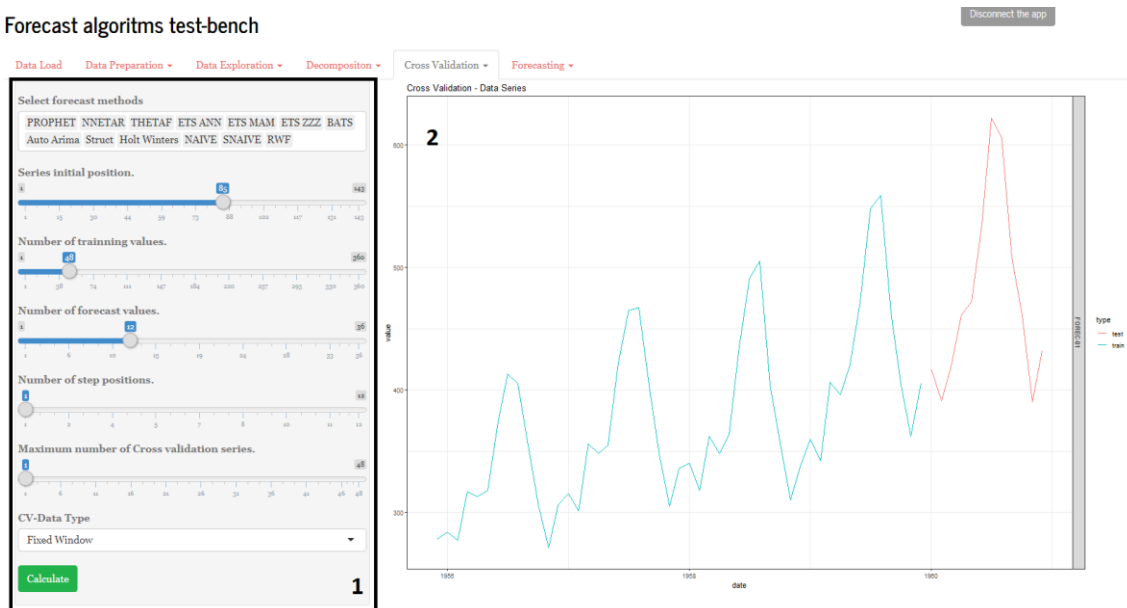


Figura 5.16. CV - OSS – Configuração

### 5.2.2.2 Fixed Window

A validação cruzada denominada por *Fixed Window*, que está descrita na subsecção 2.5.2, cria conjuntos de séries de dimensão fixa. A título de exemplo apresenta-se, de seguida, a configuração necessária para a realização de avaliação com base em 12 séries de dimensão em que as primeiras 48 observações são utilizadas no treino e as restantes 12 para teste.

CrITÉrios (Figura 5.17, quadro-1):

- Series initial position = 1,
- Number of training values=48,
- Number of forecast values=12,
- Number of steps positions=1,
- Maximum number of Cross validation series=12,
- CV- Data Type=Fixed Window.

A série é apresentada e atualizada automaticamente sempre que são efetuadas alterações às configurações (Figura 5.17, quadro-2).

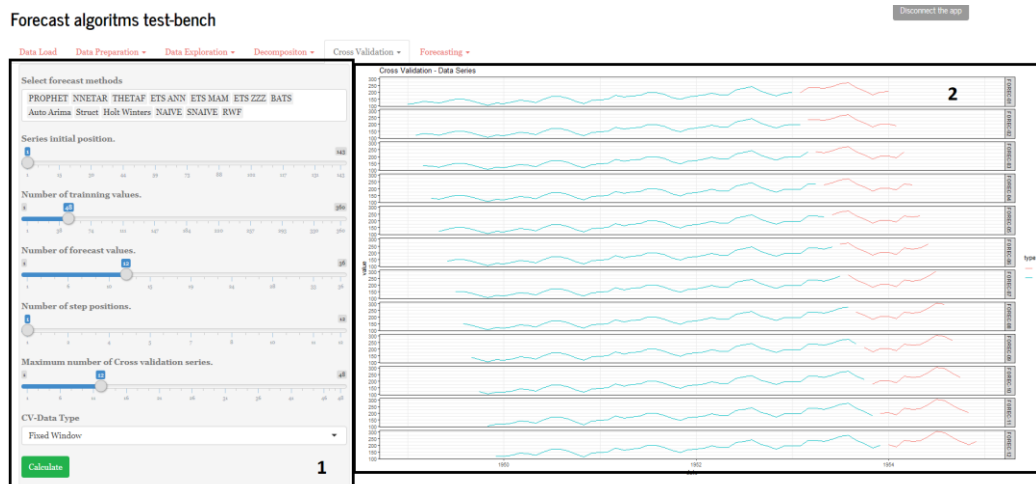


Figura 5.17. CV - Fixed Window – Configuração

### 5.2.2.3 Rolling Window

A validação cruzada denominada por *Rolling Window*, que está descrita na subsecção 2.5.2, cria conjuntos de séries de dimensão crescente. A título de exemplo apresenta-se, de seguida, a configuração necessária para a realização de avaliação com base em 24 series de dimensão em que as primeiras 36 observações são utilizadas no treino e as restantes 6 para teste.

Crítérios (Figura 5.18, quadro-1):

- Series initial position = 1,
- Number of training values=36,
- Number of forecast values=6,
- Number of steps positions=1,
- Maximum number of Cross validation series=24,
- CV- Data Type=Rolling Window.

A série é apresentada e atualizada automaticamente sempre que são efetuadas alterações às configurações (Figura 5.18, quadro-2).



Forecast algorithms test-bench

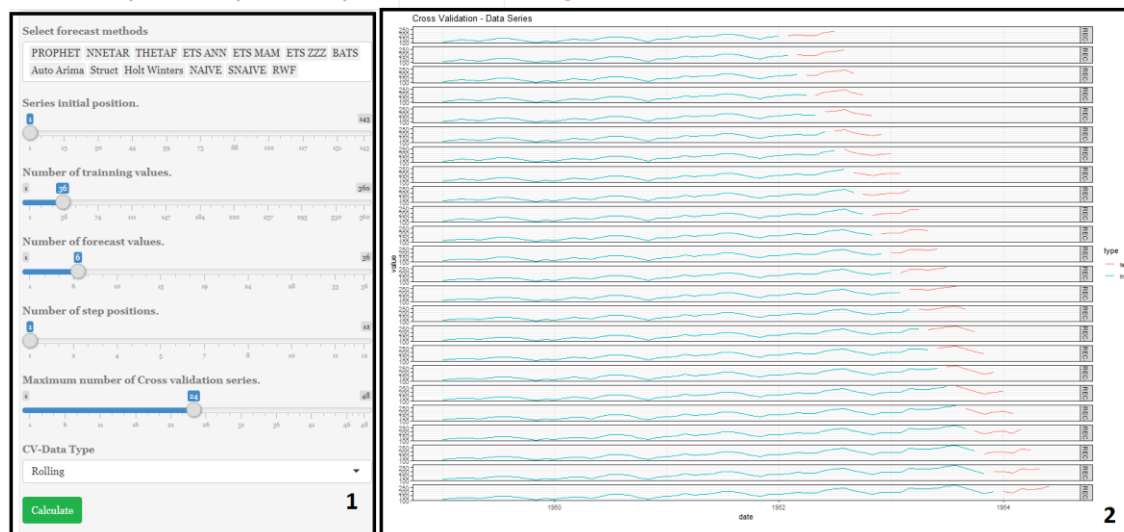


Figura 5.18. CV - Rolling Window – Configuração

5.2.2.4 Avaliação dos modelos

A avaliação dos modelos é independente da metodologia aplicada. Para ajudar na avaliação, são disponibilizados gráficos e tabelas que de seguida se descrevem.

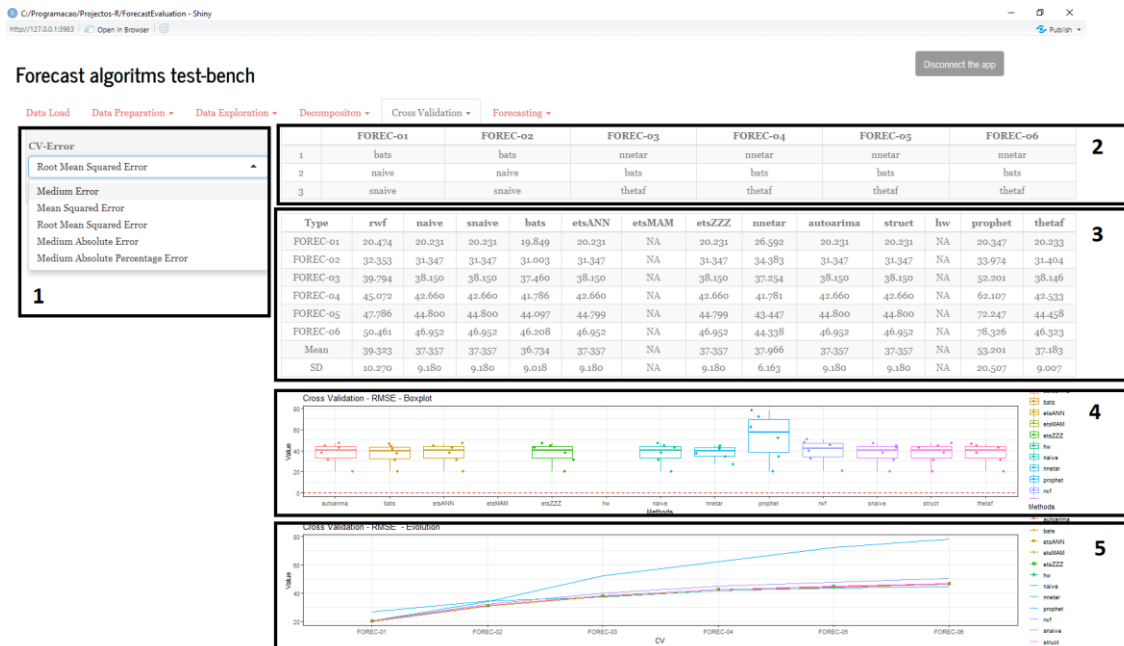


Figura 5.19. CV - Submenu - Error analysis

A Figura 5.19 corresponde ao submenu *Error Analysis* onde o utilizador tem a possibilidade de escolher a métrica com que pretende fazer a avaliação dos modelos (quadro-1) e de seguida verificar os respetivos resultados:

- Tabela *Ranking* dos Modelos – Para cada horizonte temporal (FOREC-1,...) é apresentado um ranking com os três melhores modelos (quadro-2),
- Tabela Erros Modelos – Para cada horizonte temporal e para cada modelo, são apresentados os respetivos erros de previsão (quadro-3),
- Gráfico *Boxplot Erros* – Para cada modelo são representados os erros de previsão que constam da tabela anterior (quadro-3) num gráfico do tipo *boxplot* (quadro-4),
- Gráfico *Evolution* – Para cada modelo é apresentada a evolução dos erros de previsão ao longo do horizonte temporal em análise (quadro-4).

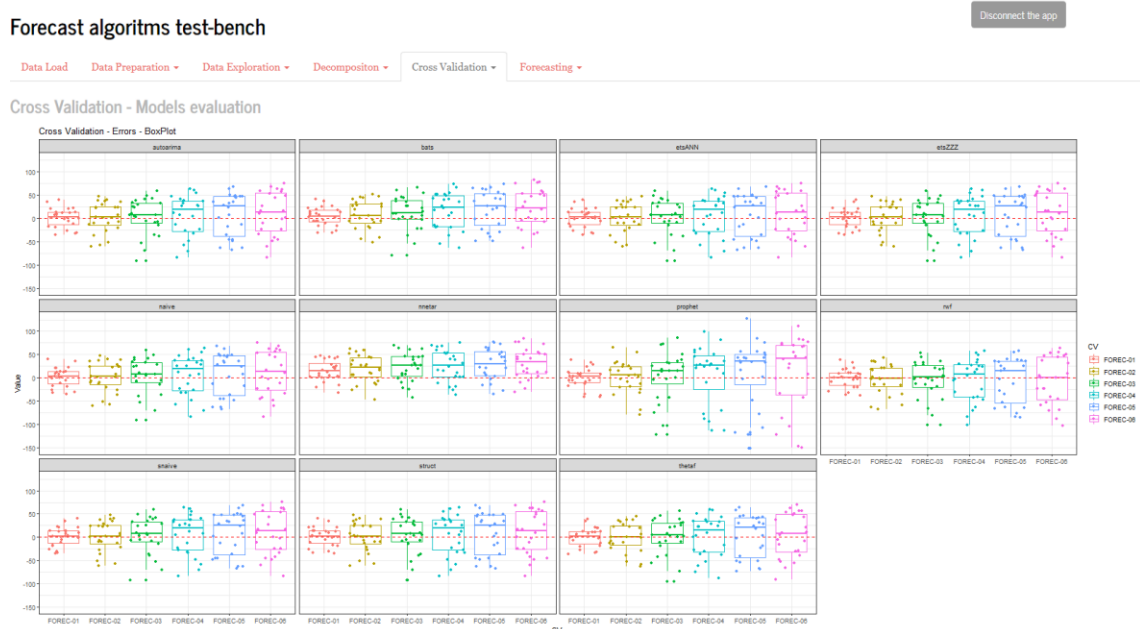


Figura 5.20. CV - Submenu - Error analysis Plots

A Figura 5.20 corresponde ao submenu *Error analysis Plots* que contém gráficos do tipo *boxplot* em que é representada, por modelo e horizonte de previsão, a dispersão dos erros calculados.

Forecast algorithms test-bench

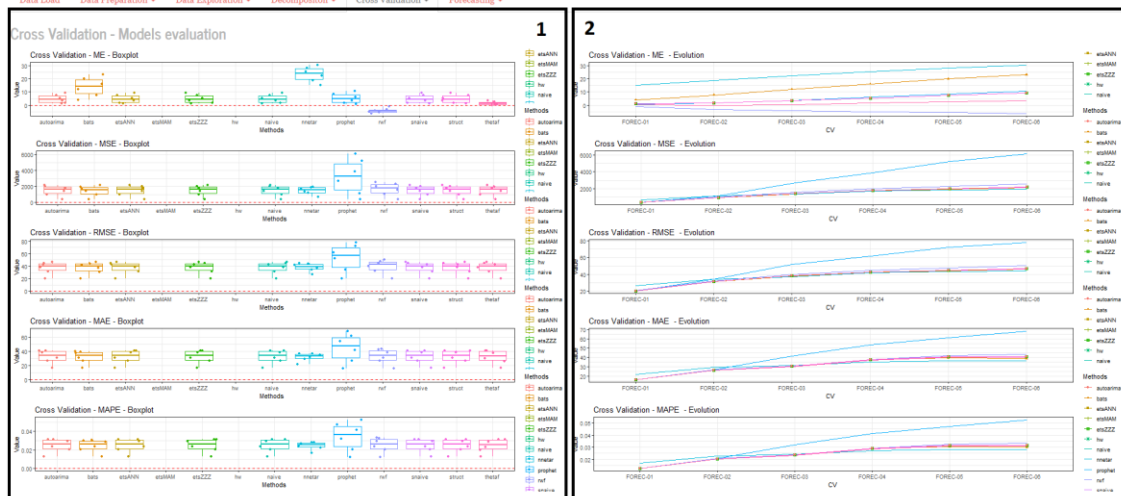


Figura 5.21. CV - Submenu - Error Plots

A Figura 5.21 corresponde ao submenu *Error Plots* onde são apresentados dois conjuntos de gráficos. O primeiro conjunto, do tipo *boxplot*, apresentada, por modelo e métrica de avaliação, os erros por horizonte temporal (quadro-1). O segundo conjunto de gráficos, apresenta a evolução dos erros de previsão ao longo do horizonte temporal em análise (quadro-2).

### 5.2.3 Avaliação de modelos e geração de previsões

Para efetuar a demonstração das capacidades de avaliação de modelos e de geração de previsões foi escolhida a série *AirPassengers*. Esta série tem um total de 144 valores mensais correspondentes ao número de passageiros de companhias aéreas nos anos de 1949 a 1960 e nesta demonstração vamos, com base nas funcionalidades da aplicação, determinar quais os melhores modelos de previsão e no final utilizar esses mesmos modelos para realizar uma previsão pontual e intervalar para um horizonte temporal de doze meses.

### 5.2.3.1 Data load

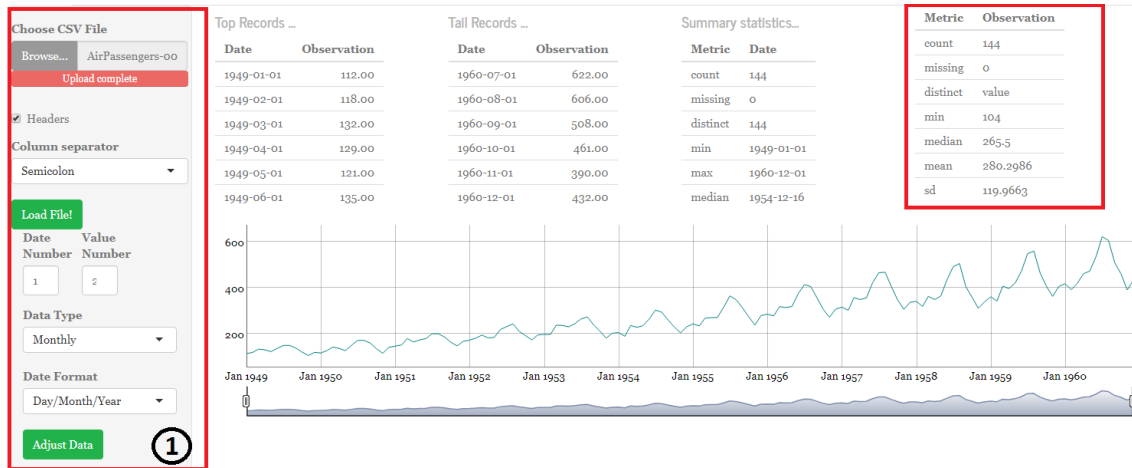


Figura 5.22. Data Load

A Figura 5.22 corresponde ao ecrã inicial da aplicação depois de se ter carregados os dados da série. Inicialmente carregou-se o ficheiro CSV que continha os dados e de seguida foram feitas as configurações necessárias para a série ser devidamente carregada (quadro-1). Verifica-se a existência de 144 observações, a inexistência de falhas, uma tendência crescente e uma sazonalidade anual.

### 5.2.3.2 Data Preparation

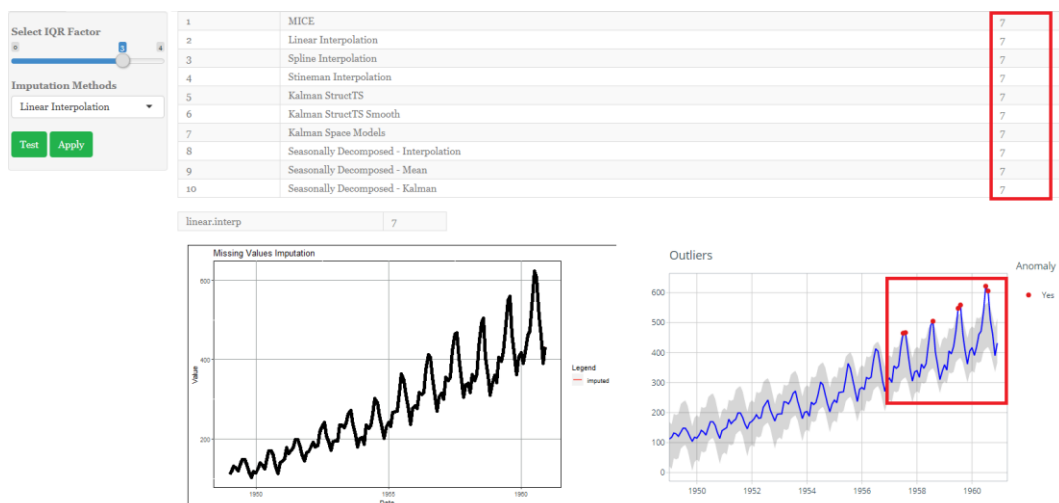


Figura 5.23. Data Preparation

A Figura 5.23 corresponde ao submenu *Data Preparation* e, apesar de não existirem falhas de dados e como tal não ser necessária a utilização da funcionalidade da imputação

de dados, permite, no entanto, pela análise dos *outliers*, perceber que, a evolução da série nos meses de julho e agosto dos últimos 4 anos está a ter um comportamento fora da normalidade.

### 5.2.3.3 Data Exploration

Os gráficos disponibilizados nos submenus do *Data Exploration* permitem confirmar as características da sazonalidade da série (ver Figura 5.24 e Figura 5.25) e a tendência de evolução crescente (ver Figura 5.26 e Figura 5.27).

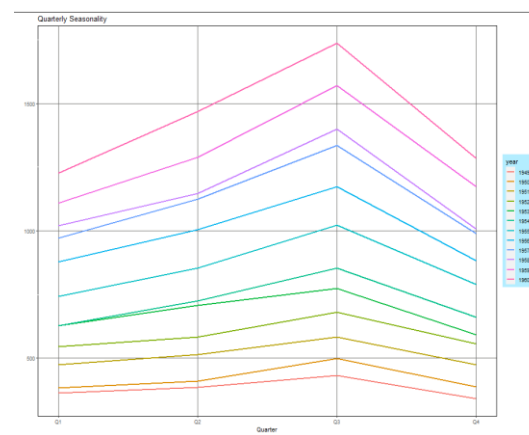
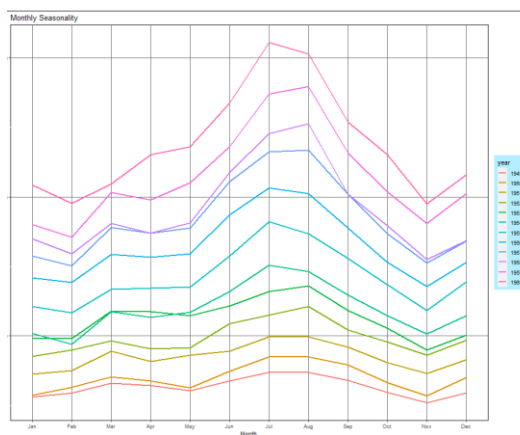


Figura 5.24. Sazonalidade – Periodicidade Mensal      Figura 5.25. Sazonalidade - Periodicidade Trimestral

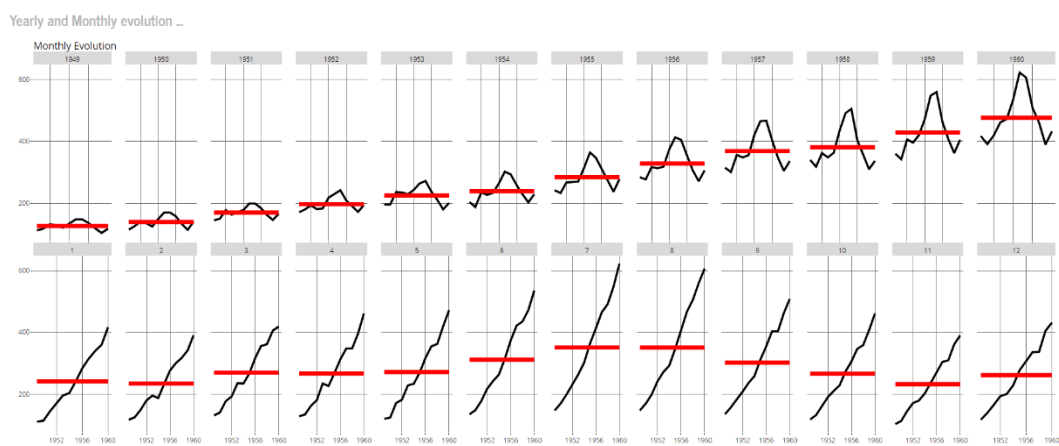


Figura 5.26. Subséries Sazonais

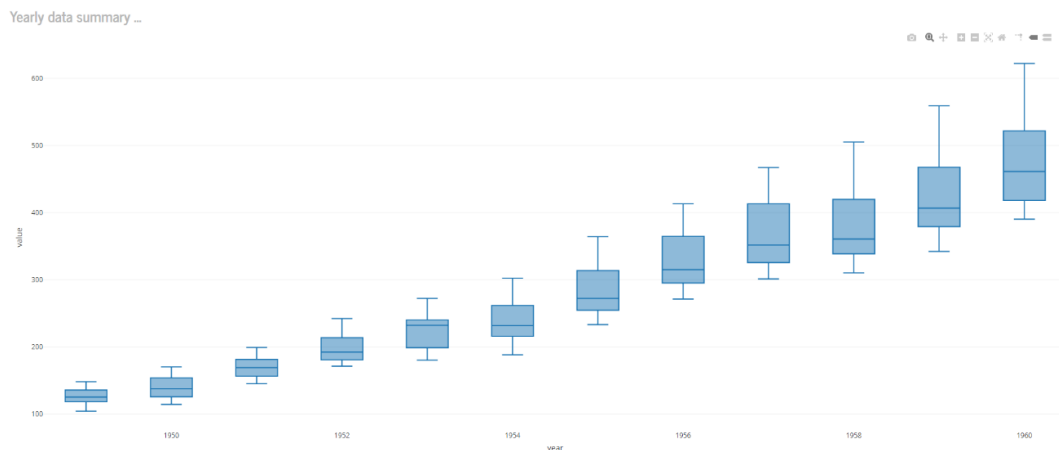


Figura 5.27. Evolução Anual

### 5.2.3.4 Decomposition

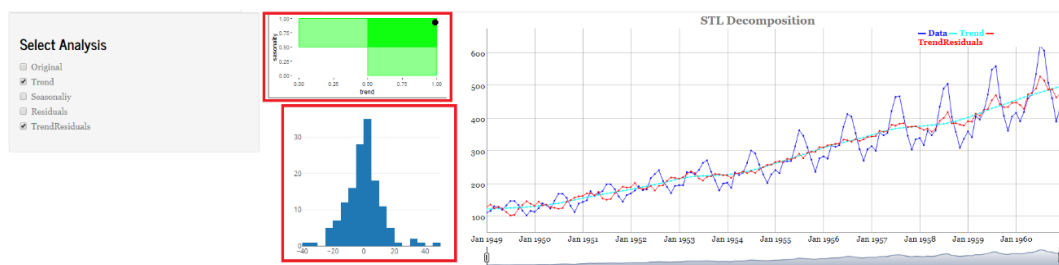


Figura 5.28. Decomposição da série

A Figura 5.28 corresponde ao menu *Decomposition* onde o utilizador pode analisar a decomposição da série nas suas diferentes componentes. Neste caso específico, permite confirmar a existência de uma forte tendência e sazonalidade (gráfico do canto superior esquerdo), algo que também é visível no gráfico da decomposição STL da série. O histograma dos resíduos permite ainda concluir que estes seguem uma distribuição normal.

### 5.2.3.5 Cross Validation

Nesta demonstração a validação cruzada foi efetuada utilizando 12 séries de dimensão fixa em que para o treino foram utilizados 5 anos de informação (60 observações) e para o teste um ano (12 observações), como se mostra na Figura 5.29.

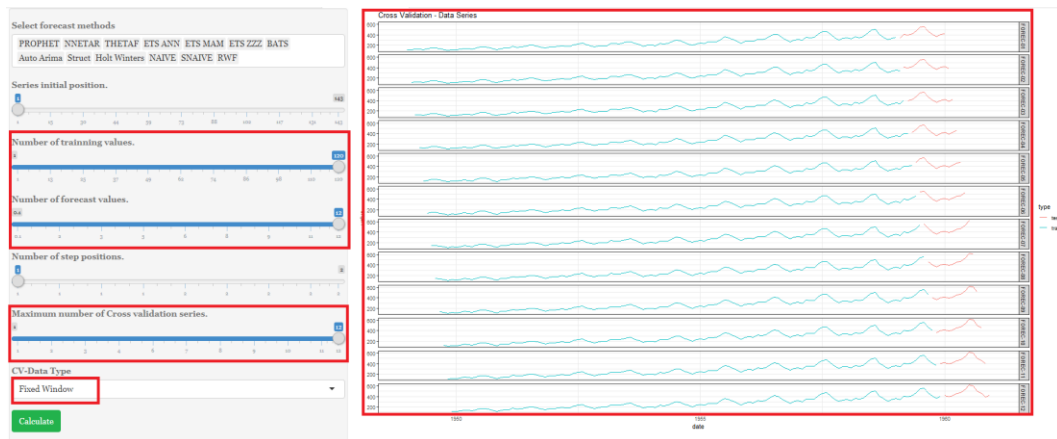


Figura 5.29. Validação cruzada – Configuração

Os resultados obtidos com base na métrica **RMSE** identificam o modelo **NNETAR** como o melhor modelo para horizontes de previsão de 1 a 11 meses (FOREC-01,...,FOREC-11) e o modelo **RWF** para previsões a 12 meses (FOREC-12).



Figura 5.30. Validação cruzada - Resumo Modelos

Considerando os gráficos *boxplot* da Figura 5.30 e Figura 5.31, verifica-se que existe uma clara diferença entre a performance do modelo **NNETAR** e os restantes. Esta diferença corresponde a um valor médio que é, pelo menos, metade dos restantes e a um desvio padrão que é, no mínimo, 4 vezes inferior Figura 5.30. Esta melhor performance pode-se também confirmar no gráfico da Figura 5.31, em que é visível a sua menor variabilidade ao longo das previsões.



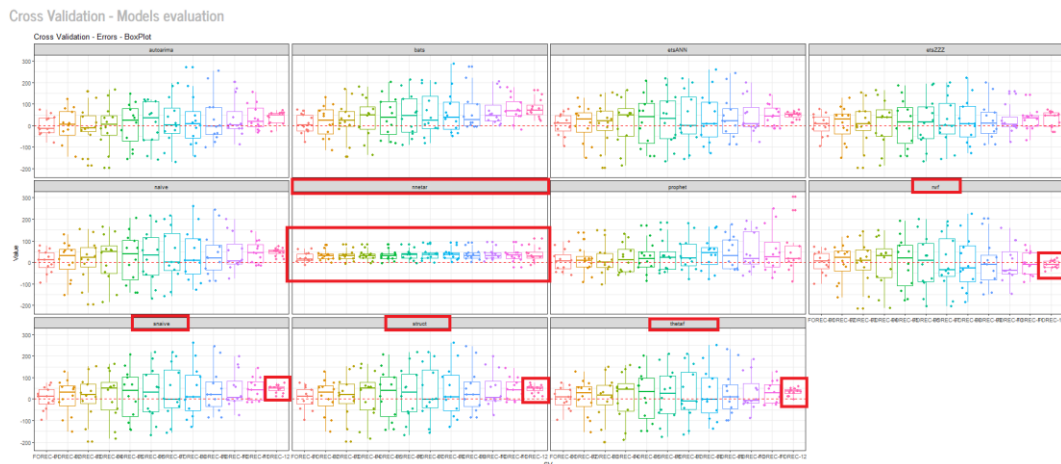


Figura 5.31. Validação cruzada - Avaliação dos Modelos

### 5.2.3.6 Forecasting – Previsão pontual

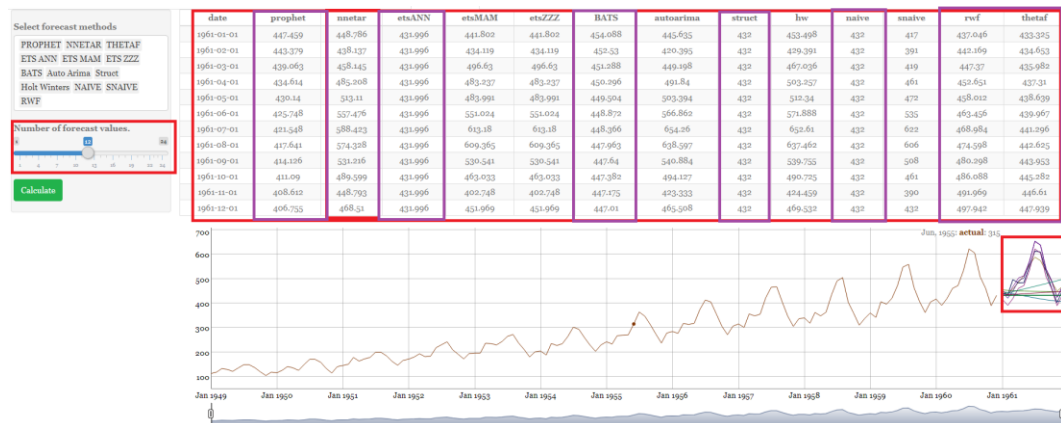


Figura 5.32. AirPassenger - Previsão Pontual – Global

A Figura 5.32 apresenta a simulação de previsões pontuais a 12 meses para todos os modelos e mostra uma separação clara entre aqueles que replicam bem o padrão da série e os restantes. Uma constatação interessante resulta da avaliação das previsões dos modelos **Prophet** e **BATS** que em termos de ranking se encontravam bem posicionados, mas que pela análise do gráfico se verifica que não seguem o padrão da série. Pela positiva destacam-se os modelos **NNETAR / ETS-MAM / ETS-ZZZ / AutoArima / Holt Winters / Snaive**, pela negativa os modelos **Prophet / THETAF / ETS ANN / BATS / Struct, NAIVE/ RWF**.



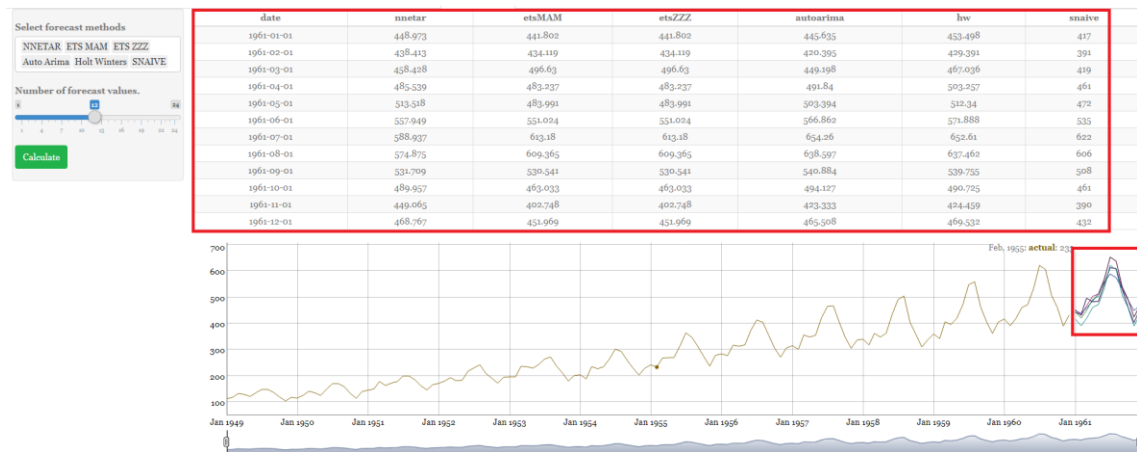


Figura 5.33. Previsão Pontual - Melhores Modelos

A Figura 5.33 representa a previsão pontual apenas dos melhores modelos de previsão **NNETAR / ETS-MAM / ETS-ZZZ / AutoArima / Holt Winters / Snaive**.

### 5.2.3.7 Forecasting – Previsão intervalar

O modelo **NNETAR** apesar de ter sido o melhor classificado nas avaliações anteriores não pôde ser incluído na previsão intervalar por não ter essa possibilidade implementada. A demonstração foi efetuada apenas com os modelos: **Auto Arima / ETS MAM / ETS ZZZ / Holt Winters / Snaive**.

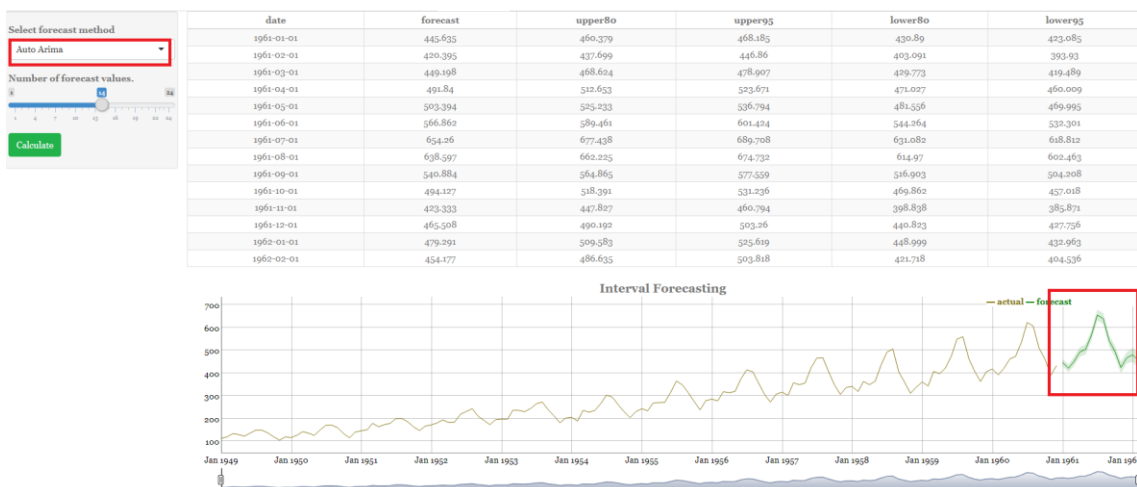


Figura 5.34. Previsão Intervalar - Auto Arima

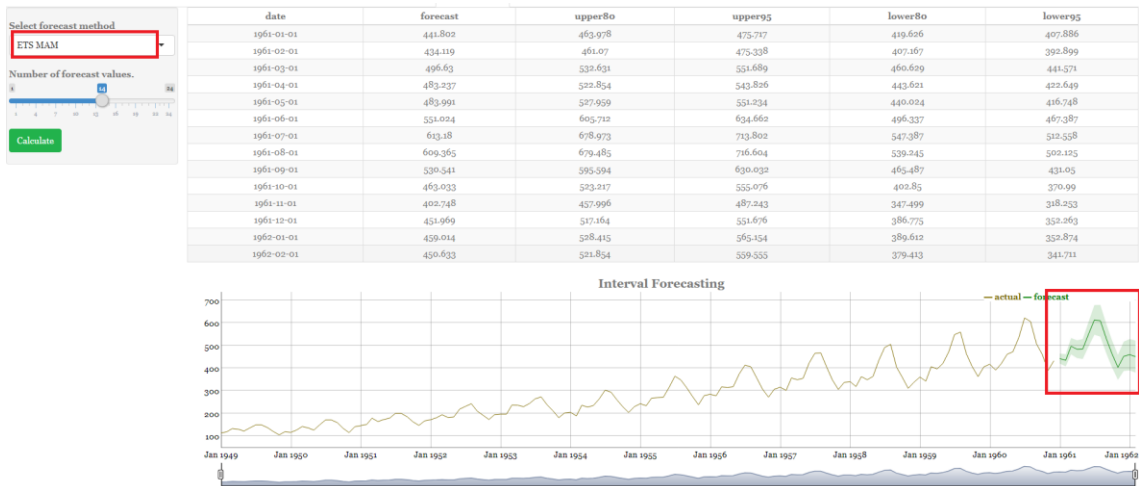


Figura 5.35. Previsão Intervalar - ETS MAM

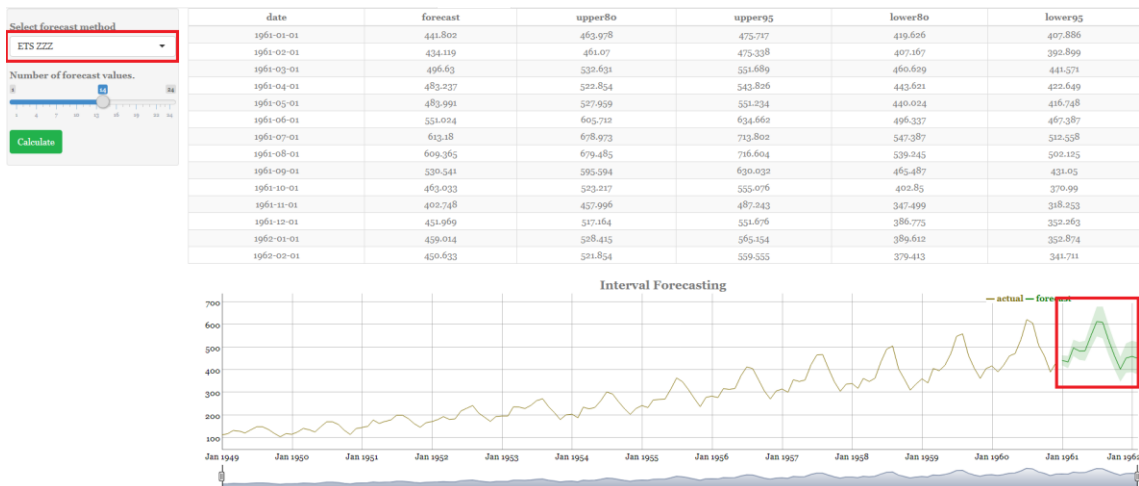


Figura 5.36. Previsão Intervalar - ETS ZZZ

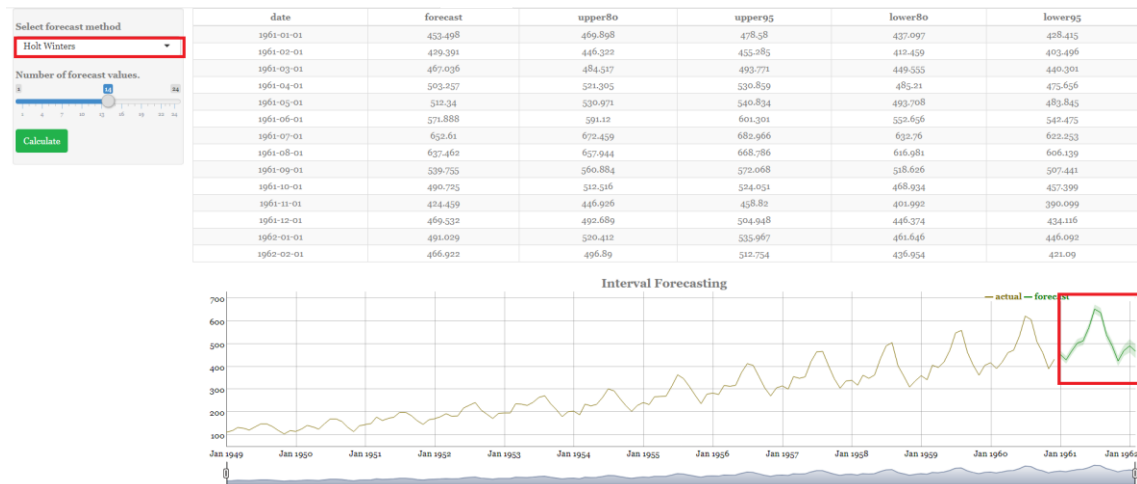


Figura 5.37. Previsão Intervalar - Holt Winters

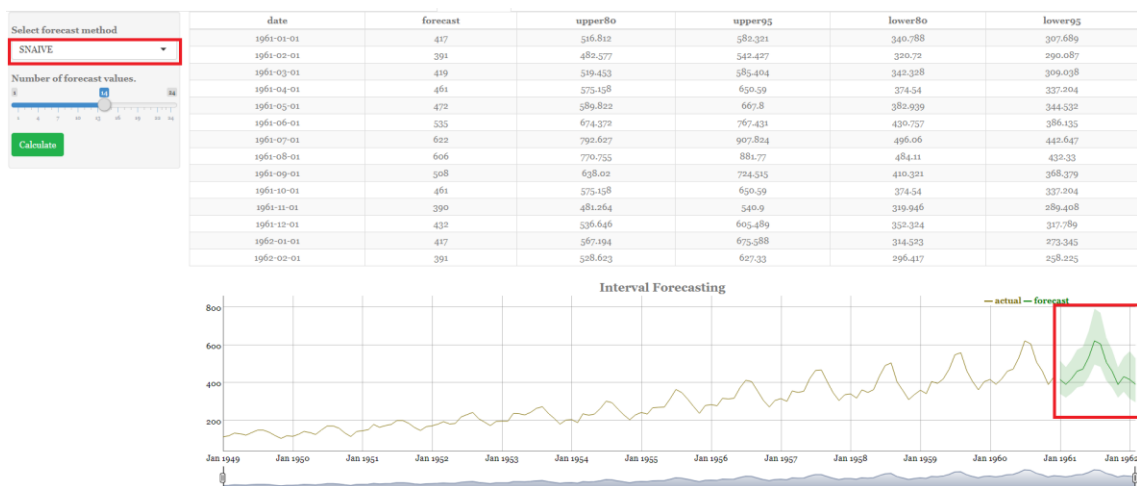


Figura 5.38. Previsão Intervalar – Snaive

Da análise dos gráficos, verifica-se que todos seguem o padrão da série, no entanto, os modelos **Auto Arima** (Figura 5.34) e **Holt Winters** (Figura 5.37), destacam-se dos restantes **ETS MAM** (Figura 5.35), **ETS ZZZ** (Figura 5.36) e **Snaive** (Figura 5.38) por terem intervalos de previsão de menor amplitude.

## 6 CONCLUSÃO

Este projeto resulta de um esforço contínuo de oito meses de trabalho que incluiu, a revisão de literatura, a avaliação de vários pacotes R, a formação em programação R, o desenvolvimento e teste de dois protótipos, desenvolvimento e testes da aplicação final e por último a elaboração deste relatório. Durante o projeto foram realizadas reuniões semanais de acompanhamento o que permitiu a sua conclusão dentro dos prazos definidos.

Resulta deste projeto uma aplicação *Web* que permite a qualquer utilizador efetuar uma avaliação de modelos e gerar previsões em séries temporais univariadas. A aplicação permite uma grande flexibilidade na sua utilização e, para além das tarefas normais de um processo de previsão, acrescenta: a capacidade de identificação de *outliers*, a imputação de valores em falta, a utilização da validação cruzada na avaliação dos modelos, a geração de previsões pontuais e intervalares. Automatiza uma parte significativa das tarefas e apresenta os resultados de uma forma simples e intuitiva o que a torna de fácil utilização para utilizadores não proficientes na temática.

A inexistência de consensos relativamente à identificação das melhores métricas de avaliações dos modelos condicionou a implementação de alguns dos automatismos que se pretendiam, tais como: escolha da melhor metodologia na imputação de dados e a escolha dos melhores modelos de previsão. Não sendo possível automatizar estes processos, optou-se pela inclusão de um número alargado de opções o que, por um lado flexibiliza a sua utilização, mas por outro, complica a análise a utilizadores menos informados na matéria.

Uma aplicação informática nunca está terminada e esta não é a exceção. A evolução dos pacotes utilizados, a evolução e disponibilização de novos modelos de previsão, a adoção por parte da comunidade científica de novas métricas de avaliação só por si serão justificação para futuras versões. Por outro lado, um dos objetivos iniciais do projeto seria esta atingir um estágio de maturidade que permitisse a sua submissão a um dos repositórios de pacotes de R. Não tendo sido possível ser efetuado durante o período do projeto esta submissão, será seguramente tentada no futuro.

## REFERÊNCIAS

- Bergmeir, C., Hyndman, R. J., & Koo, B. (2018). A note on the validity of cross-validation for evaluating autoregressive time series prediction. *Computational Statistics & Data Analysis*, 120, 70–83. <https://doi.org/https://doi.org/10.1016/j.csda.2017.11.003>
- Beynon-Davies, P., Came, C., Mackay, H., & Tudhope, D. (1999). Rapid application development (Rad): An empirical review. *European Journal of Information Systems*, 8(3), 211–232. <https://doi.org/10.1057/palgrave.ejis.3000325>
- Beynon-Davies, Paul, & Holmes, S. (2002). Design breakdowns, scenarios and rapid application development. *Information and Software Technology*, 44(10), 579–592. [https://doi.org/10.1016/S0950-5849\(02\)00078-2](https://doi.org/10.1016/S0950-5849(02)00078-2)
- Bokde, N. D., Yaseen, Z. M., & Andersen, G. B. (2020). ForecastTB-an R package as a test-bench for time series forecasting-application of wind speed and solar radiation modeling. *Energies*, 13(10). <https://doi.org/10.3390/en13102578>
- Brockwell, Peter J. ; Davis, R. A. (2002). *Introduction to Time Series and Forecasting* (2nd ed.). Springer-Verlag New York.
- Cerqueira, V., Torgo, L., & Mozetič, I. (2020). Evaluating time series forecasting models: an empirical study on performance estimation methods. In *Machine Learning* (Vol. 109, Issue 11). Springer US. <https://doi.org/10.1007/s10994-020-05910-7>
- Chen, C., Twycross, J., & Garibaldi, J. M. (2017). A new accuracy measure based on bounded relative error for time series forecasting. *PLoS ONE*, 12(3), 1–23. <https://doi.org/10.1371/journal.pone.0174202>
- Cunha, A., Gonçalves, P., Barreira, J., Trigo, A., & Hughes, S. J. (2015). Mobile RHS: A Mobile Application to Support the “river Habitat Survey” Methodology. *Procedia Computer Science*, 64, 87–94. <https://doi.org/10.1016/j.procs.2015.08.467>
- de Livera, A. M., Hyndman, R. J., & Snyder, R. D. (2011). Forecasting time series with complex seasonal patterns using exponential smoothing. *Journal of the American Statistical Association*, 106(496), 1513–1527. <https://doi.org/10.1198/jasa.2011.tm09771>
- Despa, M. L. (2014). Comparative study on software development methodologies. In *Database Systems Journal* (Issue 3).
- Ferbar Tratar, L., & Strmčnik, E. (2016). The comparison of Holt-Winters method and Multiple regression method: A case study. *Energy*, 109, 266–276. <https://doi.org/10.1016/j.energy.2016.04.115>
- Harvey, A., & Koopman, S. J. (1996). Structural time series models in medicine. *Statistical Methods in Medical Research*, 5(1), 23–49. <https://doi.org/10.1177/096228029600500103>
- Holt, C. C. (2004). Forecasting seasonals and trends by exponentially weighted moving averages. *International Journal of Forecasting*, 20(1), 5–10.

- <https://doi.org/10.1016/j.ijforecast.2003.09.015>
- Hyndman, R J, & Athanasopoulos, G. (2021). *Forecasting: Principles and Practice*. OTexts. <https://books.google.pt/books?id=gZB-zgEACAAJ>
- Hyndman, Rob J., & Billah, B. (2003). Unmasking the Theta method. *International Journal of Forecasting*, 19(2), 287–290. [https://doi.org/10.1016/S0169-2070\(01\)00143-1](https://doi.org/10.1016/S0169-2070(01)00143-1)
- Hyndman, Rob J, & Koehler, A. B. (2005). and Business Statistics Another Look at Measures of Forecast Accuracy Another look at measures of forecast accuracy. *International Journal of Forecasting*, 22(November), 679–688. <http://www.sciencedirect.com/science/article/pii/S0169207006000239%5Cnhttp://core.ac.uk/download/pdf/6340761.pdf>
- Jahanshiri, E., & Shariff, A. R. M. (2014). Developing web-based data analysis tools for precision farming using R and Shiny. *IOP Conference Series: Earth and Environmental Science*, 20(1). <https://doi.org/10.1088/1755-1315/20/1/012014>
- Koutsandreas, D., Spiliotis, E., Petropoulos, F., & Assimakopoulos, V. (2021). On the selection of forecasting accuracy measures. *Journal of the Operational Research Society*, 0(0), 1–18. <https://doi.org/10.1080/01605682.2021.1892464>
- Krispin, R. (MIT). (2020). *Package TSstudio*. Cran. <https://cran.r-project.org/web/packages/TSstudio/index.html>
- Laranjeira, M., Trigo, A., & Varajão, J. (2019). Success of Software Development Projects in Portugal-preliminary results. *CAPSI 2019 Proceedings*.
- Makridakis, S., & Hibon, M. (1997). ARMA models and the Box-Jenkins methodology. *Journal of Forecasting*, 16(3), 147–163. [https://doi.org/10.1002/\(SICI\)1099-131X\(199705\)16:3<147::AID-FOR652>3.0.CO;2-X](https://doi.org/10.1002/(SICI)1099-131X(199705)16:3<147::AID-FOR652>3.0.CO;2-X)
- Mehdiyev, N., Enke, D., Fettke, P., & Loos, P. (2016). Evaluating Forecasting Methods by Considering Different Accuracy Measures. *Procedia Computer Science*, 95, 264–271. <https://doi.org/10.1016/j.procs.2016.09.332>
- Montgomery, D. C., Jennings, C. L., & Kulahci, M. (2015). *Introduction to Time Series Analysis and Forecasting*. Wiley. <https://books.google.pt/books?id=Xeh8CAAAQBAJ>
- Moritz, S., & Bartz-Beielstein, T. (2017). imputeTS: Time series missing value imputation in R. *R Journal*, 9(1), 207–218. <https://doi.org/10.32614/rj-2017-009>
- Moritz, S., Sardá, A., Bartz-Beielstein, T., Zaefferer, M., & Stork, J. (2015). *Comparison of different Methods for Univariate Time Series Imputation in R*. <http://arxiv.org/abs/1510.03924>
- Ostertagova, E., & Ostertag, O. (2011). The Simple Exponential Smoothing Model. *Modelling of Mechanical and Mechatronic Systems*, 1(September 2011), 380–384. [http://www.researchgate.net/publication/256088917\\_THE\\_SIMPLE\\_EXPONENTIAL\\_SMOOTHING\\_MODEL](http://www.researchgate.net/publication/256088917_THE_SIMPLE_EXPONENTIAL_SMOOTHING_MODEL)

- Proietti, T. (1991). Forecasting with Structural Time Series Models. In *Blackwell Publishers, Oxford*.
- R Core Team. (2021). *R: A Language and Environment for Statistical Computing*. <https://www.r-project.org/>
- Reunanen, N., Rätty, T., Jokinen, J. J., Hoyt, T., & Culler, D. (2020). Unsupervised online detection and prediction of outliers in streams of sensor data. *International Journal of Data Science and Analytics*, 9(3), 285–314. <https://doi.org/10.1007/s41060-019-00191-3>
- Satyahadewi, N., & Perdana, H. (2021). Web Application Development for Inferential Statistics using R Shiny. *Proceedings of the 1st International Conference on Mathematics and Mathematics Education (ICMMEd 2020)*, 550(Icmmed 2020), 425–429. <https://doi.org/10.2991/assehr.k.210508.099>
- Song, Q. C., Tang, C., & Wee, S. (2021). Making Sense of Model Generalizability: A Tutorial on Cross-Validation in R and Shiny. *Advances in Methods and Practices in Psychological Science*, 4(1). <https://doi.org/10.1177/2515245920947067>
- Taylor, S. J., & Letham, B. (2018). Forecasting at Scale. *American Statistician*, 72(1), 37–45. <https://doi.org/10.1080/00031305.2017.1380080>
- Theußl, S., Ligges, U., & Hornik, K. (2011). Prospects and challenges in R package development. *Computational Statistics*, 26(3), 395–404. <https://doi.org/10.1007/s00180-010-0205-5>
- Trigo, A., & Varajão, J. (2020). IT Project Management Critical Success Factors. *Computational Science and Its Applications – ICCSA 2020 20th International Conference, Cagliari, Italy, July 1–4, 2020, Proceedings, Part VI*, 1–11. [https://doi.org/https://doi.org/10.1007/978-3-030-58817-5\\_51](https://doi.org/https://doi.org/10.1007/978-3-030-58817-5_51)
- Vallis, O., Hochenbaum, J., & Kejariwal, A. (2014). A novel technique for long-term anomaly detection in the cloud. *6th USENIX Workshop on Hot Topics in Cloud Computing, HotCloud 2014*.
- van Buuren, S., & Groothuis-Oudshoorn, K. (2011). mice: Multivariate imputation by chained equations in R. *Journal of Statistical Software*, 45(3), 1–67. <https://doi.org/10.18637/jss.v045.i03>
- Volodymyr, S. (Chief D. S. at D. (2021). *Anomaly Detection: Definition, Best Practices and Use Cases*. Datrics Inc. <https://datrics.ai/anomaly-detection-best-practices>
- Weiss, Christoph E.; Roetzer, Gernot R.; Raviv, E. (2018). *ForecastComb*.
- Weiss, C. E., Raviv, E., & Roetzer, G. (2019). Forecast combinations in R using the ForecastComb package. *R Journal*, 10(2), 262–281. <https://doi.org/10.32614/RJ-2018-052>
- Wickham, H. (2020). *Mastering Shiny* (O. Media (ed.)). O’Reilly Media. <https://mastering-shiny.org/>







## 7 APÊNDICES

### 7.1 Código R - Missing Data (secção 5.2.1.1)

```
set.seed(1)

dt1<-datasets::AirPassengers

registosTotais<-length(dt1)

registosAmostra<-trunc(registosTotais*20/100)

s=sampling::srswor(registosAmostra,registosTotais)

dt1[s==1]<-NA

readr::write_csv(tsibble::as_tsibble(dt1), 'AirPassengers-20.csv',na="NA")
```

*Código R 7-1 - Gerar séries com falhas aleatórias de dados*

## 7.2 Imputação de dados – 20% (secção 5.2.1.2)

Tabela 7.1 - Imputação20% - Valores Imputados

index	Real	Valores imputados									
		M01	M02	M03	M04	M05	M06	M07	M08	M09	M10
01/07/1949	148	141,5	147,5	143,4	137,6	141,5	141,5	141,5	280,8	141,5	172,0
01/02/1950	126	128,0	126,6	128,0	115,2	128,0	128,0	128,0	280,8	128,0	114,0
01/09/1950	158	151,5	156,9	151,8	172,5	151,5	151,5	151,5	280,8	151,5	181,0
01/09/1951	184	190,8	187,6	193,8	201,5	190,8	190,8	190,8	280,8	190,8	193,0
01/10/1951	162	182,5	174,7	182,5	204,0	182,5	182,5	182,5	280,8	182,5	181,0
01/11/1951	146	174,3	165,8	170,9	206,5	174,3	174,3	174,3	280,8	174,3	178,0
01/01/1952	171	175,0	177,8	172,9	167,4	175,0	175,0	175,0	280,8	175,0	243,0
01/02/1952	180	184,0	191,1	186,4	168,8	184,0	184,0	184,0	280,8	184,0	199,0
01/06/1952	218	189,5	193,6	187,7	184,6	189,5	189,5	189,5	280,8	189,5	178,0
01/07/1952	230	196,0	205,2	196,0	186,2	196,0	196,0	196,0	280,8	196,0	234,0
01/08/1952	242	202,5	212,1	205,2	187,8	202,5	202,5	202,5	280,8	202,5	264,0
01/03/1953	236	215,5	217,9	215,5	197,6	215,5	215,5	215,5	280,8	215,5	180,0
01/08/1954	293	280,5	297,0	282,7	304,7	280,5	280,5	280,5	280,8	280,5	237,0
01/10/1954	229	231,0	217,4	231,0	261,0	231,0	231,0	231,0	280,8	231,0	235,0
01/01/1955	242	241,7	251,0	246,2	230,6	241,7	241,7	241,7	280,8	241,7	302,0
01/02/1955	233	254,3	261,5	259,8	232,1	254,3	254,3	254,3	280,8	254,3	274,0
01/07/1955	364	331,0	348,7	335,4	317,5	331,0	331,0	331,0	280,8	331,0	355,0
01/12/1955	278	250,3	224,4	243,2	238,5	250,3	250,3	250,3	280,8	250,3	267,0
01/01/1956	284	263,7	239,3	258,0	239,9	263,7	263,7	263,7	280,8	263,7	312,0
01/05/1956	318	343,5	331,1	339,4	315,2	343,5	343,5	343,5	280,8	343,5	302,0
01/09/1957	404	413,0	410,3	426,6	470,3	413,0	413,0	413,0	280,8	413,0	359,0
01/10/1957	347	359,0	336,5	343,9	473,6	359,0	359,0	359,0	280,8	359,0	405,0
01/02/1958	318	351,0	353,0	351,0	342,0	351,0	351,0	351,0	280,8	351,0	467,0
01/06/1959	472	484,0	486,3	484,0	422,4	484,0	484,0	484,0	280,8	484,0	363,0
01/09/1959	463	483,0	496,5	483,0	562,4	483,0	483,0	483,0	280,8	483,0	559,0
01/01/1960	417	398,0	404,7	398,0	407,2	398,0	398,0	398,0	280,8	398,0	390,0
01/04/1960	461	457,7	443,3	450,4	421,2	457,7	457,7	457,7	280,8	457,7	405,0
01/05/1960	472	496,3	472,3	487,3	423,4	496,3	496,3	496,3	280,8	496,3	420,0

Tabela 7.2 - Imputação20% - Erros de imputação

index	Erro de imputação = (Valor Real – Valor Imputado)									
	M01	M02	M03	M04	M05	M06	M07	M08	M09	M10
01/07/1949	-6,5	-0,5	-4,6	-10,4	-6,5	-6,5	-6,5	132,8	-6,5	24,0
01/02/1950	2,0	0,6	2,0	-10,8	2,0	2,0	2,0	154,8	2,0	-12,0
01/09/1950	-6,5	-1,1	-6,2	14,5	-6,5	-6,5	-6,5	122,8	-6,5	23,0
01/09/1951	6,8	3,6	9,8	17,5	6,8	6,8	6,8	96,8	6,8	9,0
01/10/1951	20,5	12,7	20,5	42,0	20,5	20,5	20,5	118,8	20,5	19,0
01/11/1951	28,3	19,8	24,9	60,5	28,3	28,3	28,3	134,8	28,3	32,0
01/01/1952	4,0	6,8	1,9	-3,6	4,0	4,0	4,0	109,8	4,0	72,0
01/02/1952	4,0	11,1	6,4	-11,2	4,0	4,0	4,0	100,8	4,0	19,0
01/06/1952	-28,5	-24,4	-30,3	-33,4	-28,5	-28,5	-28,5	62,8	-28,5	-40,0
01/07/1952	-34,0	-24,8	-34,0	-43,8	-34,0	-34,0	-34,0	50,8	-34,0	4,0
01/08/1952	-39,5	-29,9	-36,8	-54,2	-39,5	-39,5	-39,5	38,8	-39,5	22,0
01/03/1953	-20,5	-18,1	-20,5	-38,4	-20,5	-20,5	-20,5	44,8	-20,5	-56,0
01/08/1954	-12,5	4,0	-10,3	11,7	-12,5	-12,5	-12,5	-12,3	-12,5	-56,0
01/10/1954	2,0	-11,6	2,0	32,0	2,0	2,0	2,0	51,8	2,0	6,0
01/01/1955	-0,3	9,0	4,2	-11,4	-0,3	-0,3	-0,3	38,8	-0,3	60,0
01/02/1955	21,3	28,5	26,8	-0,9	21,3	21,3	21,3	47,8	21,3	41,0
01/07/1955	-33,0	-15,3	-28,6	-46,5	-33,0	-33,0	-33,0	-83,3	-33,0	-9,0
01/12/1955	-27,7	-53,6	-34,8	-39,5	-27,7	-27,7	-27,7	2,8	-27,7	-11,0
01/01/1956	-20,3	-44,7	-26,0	-44,1	-20,3	-20,3	-20,3	-3,3	-20,3	28,0
01/05/1956	25,5	13,1	21,4	-2,8	25,5	25,5	25,5	-37,3	25,5	-16,0
01/09/1957	9,0	6,3	22,6	66,3	9,0	9,0	9,0	-123,3	9,0	-45,0
01/10/1957	12,0	-10,5	-3,1	126,6	12,0	12,0	12,0	-66,3	12,0	58,0
01/02/1958	33,0	35,0	33,0	24,0	33,0	33,0	33,0	-37,3	33,0	149,0
01/06/1959	12,0	14,3	12,0	-49,6	12,0	12,0	12,0	-191,3	12,0	-109,0
01/09/1959	20,0	33,5	20,0	99,4	20,0	20,0	20,0	-182,3	20,0	96,0
01/01/1960	-19,0	-12,3	-19,0	-9,8	-19,0	-19,0	-19,0	-136,3	-19,0	-27,0
01/04/1960	-3,3	-17,7	-10,6	-39,8	-3,3	-3,3	-3,3	-180,3	-3,3	-56,0
01/05/1960	24,3	0,3	15,3	-48,6	24,3	24,3	24,3	-191,3	24,3	-52,0

Tabela 7.3 - Imputação20% - Erro Absoluto

index	Erro Absoluto									
	M01	M02	M03	M04	M05	M06	M07	M08	M09	M10
01/07/1949	6,5	0,5	4,6	10,4	6,5	6,5	6,5	132,8	6,5	24,0
01/02/1950	2,0	0,6	2,0	10,8	2,0	2,0	2,0	154,8	2,0	12,0
01/09/1950	6,5	1,1	6,2	14,5	6,5	6,5	6,5	122,8	6,5	23,0
01/09/1951	6,8	3,6	9,8	17,5	6,8	6,8	6,8	96,8	6,8	9,0
01/10/1951	20,5	12,7	20,5	42,0	20,5	20,5	20,5	118,8	20,5	19,0
01/11/1951	28,3	19,8	24,9	60,5	28,3	28,3	28,3	134,8	28,3	32,0
01/01/1952	4,0	6,8	1,9	3,6	4,0	4,0	4,0	109,8	4,0	72,0
01/02/1952	4,0	11,1	6,4	11,2	4,0	4,0	4,0	100,8	4,0	19,0
01/06/1952	28,5	24,4	30,3	33,4	28,5	28,5	28,5	62,8	28,5	40,0
01/07/1952	34,0	24,8	34,0	43,8	34,0	34,0	34,0	50,8	34,0	4,0
01/08/1952	39,5	29,9	36,8	54,2	39,5	39,5	39,5	38,8	39,5	22,0
01/03/1953	20,5	18,1	20,5	38,4	20,5	20,5	20,5	44,8	20,5	56,0
01/08/1954	12,5	4,0	10,3	11,7	12,5	12,5	12,5	12,3	12,5	56,0
01/10/1954	2,0	11,6	2,0	32,0	2,0	2,0	2,0	51,8	2,0	6,0
01/01/1955	0,3	9,0	4,2	11,4	0,3	0,3	0,3	38,8	0,3	60,0
01/02/1955	21,3	28,5	26,8	0,9	21,3	21,3	21,3	47,8	21,3	41,0
01/07/1955	33,0	15,3	28,6	46,5	33,0	33,0	33,0	83,3	33,0	9,0
01/12/1955	27,7	53,6	34,8	39,5	27,7	27,7	27,7	2,8	27,7	11,0
01/01/1956	20,3	44,7	26,0	44,1	20,3	20,3	20,3	3,3	20,3	28,0
01/05/1956	25,5	13,1	21,4	2,8	25,5	25,5	25,5	37,3	25,5	16,0
01/09/1957	9,0	6,3	22,6	66,3	9,0	9,0	9,0	123,3	9,0	45,0
01/10/1957	12,0	10,5	3,1	126,6	12,0	12,0	12,0	66,3	12,0	58,0
01/02/1958	33,0	35,0	33,0	24,0	33,0	33,0	33,0	37,3	33,0	149,0
01/06/1959	12,0	14,3	12,0	49,6	12,0	12,0	12,0	191,3	12,0	109,0
01/09/1959	20,0	33,5	20,0	99,4	20,0	20,0	20,0	182,3	20,0	96,0
01/01/1960	19,0	12,3	19,0	9,8	19,0	19,0	19,0	136,3	19,0	27,0
01/04/1960	3,3	17,7	10,6	39,8	3,3	3,3	3,3	180,3	3,3	56,0
01/05/1960	24,3	0,3	15,3	48,6	24,3	24,3	24,3	191,3	24,3	52,0

Tabela 7.4 - Imputação 20% - Percentagem Erro Absoluto

index	Percentagem Erro Absoluto									
	M01	M02	M03	M04	M05	M06	M07	M08	M09	M10
01/07/1949	4,4	0,3	3,1	7,1	4,4	4,4	4,4	89,7	4,4	16,2
01/02/1950	1,6	0,5	1,6	8,6	1,6	1,6	1,6	122,8	1,6	9,5
01/09/1950	4,1	0,7	3,9	9,2	4,1	4,1	4,1	77,7	4,1	14,6
01/09/1951	3,7	1,9	5,3	9,5	3,7	3,7	3,7	52,6	3,7	4,9
01/10/1951	12,7	7,9	12,7	25,9	12,7	12,7	12,7	73,3	12,7	11,7
01/11/1951	19,3	13,5	17,1	41,4	19,3	19,3	19,3	92,3	19,3	21,9
01/01/1952	2,3	4,0	1,1	2,1	2,3	2,3	2,3	64,2	2,3	42,1
01/02/1952	2,2	6,2	3,6	6,2	2,2	2,2	2,2	56,0	2,2	10,6
01/06/1952	13,1	11,2	13,9	15,3	13,1	13,1	13,1	28,8	13,1	18,3
01/07/1952	14,8	10,8	14,8	19,0	14,8	14,8	14,8	22,1	14,8	1,7
01/08/1952	16,3	12,4	15,2	22,4	16,3	16,3	16,3	16,0	16,3	9,1
01/03/1953	8,7	7,7	8,7	16,3	8,7	8,7	8,7	19,0	8,7	23,7
01/08/1954	4,3	1,4	3,5	4,0	4,3	4,3	4,3	4,2	4,3	19,1
01/10/1954	0,9	5,1	0,9	14,0	0,9	0,9	0,9	22,6	0,9	2,6
01/01/1955	0,1	3,7	1,7	4,7	0,1	0,1	0,1	16,0	0,1	24,8
01/02/1955	9,2	12,2	11,5	0,4	9,2	9,2	9,2	20,5	9,2	17,6
01/07/1955	9,1	4,2	7,9	12,8	9,1	9,1	9,1	22,9	9,1	2,5
01/12/1955	10,0	19,3	12,5	14,2	10,0	10,0	10,0	1,0	10,0	4,0
01/01/1956	7,2	15,7	9,1	15,5	7,2	7,2	7,2	1,1	7,2	9,9
01/05/1956	8,0	4,1	6,7	0,9	8,0	8,0	8,0	11,7	8,0	5,0
01/09/1957	2,2	1,6	5,6	16,4	2,2	2,2	2,2	30,5	2,2	11,1
01/10/1957	3,5	3,0	0,9	36,5	3,5	3,5	3,5	19,1	3,5	16,7
01/02/1958	10,4	11,0	10,4	7,6	10,4	10,4	10,4	11,7	10,4	46,9
01/06/1959	2,5	3,0	2,5	10,5	2,5	2,5	2,5	40,5	2,5	23,1
01/09/1959	4,3	7,2	4,3	21,5	4,3	4,3	4,3	39,4	4,3	20,7
01/01/1960	4,6	3,0	4,6	2,4	4,6	4,6	4,6	32,7	4,6	6,5
01/04/1960	0,7	3,8	2,3	8,6	0,7	0,7	0,7	39,1	0,7	12,1
01/05/1960	5,2	0,1	3,2	10,3	5,2	5,2	5,2	40,5	5,2	11,0