



Maria Goreti Simão da Silva - N° 41142

Mineração de Repositórios de *Software*
Modelos de Previsão de Pedidos de Evolução de *Software*

Dissertação para obtenção do Grau de Mestre em
Engenharia Informática

Orientador: Prof. Doutor **Miguel Carlos Pacheco Afonso Goulão**,
Professor Auxiliar da Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa.

Júri:

Presidente: Doutor **Henrique João Lopes Domingos**, Professor Auxiliar da
Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa.

Vogais: Doutor **João Carlos Pascoal Faria**, Professor Auxiliar da Faculdade
de Engenharia da Universidade do Porto.

Doutor **Miguel Carlos Pacheco Afonso Goulão**,
Professor Auxiliar da Faculdade de Ciências e Tecnologia da
Universidade Nova de Lisboa.



Maria Goreti Simão da Silva - N° 41142

Mineração de Repositórios de *Software*
Modelos de Previsão de Pedidos de Evolução de *Software*

Dissertação para obtenção do Grau de Mestre em
Engenharia Informática

Orientador: Prof. Doutor **Miguel Carlos Pacheco Afonso Goulão**,
Professor Auxiliar da Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa.

Júri:

Presidente: Doutor **Henrique João Lopes Domingos**, Professor Auxiliar da
Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa.

Vogais: Doutor **João Carlos Pascoal Faria**, Professor Auxiliar da Faculdade de
Engenharia da Universidade do Porto.

Doutor **Miguel Carlos Pacheco Afonso Goulão**,
Professor Auxiliar da Faculdade de Ciências e Tecnologia da
Universidade Nova de Lisboa.

Mineração de Repositórios de *Software*

Modelos de previsão de pedidos de evolução de *software*

Copyright © **Maria Goreti Simão da Silva**, Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa.

A Faculdade de Ciências e Tecnologia e a Universidade Nova de Lisboa têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer outro meio conhecido ou que venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objetivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.

Dedicatória à minha família.

Agradecimentos

Para a consecução do trabalho apresentado foram imprescindíveis o contributo, a disponibilidade e o incentivo por parte de algumas pessoas, a quem reconhecidamente pretendo expressar o meu apreço e gratidão.

Ao meu Orientador, Professor Doutor Miguel Carlos Pacheco Afonso Goulão, pelas rápidas e eficazes orientações prestadas, pelo incentivo e apoio incondicionais, sem os quais não teria sido possível esta produção.

Aos meus amigos, em especial ao Fernando Gama, pela troca de ideias e apoio prestado.

À minha família pelo incentivo e sensibilidade demonstrados durante a execução deste trabalho.

A descoberta, ou confirmação, de tendências e padrões na evolução de sistemas de *software* tem vindo a conferir relevância à mineração de repositórios de *software*. A Engenharia de *Software* recorre assim a abordagens específicas para a mineração de dados, originários da construção de *software*, tais como: código-fonte, histórico de versões (*logs*), relatórios de erros (rastreamento de defeitos), entre outros. Os membros da equipa de desenvolvimento são um recurso valioso no processo de desenvolvimento e manutenção de um *software*. Para otimizar o seu trabalho têm surgido ferramentas de *software* integradas e ligadas às atividades de desenvolvimento que permitem que os dados (tais como, pedidos de evolução, repositórios de controlo de versões, entre outros) sejam armazenados automaticamente. Estes dados poderão então ser recuperados e devidamente tratados para que se obtenha informação importante para uma melhoria do processo de desenvolvimento de *software*. A mineração de dados nos repositórios do processo de desenvolvimento permite detetar tendências e padrões quer no processo de desenvolvimento quer nos artefactos desenvolvidos, constituindo assim uma importante ferramenta de apoio à gestão desse processo.

Pretendemos com este estudo usar a informação contida em repositórios de pedidos de evolução para criar modelos de previsão da distribuição desses pedidos ao longo do tempo. Esse tipo de modelos é útil para facilitar a gestão do processo de desenvolvimento e manutenção de *software*, na medida em que permitem prever períodos em que a densidade de pedidos será maior, contrastando com outros em que há menos pedidos, sendo essa informação relevante para a alocação de recursos humanos ao processo de desenvolvimento e manutenção. A abordagem a utilizar visa estudar que tipos de modelos são mais adequados, conforme o volume de dados histórico existente e o padrão de versões a que o repositório diz respeito. Em particular, queremos saber:

- Será que a escolha do “melhor” modelo é relativamente estável, ou muito volátil? A implicação é que podemos ter de atualizar modelos muito frequentemente, ou nem por isso.
- Será que os modelos que integram informação sazonal se tornam dominantes? Se sim, quanto tempo de dados históricos é necessário para que a informação sazonal seja relevante?

Palavras-chave: Repositórios de *Software*, Sistemas de Rastreamento de Defeitos, Evolução de *Software*, Modelos de Previsão, Séries Temporais.

The discovery, or confirmation, of trends and patterns in the evolution of software systems has been giving importance to mining software repositories. Software Engineering uses specific approaches to mining data from software construction, such as: source code, version history (logs), bug reports (defect tracking), among others. Members of the development team are essential during the preparation and maintenance of software. New integrated software tools related to development activities have been adopted to optimize their work and these have allowed data (such as evolution requests, version control repositories, etc.) to be stored automatically. These data can be recovered and analyzed in order to obtain important information for facilitating software process improvement. Mining the repositories of the development process allows detecting trends and patterns both in the development process and in the developed artifacts, making it an important tool to support the management of the software development process.

In this study we use information contained in repositories systems of evolution requests so as to create predictive models of the distribution of those requests throughout the time. These types of models are useful to facilitate the management of the development and maintenance of software, since they will be able to predict periods with a great deal of orders, in contrast to other periods in which there will be fewer, and that information is relevant to the allocation of human resources to the process of development and maintenance. The approach to be used aims to study the most appropriate types of models, according to existing historical data and the pattern of versions which the repository deals with. In particular, we want to know:

- Is the choice of the "best" model relatively stable, or too volatile? The implication is that we may have to update models very often or not at all.*
- Do the models incorporating seasonal information become dominant? If so, how much historical data is required for the seasonal information to be relevant?*

Keywords: *Software Repositories, Defect Tracking Systems, Software Evolution, Prediction Models, Time Series.*

Índice Geral

ÍNDICE DE FIGURAS	VIII
ÍNDICE DE QUADROS.....	IX
ÍNDICE DE TABELAS.....	IX
ÍNDICE DE APÊNDICES	XII
SIGLAS E ABREVIATURAS	XII
1. INTRODUÇÃO.....	1
1.1. MOTIVAÇÃO	2
1.2. DESCRIÇÃO DO PROBLEMA.....	3
1.3. SOLUÇÃO APRESENTADA	4
1.4. PRINCIPAIS CONTRIBUIÇÕES PREVISTAS.....	5
1.5. ESTRUTURA DO RELATÓRIO.....	6
PARTE I – ENQUADRAMENTO TEÓRICO	7
2. CONTEXTO.....	8
2.1. EVOLUÇÃO DE <i>SOFTWARE</i>	VII
2.2. REPOSITÓRIOS DE SISTEMAS DE <i>SOFTWARE</i>	10
2.2.1.1. <i>Bugzilla</i>	11
2.3. ANÁLISE DE SÉRIES TEMPORAIS.....	12
3. TRABALHO RELACIONADO.....	15
3.1. ESTUDOS SOBRE FERRAMENTAS DE EXTRAÇÃO DE DADOS EM S. DE REPOSITÓRIOS SOFT. ...	15
3.2. ESTUDOS SOBRE FERRAMENTAS DE VISUALIZAÇÃO E ANÁLISE DE INFORMAÇÃO.....	15
3.3. ESTUDOS SOBRE PREVISÃO DE PEDIDOS DE EVOLUÇÃO NOS SISTEMAS DE <i>SOFTWARE</i>	15
PARTE II – ESTUDO EMPÍRICO	21
4. METODOLOGIA DA INVESTIGAÇÃO	22
4.1. MOTIVAÇÃO	22
4.1.1. <i>Contexto e descrição do problema</i>	22
4.1.2. <i>Objetivos do trabalho</i>	22
4.1.3. <i>Natureza da investigação</i>	23
4.1.4. <i>Questões de investigação</i>	24
4.2. PLANEAMENTO EXPERIMENTAL	27
4.2.1. <i>Objetivos experimentais</i>	27
4.2.2. <i>Unidades experimentais</i>	27
4.2.3. <i>Material experimental</i>	28
4.2.4. <i>Tarefas</i>	28
4.2.5. <i>Hipóteses e variáveis</i>	29
4.2.6. <i>Desenho</i>	30
4.2.7. <i>Procedimento</i>	30
4.2.8. <i>Procedimento de análise</i>	34
4.2.8.1. <i>Métricas de ajuste</i>	38
4.2.8.2. <i>Métricas de erro</i>	39
4.2.8.3. <i>Métricas de normalização</i>	42
4.3. EXECUÇÃO.....	43

5. RESULTADOS	44
5.1. APRESENTAÇÃO DE RESULTADOS	44
5.1.1. Eclipse JDT – Modelo de Previsão EXSS	46
5.1.2. Eclipse JDT – Modelo de Previsão EXWA.....	46
5.1.3. Eclipse JDT – Modelo de Previsão ANS100	47
5.1.4. Eclipse JDT – Modelo de Previsão ANS000	48
5.1.5. Eclipse JDT – Métricas estatísticas dos modelos gerados	49
5.1.6. Eclipse JDT – Comparação do número de modelos gerados segundo os critérios RSQUARE, MAPE e Normalized BIC	50
5.1.7. Eclipse JDT – Comparação das métricas de ajuste, de erro e de normalização segundo os critérios RSQUARE, MAPE e Normalized BIC	51
5.1.8. Eclipse JDT – Avaliação global dos modelos selecionados.....	59
5.1.9. Eclipse JDT – Modelo emergente.....	60
5.1.10. Outros Projetos.....	61
5.1.10.1. Eclipse Platform.....	61
5.1.10.2. Eclipse.....	62
5.1.10.3. Eclipse Android.....	62
5.1.10.4. Gnome Evolution.....	63
5.1.10.5. Gnome GStreamer	64
5.1.10.6. Mozilla Core	65
5.1.10.7. Mozilla Firefox	66
6. INTERPRETAÇÃO DE RESULTADOS.....	67
6.1. AVALIAÇÃO DOS RESULTADOS E IMPLICAÇÕES	67
6.2. AMEAÇAS À VALIDADE	70
6.3. INFERÊNCIAS	71
6.4. APRENDIZAGENS.....	71
6.5. LIMITAÇÕES DO ESTUDO.....	71
7. CONSIDERAÇÕES FINAIS.....	73
7.1. CONCLUSÕES.....	73
7.2. TRABALHO FUTURO.....	74
BIBLIOGRAFIA	75

Índice de Figuras

FIGURA 1.1 - SOLUÇÃO PARA CAPACIDADE DE PREVISÃO DA DISTRIBUIÇÃO DOS PEDIDOS DE EVOLUÇÃO.....	4
FIGURA 2.1 - MODELO DE UM RELATÓRIO DE DEFEITOS DO <i>BUGZILLA</i>	10
FIGURA 4.1 – PADRÃO GMQ APLICADO AO PROBLEMA.....	25
FIGURA 4.2 – DIAGRAMA DE ATIVIDADES DA SOLUÇÃO PROPOSTA (UML)	28
FIGURA 4.3 – DIAGRAMA DE SEQUÊNCIA DE ATIVIDADES EM UML	32
FIGURA 5.1 – MODELO DE PREVISÃO EXWA.....	46
FIGURA 5.2 – MODELO DE PREVISÃO EXSS	46
FIGURA 5.3 – MODELO DE PREVISÃO ANS100.....	46
FIGURA 5.4 – MODELO DE PREVISÃO ANS000.....	46
FIGURA 5.5 - MODELOS GERADOS SEGUNDO O CRITÉRIO <i>RSQUARE</i> E <i>MAPE</i> (<i>ECLIPSE JDT</i>).....	50
FIGURA 5.6 – MODELOS GERADOS SEGUNDO O CRITÉRIO <i>NORMALIZED BIC</i> (<i>ECLIPSE JDT</i>).....	51
FIGURA 5.7 – MÉTRICA DE ERRO <i>ERRVARIANCE</i>	51
FIGURA 5.8 – MÉTRICA DE ERRO <i>MEANSQERR</i>	52
FIGURA 5.9 – MÉTRICA DE ERRO <i>ROOTMEANSQERR</i>	52
FIGURA 5.10 – MÉTRICA DE ERRO <i>MAPE</i>	53
FIGURA 5.11 – MÉTRICA DE ERRO <i>MEANABSERR</i>	54
FIGURA 5.12 – MÉTRICA DE ERRO <i>MAXABSPCTERR</i>	54
FIGURA 5.13 – MÉTRICA DE ERRO <i>MAXABSERR</i>	55
FIGURA 5.14 – MÉTRICA DE AJUSTE <i>RSQUARE</i>	56
FIGURA 5.15 – MÉTRICA DE AJUSTE <i>STATIONARYRSQUARE</i>	56
FIGURA 5.16 – MÉTRICA <i>LOGLIKELIHOOD</i>	57
FIGURA 5.17 – MÉTRICA <i>BIC</i>	57
FIGURA 5.18 – MÉTRICA <i>NORMALIZED BIC</i>	58
FIGURA 5.19 – MÉTRICA <i>AKAIKEIC</i>	58
FIGURA 5.20 – DIAGRAMA DE <i>KIVIAT</i> – MÉTRICAS DE ERRO E DE NORMALIZAÇÃO	59
FIGURA 5.21 – DIAGRAMA DE <i>KIVIAT</i> – MÉTRICAS DE AJUSTE E DE NORMALIZAÇÃO	60
FIGURA 5.22 – COMPARAÇÃO DA PRECISÃO DAS PREVISÕES	60
FIGURA 5.23 - MODELOS GERADOS SEGUNDO O CRITÉRIO <i>RSQUARE</i> E <i>NORMALIZED BIC</i> (<i>ECLIPSE PLATFORM</i>)	61
FIGURA 5.24 - MODELOS GERADOS SEGUNDO O CRITÉRIO <i>RSQUARE</i> , <i>MAPE</i> E <i>NORMALIZED BIC</i> (<i>ECLIPSE</i>) ...	62
FIGURA 5.25 - MODELOS GERADOS SEGUNDO O CRITÉRIO <i>RSQUARE</i> , <i>MAPE</i> E <i>NORMALIZED BIC</i> (<i>ANDROID</i>)..	63
FIGURA 5.26 - MODELOS GERADOS SEGUNDO O CRITÉRIO <i>RSQUARE</i> , <i>MAPE</i> E <i>NORMALIZED BIC</i> (<i>G.EVOLUTION</i>)	64
FIGURA 5.27 - MODELOS GERADOS SEGUNDO O CRITÉRIO <i>RSQUARE</i> , <i>MAPE</i> E <i>NORMALIZED BIC</i> (<i>G.GSTREAMER</i>)	65
FIGURA 5.28 - MODELOS GERADOS SEGUNDO O CRITÉRIO <i>RSQUARE</i> , <i>MAPE</i> E <i>NORMALIZED BIC</i> (<i>MOZILLA CORE</i>)	65
FIGURA 5.29 - MODELOS GERADOS SEGUNDO O CRITÉRIO <i>RSQUARE</i> , <i>MAPE</i> E <i>NORMALIZED BIC</i> (<i>FIREFOX</i>)...	66

Índice de Quadros

QUADRO 2.1 – LEIS DE <i>LEHMAN</i>	8
QUADRO 2.2 – TIPOS DE MODELOS DE SUAVIZAÇÃO EXPONENCIAL	12
QUADRO 3.1 – FERRAMENTAS E TÉCNICA DE ANÁLISE DE EVOLUÇÃO APRESENTADAS	17
QUADRO 4.1 – MODELO GQM – QUESTÕES E MÉTRICAS DE INVESTIGAÇÃO	26
QUADRO 4.2 – HIPÓTESES TESTADAS	29
QUADRO 4.3 – EXEMPLO DE PLANEAMENTO DE TESTES	36
QUADRO 4.4 – DESCRIÇÃO DE INCÓGNITAS USADAS NAS FÓRMULAS DAS MÉTRICAS	38
QUADRO 4.5 – MÉTRICAS DE AJUSTE	39
QUADRO 4.6 – MÉTRICAS DE ERRO	41
QUADRO 4.7 – MÉTRICAS DE NORMALIZAÇÃO	43

Índice de Tabelas

TABELA 4.1 – TABELA DE DADOS PARCIAIS DO SOFTWARE ECLIPSE JDT	29
TABELA 4.2 – NÚMERO TOTAL DE OBSERVAÇÕES	30
TABELA 4.3 – N° DE PEDIDOS DE EVOLUÇÃO DOS SISTEMAS	31
TABELA 4.5 – CORRESPONDÊNCIA ENTRE OS DIAGRAMAS DE ATIVIDADE E DE SEQUÊNCIA	34
TABELA 5.1 – MÉTRICAS ESTATÍSTICAS DOS MODELOS GERADOS	49
TABELA 6.1 – MODELOS QUE MAXIMIZAM A CAPACIDADE DE EXPLICAÇÃO PEDIDOS EVOLUÇÃO ..	68
TABELA 6.2 – MODELOS QUE APRESENTAM MENOR TAXA DE ERRO PEDIDOS EVOLUÇÃO	68

Índice de Apêndices

APÊNDICE I – EXEMPLOS DE FICHEIROS MANIPULADOS NO TRABALHO EXPERIMENTAL	80
APÊNDICE II – EXEMPLOS DE FICHEIROS MANIPULADOS NO TRABALHO EXPERIMENTAL.....	81
APÊNDICE III - FIGURAS: <i>SOFTWARE ECLIPSEPLATFORM</i> – MODELOS DE PREVISÃO	82
APÊNDICE IV - FIGURAS: <i>SOFTWARE ECLIPSE</i> – MODELOS DE PREVISÃO.....	82
APÊNDICE V - FIGURAS: <i>SOFTWARE ANDROID</i> – MODELOS DE PREVISÃO	84
APÊNDICE VI - FIGURAS: <i>SOFTWARE GNOME EVOLUTION</i> – MODELOS DE PREVISÃO	86
APÊNDICE VII - FIGURAS: <i>SOFTWARE GNOME GSTREAMER</i> – MODELOS DE PREVISÃO.....	87
APÊNDICE VIII - FIGURAS: <i>SOFTWARE MOZILLA CORE</i> – MODELOS DE PREVISÃO.....	91
APÊNDICE IX - FIGURAS: <i>SOFTWARE MOZILLA FIREFOX</i> – MODELOS DE PREVISÃO.....	91
APÊNDICE X - FIGURAS: <i>SOFTWARE ECLIPSE PLATFORM</i> - MÉTRICAS DE ERRO	95
APÊNDICE XI - FIGURAS: <i>SOFTWARE ECLIPSE PLATFORM</i> - MÉTRICAS DE AJUSTE	97
APÊNDICE XII - FIGURAS: <i>SOFTWARE ECLIPSE PLATFORM</i> - MÉTRICAS DE C. DE INFORMAÇÃO	98
APÊNDICE XIII - FIGURAS: <i>SOFTWARE ECLIPSE</i> - MÉTRICAS DE ERRO	101
APÊNDICE XIV - FIGURAS: <i>SOFTWARE ECLIPSE</i> - MÉTRICAS DE AJUSTE	101
APÊNDICE XV - FIGURAS: <i>SOFTWARE ECLIPSE</i> - MÉTRICAS DE C. DE INFORMAÇÃO.....	102
APÊNDICE XVI – FIGURAS: <i>SOFTWARE ANDROID</i> - MÉTRICAS DE ERRO.....	103
APÊNDICE XVII - FIGURAS: <i>SOFTWARE ANDROID</i> - MÉTRICAS DE AJUSTE	105
APÊNDICE XVIII - FIGURAS: <i>SOFTWARE ANDROID</i> - MÉTRICAS DE C. DE INFORMAÇÃO	106
APÊNDICE XIX – FIGURAS: <i>SOFTWARE GNOME EVOLUTION</i> - MÉTRICAS DE ERRO	107
APÊNDICE XX - FIGURAS: <i>SOFTWARE GNOME EVOLUTION</i> - MÉTRICAS DE AJUSTE	109
APÊNDICE XXI - FIGURAS: <i>SOFTWARE GNOME EVOLUTION</i> - MÉTRICAS DE C. INFORMAÇÃO	110
APÊNDICE XXII – FIGURAS: <i>SOFTWARE GNOME GSTREAMER</i> - MÉTRICAS DE ERRO	111
APÊNDICE XXIII - FIGURAS: <i>SOFTWARE GNOME GSTREAMER</i> - MÉTRICAS DE AJUSTE	111
APÊNDICE XXIV - FIGURAS: <i>SOFTWARE GNOME GSTREAMER</i> – MÉTRICAS DE C. INFORMAÇÃO	113
APÊNDICE XXV – FIGURAS: <i>SOFTWARE MOZILLA CORE</i> - MÉTRICAS DE ERRO.....	114
APÊNDICE XXVI - FIGURAS: <i>SOFTWARE MOZILLA CORE</i> - MÉTRICAS DE AJUSTE.....	116
APÊNDICE XXVII - FIGURAS: <i>SOFTWARE MOZILLA CORE</i> - MÉTRICAS DE C. DE INFORMAÇÃO.....	117
APÊNDICE XXVIII – FIGURAS: <i>SOFTWARE MOZILLA FIREFOX</i> - MÉTRICAS DE ERRO	118
APÊNDICE XXVIII – FIGURAS: <i>SOFTWARE MOZILLA FIREFOX</i> - MÉTRICAS DE AJUSTE	118
APÊNDICE XXX - FIGURAS: <i>SOFTWARE MOZILLA FIREFOX</i> - MÉTRICAS DE C. DE INFORMAÇÃO	121

Siglas e Abreviaturas

ADT - Android Development Tools
ARIMA - *Autoregressive Integrated Moving Average*
BIC - *Bayesian Information Criterion*
CVS - *Concurrent Version System*
DFA - *Detrended Fluctuation Analysis*
EXS - *Exponential Smoothing - Simple*
EXSS - *Exponential Smoothing - Simple Seasonal*
EXWA - *Exponential Smoothing – Winters’ Additive*
EXWM - *Exponential Smoothing – Winters’ Multiplicative*
JDT - *Java Development Tooling*
LCL – *Lower Confidence Limit*
MAPE – *Mean Absolute Percentual Error*
SAV – *Saved (Statistical Data file)*
SDK - *Software Development Kit*
SPS - *Statistical Portable Sintax file*
SPPS - *Statistical Package for the Social Sciences*
SPV - *Statistical Portable Viewer (Output file)*
UCL – *Upper Confidence Limit*
UML - *Unified Modeling Language*
XML - *Extensible Markup Language*



Introdução

O crescimento do desenvolvimento de *software open source* levou a uma crescente disponibilidade de repositórios de livre acesso. Um pouco à semelhança do que é feito na arqueologia, podemos aprender muito sobre o processo de desenvolvimento de *software* estudando a evolução histórica dos sistemas, a partir da informação guardada nos seus repositórios. Consequentemente, tornou-se viável a obtenção de dados que abrangem o histórico de alteração de código-fonte obtido a partir de super-repositórios¹, sistemas de rastreio de defeitos, entre outros [1-3]. É importante salientar que, com a gradual disponibilidade de repositórios passou a ser possível aos investigadores estudarem soluções para determinados problemas², tais como: a deteção de sintomas de falta de qualidade nos artefactos, a previsão de que artefactos têm maior probabilidade de vir a necessitar de ações de evolução, entre outros. Existem muitas formas de explorar os repositórios de *software*.

Neste capítulo são apresentadas algumas considerações gerais sobre a motivação desta investigação, o problema que a justifica, a solução proposta e as principais contribuições previstas.

1.1. Motivação

Nos dias de hoje, somos expostos a frequentes mudanças no mundo real. Perante tal cenário, torna-se crucial que o *software* dê resposta às novas necessidades impostas pelo contexto do problema ao qual está associado.

A evolução de *software* torna-se assim imperativa pois, caso não se implementem novas funcionalidades ou não se resolvam as falhas identificadas, a qualidade do *software* ficará comprometida e, por conseguinte, o tempo de vida útil do mesmo pode ficar limitado. É sabido

¹ Locais de armazenamento de pacotes de *software*, os quais podem ser recuperados e armazenados num computador. São usados na gestão do desenvolvimento dos projetos de *software*. Exemplo: CVS (*Concurrent Version System*).

² <http://2013.msrfconf.org>.

que, o envelhecimento do *software* é algo incontornável³, todavia podemos atenuar este processo caso se identifiquem e se estudem atempadamente as suas causas (geralmente, falhas na atualização de novos requisitos) [4, 5].

No entanto, o mais recomendável é realizar-se uma análise cuidada do *software*, logo no início de um projeto, pois muito embora não se consigam prever com rigor as eventuais mudanças futuras, tudo se tornará mais fácil caso haja uma identificação atempada das partes de *software* mais suscetíveis a alterações [4-6].

A evolução de sistemas informáticos tem vindo a evidenciar a importância de repositórios de projetos de *software*, onde se guardam grandes volumes de dados sobre a evolução desses projetos, que servem de base ao processo de mineração.

A informação histórica armazenada em repositórios de *software* fornece uma oportunidade para se estudar projetos e produtos de grande complexidade sem interferir com os processos de desenvolvimento: sistemas de controlo de alterações no código fonte, sistemas de rastreio de defeitos, comunicações arquivadas entre a equipa de desenvolvimento do projeto, etc. Estes dados históricos estão disponíveis em muitos projetos de *software*⁴ e representam um registo detalhado do desenvolvimento histórico de um sistema de *software*. Estas informações poderão ser muito úteis aos programadores e aos gestores de projetos para construir e manter sistemas de *software* complexos, beneficiando assim as organizações [7].

As organizações tornaram-se muito competitivas e, no nosso quotidiano, o recurso apropriado à informação veio permitir ganhos significativos, pois o conhecimento passou também a ser considerado ferramenta de poder [8].

Nesta dissertação pretendemos estudar modelos de previsão mais adequados para a distribuição no tempo do número dos pedidos de evolução futuros. O nosso foco está, portanto, na previsão de quantos pedidos de evolução vamos ter num prazo de um ano, e na forma como esses pedidos se distribuem ao longo desse ano. Esta informação será essencialmente interessante para os gestores do processo. O estudo deste problema é importante porque prevendo o número de pedidos de evolução futuros, os gestores do processo poderão fazer um planeamento oportuno dos recursos necessários para responder a esses pedidos.

3 Pode ser causado por uma falha cometida pelo proprietário do sistema de *software* no caso de não proceder a alterações no produto para atender às necessidades atuais ou pode ser resultado das alterações efetuadas (degradação do desempenho, aumento da ocorrência de falhas, etc.). Estas duas causas podem levar a um rápido declínio no valor de um produto de *software*.

4 Exemplo: repositório *sourceforge* (repositório de código-fonte aberto).

1.2. Descrição do problema

O problema que está na base desta investigação consiste, então, na necessidade de previsão da distribuição dos pedidos de evolução que serão recebidos ao longo de um período relativamente alargado (até um ano), de modo a se poder planear atempadamente os recursos necessários para fazer face a esses pedidos.

Com este trabalho pretendemos realizar um estudo recorrendo a informação contida em sistemas de repositórios de pedidos de evolução do *software Eclipse*⁵, *Mozilla*⁶ e *Gnome*⁷ para criar modelos de previsão da distribuição desses pedidos ao longo do tempo.

A abordagem a utilizar visa estudar que tipos de modelos são mais adequados, conforme o volume de dados históricos existente no repositório a que diz respeito. Pretende-se saber concretamente:

- Se a escolha do “melhor” modelo de previsão é relativamente estável ou demasiado volátil, o que poderá implicar, ou não, uma frequente atualização dos modelos.
- Se os modelos que agregam informação sazonal se tornam dominantes? No caso afirmativo, será interessante averiguar quanto tempo de dados históricos é necessário para que a informação sazonal seja relevante?

1.3. Solução apresentada

A solução para o problema apresentado na secção 1.2. irá basear-se na recolha de informação importante de sistemas de rastreio de defeitos (e.g. *Bugzilla*⁸), na utilização de uma ferramenta para geração automática de modelos de previsão que a seguir vamos testar; na produção de métricas sobre a precisão dos modelos de previsão a construir⁹, na aplicação de uma ferramenta estatística de análise de previsões denominada SPSS¹⁰, tendo como objetivo principal a obtenção da previsão do número de pedidos de evolução futuros. Para melhor exemplificar a solução descrita, vejamos a Figura 1.1.

⁵ www.eclipse.org

⁶ www.mozilla.org

⁷ www.gnome.org

⁸ www.bugzilla.org

⁹ Será efetuada uma comparação sistemática de várias medidas desses modelos com base no erro relativo e na capacidade de explicação.

¹⁰ *Statistical Package for the Social Sciences* – www.spss.com

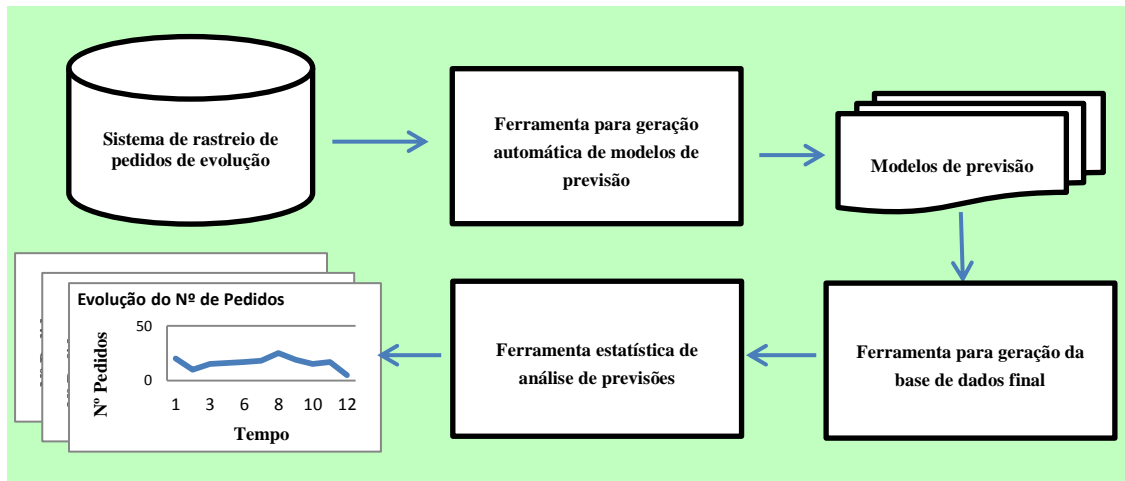


Figura 1.1 - Solução para capacidade de previsão da distribuição dos pedidos de evolução

1.4. Principais contribuições previstas

Sendo este estudo relacionado com a mineração de repositórios de *software* as principais contribuições previstas são as seguintes:

- Suporte de ferramentas para gerar uma análise semiautomática. Concretamente, vamos desenvolver uma ferramenta geradora do *script SPSS* que depois é usado para construir os vários modelos, com a ferramenta SPSS. Construiremos uma outra ferramenta para ler os dados dos ficheiros em XML que forem gerados pelo SPSS com os dados dos modelos simulados, agregando-os para análise posterior novamente com o SPSS.
- Construção de modelos de previsão para a distribuição de pedidos de evolução, com e sem informação sazonal.
- Análise comparativa de diferentes tipos de previsão.

1.5. Estrutura do relatório

O estudo que apresentamos encontra-se organizado da seguinte forma: introdução, desenvolvimento (enquadramento teórico e estudo empírico) e considerações finais acerca da investigação realizada.

- ▶ A **introdução** aborda o problema, objeto de estudo, que de forma simplificada, procura identificar, fundamentar e caracterizar o estudo.
- ▶ A **primeira parte** diz respeito ao enquadramento teórico e é constituída por dois capítulos, onde é descrita a problemática ao nível concetual com base num referencial teórico.

- O **segundo capítulo** - “Contexto”- apresenta alguns conceitos que servem de base à investigação;
 - O **terceiro capítulo** - “Trabalho relacionado” - reflete a linha de pensamento de alguns autores.
- ▶ A **segunda parte** abrange o estudo empírico e é composta por quatro capítulos, onde são apresentados a metodologia usada no estudo, os resultados, a interpretação dos mesmos e as conclusões do estudo.
- O **quarto capítulo** - “Metodologia da investigação” - inicia-se com a descrição do contexto e do problema que está na base desta investigação, seguida da apresentação do planeamento experimental e da sua execução;
 - O **quinto capítulo** - “Resultados” - apresenta a descrição dos resultados obtidos no trabalho experimental;
 - O **sexto capítulo** - “Interpretação dos resultados” - apresenta a interpretação dos resultados obtidos;
 - **Referências bibliográficas** - descrevem todos os artigos, revistas e publicações *on-line* consultados.
 - **Apêndices** – Apresentam os documentos produzidos pela investigadora e que servem de suporte à realização do estudo empírico.

PARTE I – ENQUADRAMENTO TEÓRICO



Contexto

Neste capítulo são apresentados alguns conceitos que servem de base à investigação, a saber: evolução de *software*, repositórios de *software* e análise de séries temporais. A importância destes conteúdos assenta no facto da evolução de *software* justificar os pedidos de evolução de *software* a serem estudados, dos repositórios de *software* serem as fontes de informação a serem utilizadas e da análise de séries temporais ser o meio para atingir o fim desta investigação.

2.1. Evolução de *Software*

A evolução de *software* tem sido uma área de grande interesse quer para os investigadores quer para os profissionais de *software*, pois é inevitável a ocorrência de mudanças ao longo do tempo para assegurar a utilidade e a qualidade do mesmo. Tais mudanças podem surgir da identificação de defeitos no *software*, do levantamento de novos requisitos aquando da utilização do mesmo, da necessidade de melhorias de desempenho, de alterações no ambiente de negócio, da instalação de novos equipamentos informáticos e da sua associação ao sistema de *software*. No entanto, há que salientar a dificuldade que as organizações têm em implementar e gerir estas mudanças, pois não deixam de estar envolvidos grandes investimentos financeiros. Na verdade, grande parte do orçamento de *software* nas grandes empresas é atribuída à manutenção e evolução de *software* em detrimento do desenvolvimento de novos sistemas [4, 9].

2.1.1. Leis de *Lehman*

Após vários estudos empíricos, *Lehman* propôs um conjunto de oito leis, apresentadas no Quadro 2.1, que segundo o autor se aplicavam à evolução de todos os sistemas de *software*. Estas leis surgem assim como uma teoria unificada para a evolução de *software*. Essas leis tornaram-se um marco na Engenharia de *Software* com respeito à evolução de sistemas [10].

1. Mudança contínua	Um <i>software</i> deve ser continuamente atualizado, caso contrário tornar-se-á progressivamente menos satisfatório.
2. Complexidade crescente	À medida que um <i>software</i> é alterado verifica-se um acréscimo da sua complexidade, a não ser que seja feito trabalho para mantê-la ou reduzi-la.
3. Auto regulação	O processo de evolução de <i>software</i> é autorregulado próximo à distribuição normal em relação às medidas dos atributos de produtos e dos processos.
4. Conservação da estabilidade organizacional	A taxa média de atividade global efetiva num <i>software</i> em evolução tende a ser contante durante o tempo de vida do produto.
5. Conservação da familiaridade	Regra geral, a taxa de crescimento incremental e a taxa de crescimento a longo prazo tendem a declinar.
6. Crescimento contínuo	O conteúdo funcional de um <i>software</i> deve ser aumentado continuamente durante o seu tempo de vida, a fim de ir ao encontro das necessidades dos utilizadores.
7. Qualidade decrescente	A qualidade do <i>software</i> será entendida como declinante a não ser que o <i>software</i> seja adaptado rigorosamente às mudanças no ambiente operacional.
8. Sistema de retroalimentação	Os processos de evolução de <i>software</i> são sistemas de retroalimentação em múltiplos níveis, em múltiplos <i>loops</i> e envolvendo múltiplos agentes.

Quadro 2.1 – Leis de *Lehman*

De acrescentar que, as leis de *Lehman* parecem ser, geralmente, aplicáveis a sistemas de grande porte desenvolvidos por grandes organizações, visto que, não está devidamente claro como podem ser modificadas para pequenas organizações e para sistemas de média dimensão [10].

2.2. Repositórios de Sistemas de Software

Os repositórios de Sistemas de *Software* são considerados depósitos de informação sobre as atividades de desenvolvimento e de manutenção de um *software*. De salientar que incluem artefactos de *software*, a saber: sistemas de controlo de versões, sistemas de rastreamento de defeitos e ferramentas de comunicação pública (*mailing lists*). Tomemos como exemplo de um artefacto um ficheiro de código-fonte em C++ e como exemplo de um repositório a meta-informação referente à alteração desse artefacto, que conterà informações sobre os erros e as respetivas correções, o autor, a data de alteração, entre outras [1-3, 5,7, 11-12].

Os profissionais e investigadores de projetos de *software* constroem modelos, muitas vezes, para compreender o passado [13] e, menos frequentemente, constroem modelos de previsão. Por outro lado, estes profissionais podem ainda estudar a evolução de vários elementos dos repositórios. Nesta dissertação vamos estudar especificamente os pedidos de evolução e construir modelos de previsão da distribuição desses pedidos ao longo do tempo. Estes pedidos poderão resultar de defeitos encontrados no *software* ou de potenciais melhorias solicitadas para o sistema de *software* [3, 12-22].

2.2.1. Sistema de rastreio de defeitos

Um sistema de rastreio de defeitos tem como objetivo controlar os mesmos, não só nas diferentes fases de desenvolvimento de um projeto de *software* como também na fase de execução do mesmo [1, 12, 15, 17, 20-21]. Na prática, o seu âmbito é um pouco mais alargado, porque além dos defeitos, são também usados para registar e gerir pedidos de evolução do *software*, mesmo quando tais pedidos não resultam de defeitos (podem, por exemplo, resultar de novos requisitos que os utilizadores pretendem ver satisfeitos pelo *software*). Em sistemas de *software* de grande complexidade verifica-se a existência de um grande número de pedidos de evolução, os quais carecem de uma gestão e análise eficientes. Os sistemas de rastreio de defeitos constituem assim uma solução para simplificar a identificação e monitorização desses pedidos. Estes sistemas permitem aos utilizadores a possibilidade de reportarem os pedidos de evolução, e posteriormente acompanharem a evolução desses pedidos. Para cada pedido efetuado é armazenada informação referente à identificação do pedido, ao estado em que se encontra, à resolução do mesmo, ao nome do produto, à versão, ao *hardware*, ao nível de importância, às dependências, ao utilizador responsável pelo registo do pedido, ao utilizador responsável pela resolução do mesmo, aos utilizadores notificados, à data de submissão do pedido, à data de modificação, entre outros. Esta informação poderá ser ainda extraída para servir de base a análises específicas. Neste estudo, a informação proveniente do sistema de rastreio *Bugzilla* servirá de suporte à criação de modelos de previsão de pedidos de evolução distribuídos ao longo do tempo. Neste sentido, o que pretendemos fazer é uma espécie de extensão às capacidades de suporte à análise que este sistema nos oferece [1, 12, 15, 17, 20-21].

2.2.1.1. Bugzilla

O *Bugzilla* é um sistema de rastreio de defeitos de *software* criado no contexto de um projeto de *software* livre (produto *Mozilla*¹¹ de código aberto baseado em *web* e em *email*). É usado por

¹¹ www.mozilla.org

muitas organizações e em bastantes projetos na gestão de desenvolvimento de *software* [23, 24]. O sistema permite que os utilizadores reportem defeitos, ou façam pedidos de evolução sobre o *software*, podendo depois monitorizar a evolução do seu pedido de forma eficiente. Estas atividades são realizadas tendo em conta um modelo de relatório de defeitos exibido na Figura 2.1. Este relatório é composto por vários campos, a saber: número do defeito, estado (não confirmado, confirmado, em progresso, resolvido e verificado), resolução (não resolvido, resolvido, inválido, sem resolução, duplicado, inconclusivo), nome do produto, componente, versão, *hardware*, nível de importância, dependências, utilizador responsável pelo registo do erro, utilizador responsável pela resolução do mesmo, utilizadores notificados, data de submissão do erro, data de modificação, entre outros¹².



Figura 2.1 - Modelo de um relatório de defeitos do *Bugzilla*

2.3. Análise de séries temporais

Para analisar de modo rigoroso os pedidos de evolução submetidos ao longo do tempo ao sistema de rastreio discutido na subsecção 2.2.1., necessitamos de ferramentas de análise adequadas. Podemos olhar para a distribuição desses pedidos ao longo do tempo como uma série temporal. Uma série temporal é um conjunto de observações realizadas ao longo do tempo, seguindo uma ordem sequencial. Esta ordem implica a existência de uma dependência entre as observações vizinhas, o que torna interessante a sua análise e a respetiva modelação. De referir ainda que, as séries temporais estão presentes em diversas áreas do saber: ciências económicas, financeiras, sociais, entre outras.

A análise de séries temporais assenta no pressuposto que existe um sistema causal, relativamente constante, relacionado com o tempo, que exerceu influência sobre os dados no passado e poderá vir

¹² www.bugzilla.org

a fazê-lo no futuro. Este sistema causal, aquando da sua atuação, cria padrões não aleatórios, visíveis em gráficos de séries temporais ou em outros processos estatísticos. O objetivo da análise de séries temporais consiste então na identificação destes padrões, que poderão ser integrados num modelo estatístico, e, por conseguinte, servirão de base à realização de previsões para o comportamento futuro de uma determinada variável de interesse (variável dependente), o que facilitará a tomada de decisões. Contudo, é importante referir a dificuldade que existe em utilizar um modelo que possibilite a inclusão dos diversos padrões identificados. Para solucionar este problema, procede-se à decomposição da série temporal em padrões.

2.3.1. Métodos de decomposição

Tomemos então como referência os quatro padrões (ou componentes) que compõem todas as séries temporais, de acordo com o modelo clássico:

- *Tendência (T)*: trata-se de um comportamento de longo prazo da série, causado por um aspeto que afeta a variável de interesse no período em causa (e.g. crescimento demográfico).
- *Sazonalidade (S)*: consiste em flutuações nos valores da variável dependente, usualmente com duração inferior a um ano, e que, regra geral, se repete anualmente (em função de estações do ano, feriados, entre outros). De ressaltar que, se os dados forem registados anualmente, a sazonalidade não exercerá a sua influência na série, pois as flutuações só se repetem dentro dos anos.
- *Variações Irregulares (I)*: abrange flutuações anormais, resultantes de acontecimentos imprevistos (e.g. catástrofes naturais).
- *Variações Cíclicas (C)*: consistem em flutuações nos valores da variável de interesse, com duração superior a um ano, e que se repetem periodicamente (e.g. períodos de recessão ou de crescimento).

Importa referir que, os dois primeiros componentes são os mais utilizados devido à natureza dos dados que representam.

Visto que uma série temporal nem sempre irá exibir todos os componentes, será conveniente proceder-se à decomposição da mesma, processo esse que permitirá não só identificar os componentes que poderão estar implícitos num determinado intervalo de tempo como também possibilitará a obtenção e índices e/ou equações necessárias na realização de previsões para períodos futuros da série.

Para além dos métodos de decomposição de séries temporais, existem também os métodos simples de previsão e os métodos avançados de séries temporais.

2.3.2. Métodos simples de previsão

Relativamente aos métodos simples de previsão destacam-se os de média móvel e os de suavização exponencial (simples, linear, sazonal), também conhecidos por *Exponential Smoothing*. No quadro 2.2 apresentamos os tipos de modelos *Exponential Smoothing* [25, 26].

Tipo de modelo	Descrição
<i>Exponential Smoothing</i>	
<i>SIMPLE</i>	Este modelo deve ser usado quando os dados de séries temporais não têm tendência nem sazonalidade .
<i>BROWN</i>	Este modelo deve ser usado quando os dados da série temporal têm tendência , mas não sazonalidade .
<i>SIMPLE SEASONAL</i>	Este modelo deve ser usado quando os dados de séries temporais não têm tendência , mas têm sazonalidade .
<i>WINTERS ADDITIVE</i>	Este modelo é usado quando os dados de séries temporais têm tendência e sazonalidade . A série mostra flutuações sazonais estáveis, independentemente do nível geral da série. É apropriado para uma série temporal em que a amplitude do padrão sazonal é independente do nível médio da série, isto é, uma série de tempo exibindo sazonalidade aditiva.
<i>WINTERS MULTIPLICATIVE</i>	Este modelo é usado quando os dados de séries temporais têm tendência e sazonalidade . O tamanho das flutuações sazonais varia, dependendo do nível global das séries. Este modelo é adequado para uma série temporal em que a amplitude do padrão sazonal é proporcional ao nível médio da série, isto é, uma série de tempo que exhibe sazonalidade multiplicativa.

Quadro 2.2 – Tipos de modelos de suavização exponencial

Estes métodos efetuam a previsão do valor futuro da série temporal através da suavização das observações da variável de interesse realizadas no passado. O objetivo destes métodos consiste na identificação de um padrão básico presente nos dados históricos, e na sua posterior utilização na previsão de valores futuros, considerando para o efeito que os valores extremos da série representam flutuações aleatórias. Estes métodos são frequentemente usados para horizontes de previsão de curtos períodos.

2.3.3. Métodos avançados de previsão

Quanto aos métodos de previsão de séries temporais estacionárias mais complexos importa referir os principais modelos de processo linear geral, autorregressivos (AR), médias móveis (MA), autorregressivos e médias móveis (ARMA) e autorregressivos integrados de médias móveis (ARIMA). Estes métodos são frequentemente usados para horizontes de previsão de média ou longa duração, daí serem mais adequados ao estudo implícito nesta dissertação.

Os modelos de processo linear geral são considerados uma combinação linear ponderada, do termo atual, mais os termos passados do ruído branco (sequência de valores aleatórios independentes, com distribuição Normal e com média zero e com variância (σ_a^2)).

Nos modelos autorregressivos (AR) os valores da série representam uma combinação linear de um número finito de valores passados, mais um termo de ruído, que não é explicado por esses valores passados.

Nos modelos de média móvel (MA) os valores previstos para a observação dependem dos valores dos erros observados em cada período passado, em vez das observações propriamente ditas. Representam assim uma combinação linear de um número finito de ruídos.

Os modelos autorregressivos e de média móvel (ARMA) constituem assim a combinação dos dois modelos descritos anteriormente. Estes modelos relacionam não só os valores futuros com as observações passadas como também os erros passados apurados entre os valores reais e os previstos.

Os modelos ARIMA representam o processo estocástico gerador da série temporal, a partir de um modelo ARMA adequável a séries temporais estacionárias e estendendo o conceito para séries temporais não estacionárias. Notação utilizada para este modelo: ARIMA (ps, ds, qs). Estes parâmetros incluem a característica de sazonalidade. Estes modelos de previsão obedecem a uma metodologia, conhecida por metodologia *Box-Jenkins*, que inclui seis fases:

- *Análise exploratória dos dados*: observação do comportamento geral da série (tendência, sazonalidade, variações irregulares, ciclos) através da representação gráfica da série inicial. Nesta fase será verificada a necessidade de estacionaridade da série.
- *Estacionaridade*: são aplicadas transformações (diferenciações) geralmente lineares à série temporal com objetivo de retirar à série inicial características de tendência, sazonalidade, entre outras.

- *Identificação*: análise do correlograma e do correlograma parcial para obtenção dos parâmetros p (autorregressivo), d (diferenciação) e q (média móvel). A notação utilizada para este modelo é a seguinte ARIMA (p,d,q). Por Exemplo, um modelo ARIMA (1,1,1) significa que contém um parâmetro autorregressivo (p) e um parâmetro de média móvel (q), que foram calculados depois da série ter sido diferenciada uma vez ($d=1$). Nesta fase estuda-se o comportamento destas funções em cada tipo de modelo para a identificação do modelo mais adequado.
- *Estimação paramétrica*: utilização de métodos de máxima verossimilhança ou do tipo mínimos quadrados. Estes métodos incluem as auto correlações e as auto correlações parciais empíricas da série.
- *Validação*: verificação da adequação do modelo escolhido, em termos da qualidade estatística do modelo estimado (testes aos parâmetros) e da análise dos resíduos da estimação. Se o modelo for válido os resíduos deverão apresentar propriedades compatíveis com o ruído branco (sequência de erros aleatórios). Caso o modelo seja considerado válido, far-se-á a previsão, caso contrário repetir-se-ão as fases anteriores.
- *Previsão*: obtenção de previsões de qualidade para um horizonte temporal definido, a partir de um determinado momento.

A notação completa dos modelos ARIMA envolve seis parâmetros que se representam da seguinte forma: ARIMA (p, d, q) (ps, ds, qs). A sazonalidade poderá estar presente apenas nos últimos três parâmetros (ps – parâmetro autorregressivo sazonal, ds - diferenciação sazonal e qs - média móvel sazonal).

Os modelos acima descritos dizem respeito a uma única série temporal. Todavia, existem metodologias nas quais a série temporal em causa é explicada através do seu passado histórico e também por outras séries temporais não correlacionadas entre si. Tais metodologias são denominadas funções de transferência, e abrangem mais do que uma série temporal, sendo sempre assegurada a relação de causalidade entre elas. Os conceitos de séries temporais apresentados anteriormente podem ser analisados em mais pormenor em [3, 14, 18, 27-29].



Trabalho Relacionado

Neste capítulo são expostas algumas considerações sobre outros trabalhos realizados no âmbito da problemática que serve de base a este estudo.

3.1. Estudos sobre ferramentas de extração de dados em sistemas de repositórios de *software*

A criação de modelos de previsão da distribuição de pedidos de evolução de *software*, ao longo do tempo, com base em informação recolhida em sistemas de repositórios, oferece uma ferramenta muito útil para a gestão da atividade dos membros da equipa de desenvolvimento [19].

Depois da extração da informação dos repositórios de *software*, é necessário a sua inserção em modelos para relacionar os diferentes tipos de informação, tais como, histórico de versões e defeitos.

Cubranic et al. exibem um estudo acerca da abordagem *HipiKat*¹³ que consiste na formação de um grupo de memória do projeto, possibilitando assim a recomendação das partes apropriadas de memória para os elementos recém-chegados à equipa de trabalho como o objetivo de melhorar as tarefas de desenvolvimento de *software*. Este modelo, para além de integrar os dois tipos de informação acima mencionados, contempla também informação referente à documentação armazenada nos *sites* dos sistemas de *software* em análise e ao histórico de *emails* [32].

Hassan fez um estudo que consistiu na exploração da mineração de dados armazenados em repositórios de *software* para apoiar os programadores e gestores de projeto de *software* no esforço para construir e manter sistemas de *software* complexos. Um dos seus contributos consistiu na apresentação de um extrator de evolução de *software* designado *C-Rex* [7].

¹³ Suporta os repositórios: *CVS* e *Bugzilla*.

3.2. Estudos sobre ferramentas de visualização e análise de informação

Para se compreender melhor estes modelos e sobretudo a evolução dos sistemas de *software* a que dizem respeito, os investigadores desta área tiveram necessidade de desenvolver técnicas e ferramentas de visualização e análise de informação.

D' Ambros apresentou em [1] uma abordagem baseada nas dependências históricas e nos defeitos de *software* recorrendo à técnica *Evolution Radar*. Segundo o autor estas informações foram importantes no desenvolvimento da aprendizagem sobre os sistemas de *software* e na deteção de possíveis problemas existentes no código fonte. Acrescentou ainda que, com base em falhas detetadas no código fonte é possível prever a localização de erros futuros e, como tal, concentrar as atividades de manutenção nas partes do sistema mais suscetíveis a erros. A abordagem foi testada em várias versões de três sistemas de *software* (*Eclipse JDT Core*, *ArgoUML*, *AspectJ*), e concluiu-se que a existência de desarmonias nos projetos representava um bom meio para prever defeitos, obtendo um poder de previsão superior a outras abordagens.

Fischer et al. descreveram em [2] uma ferramenta *EvoGraph*, baseada em diagramas, adequada para extrair rapidamente as entidades mais relevantes do histórico de versões de um sistema. Os autores evidenciam que tal abordagem fornece um *feedback* visual acerca da evolução das dependências estruturais. Trata-se de uma abordagem eficiente, pois apenas são inspeccionadas porções de código alteradas. Os autores usaram anti-padrões denominados “*Day-Fly*”¹⁴ e “*Daisy-Chain*”¹⁵ que, segundo afirmam neste estudo, facilitam a avaliação da estabilidade estrutural.

Pinzger et al. apresentaram em [16] uma técnica de visualização denominada *Relvis* que permite a disponibilização de vistas gráficas (diagramas *Kiviat*) que integram o código-fonte e o histórico de dados das várias versões (mais valia em relação a outras técnicas de visualização existentes). Segundo os autores o desenvolvimento desta técnica veio dar resposta à análise de modelos de dados de grande complexidade, provenientes de sistemas de *software* de grande dimensão. A justificação assenta no facto desta técnica contemplar a separação do volume de dados e o seu mapeamento em dois diagramas *Kiviat* (um para as métricas dos nós e outro para as métricas de relações de união). Estes diagramas exibem o estado corrente da implementação.

D' Ambros descreveu ainda em [30, 31] uma ferramenta denominada *BugCrawler* que suporta a evolução de *software* e a engenharia reversa. Trata-se de uma ferramenta baseada numa combinação de métricas de *software* e de visualizações interativas. O *BugCrawler* integra

¹⁴ Indica uma dependência estrutural que é introduzida e novamente removida no mesmo *time-slot* ou depois de um pequeno número de revisões.

¹⁵ Indica uma dependência estrutural que primeiro é usada num ficheiro e posteriormente usada exclusivamente num segundo ficheiro.

informação estruturada a partir de código fonte com informação evolutiva obtida a partir de ficheiros *Log* (CVS - *Concurrent Version System*) e relatórios de problemas *Bugzilla*. Segundo o autor, esta ferramenta é válida para vários projetos de grande porte. Consideramos importante a sua utilização no nosso estudo, visto que trataremos também de defeitos de *software* em sistemas de grande complexidade.

No Quadro 3.1 encontra-se um resumo das ferramentas e técnicas apresentadas. As características suportadas por estas ferramentas e técnicas estão representadas por um X.

		<i>Evolution Radar</i>	<i>Evograph</i>	<i>Relvis</i>	<i>BugCrawler</i>
Adequação	Séries temporais	X			X
Atividades (Dados)	Visualização	X	X	X	X
	Exploração	X			
Repositórios	CVS	X		X	X
	SVN	X			X
	<i>Bugzilla</i>				X
	<i>Issuezilla</i>				X
Artefactos	Pacotes	X			
	Ficheiros	X		X	X
	Classes	X	X	X	
Avaliação da evolução	Código Fonte	X	X	X	
	Dependências	X			

Quadro 3.1 – Ferramentas e técnica de análise de evolução apresentadas

3.3. Estudos sobre previsão de pedidos de evolução nos sistemas de *software*

Foquemo-nos agora na importância da previsão nos sistemas de *software*. No desenvolvimento de projetos de *software* estão envolvidos muitos recursos que estão associados inevitavelmente ao tempo e aos custos limitados. Como tal, é crucial que o gestor de projeto faça uma gestão eficiente desses recursos para que a qualidade do *software* não seja colocada em causa. Apesar da experiência do gestor de projeto ser importante, não é suficiente. Os sistemas de rastreio de defeitos constituem um bom contributo para estes profissionais, pois contém informação sobre as falhas em diversos artefactos (pacotes, ficheiros, classes, etc). Tendo acesso ao histórico de defeitos, os gestores de projetos poderão estudar a evolução do *software* com base no passado e obter previsões do número de pedidos de evolução futuros. Esta temática tem suscitado interesse por parte de investigadores da área. A seguir, apresentamos algumas reflexões sobre este assunto.

Nagappan et al. fizeram um estudo empírico acerca do histórico de defeitos *post-release* (após lançamento) em cinco sistemas de *software* da *Microsoft*, e descobriram que as entidades sujeitas a

falhas de *software* são estatisticamente correlacionadas com medidas de complexidade de código. Todavia, os autores consideraram que não existia um único conjunto de métricas de complexidade responsável pela previsão mais otimizada. Os autores acrescentaram ainda que, recorreram à análise de componentes principais das métricas de código para construção de modelos de regressão a fim de prever com precisão a probabilidade de defeitos *post-release* para novas entidades [17].

Zimmerman et al. realizaram um estudo sobre defeitos do *Eclipse* e sobre a sua localização no código-fonte. Estes investigadores construíram modelos de regressão linear para o conjunto de defeitos definido para prever se os ficheiros / pacotes apresentavam defeitos pós-lançamento. Segundo os autores, as experiências realizadas mostraram que a combinação de métricas de complexidade pode efetivamente prever defeitos, sugerindo que quanto mais complexo for o código, mais defeitos apresentará [20].

Moser et al. realizou uma análise comparativa de dois conjuntos diferentes de métricas para a previsão de defeitos, a saber: métricas de código-fonte e métricas de processo. A experiência consistiu na construção de três modelos: um modelo de alteração, usando apenas métricas de alterações propostas pelos autores; um modelo de código, utilizando métricas de código estático; e, por fim, um modelo combinado, utilizando ambos os tipos de métricas anteriores para prever a presença ou ausência de defeitos em ficheiros. Os resultados indicaram que, para os dados do *Eclipse*, as métricas de processo são mais eficientes para a previsão de defeitos do que as métricas de código [21].

Os estudos apresentados anteriormente focaram aspetos que iremos abordar na nossa investigação, que assenta na capacidade de previsão da distribuição do número de pedidos de evolução de *software* (*Eclipse*, entre outros). No entanto, outros investigadores estudam a evolução de outros pedidos, também contemplados no nosso trabalho, nomeadamente: a implementação de novas funcionalidades ou melhorias. Apresentamos a seguir um estudo que descreve esta problemática. *Tsantalis et al.* descreveram em [15] uma abordagem probabilística para estimar a propensão de alteração de um projeto orientado a objetos, avaliando a probabilidade de que cada classe do sistema será afetada quando uma nova funcionalidade foi adicionada ou quando funcionalidade existente é modificada. O modelo proposto foi avaliado em dois projetos de código aberto denominados *JFlex* e *JMOL*. O processo foi completamente automatizado por um programa *Java*, e a análise estatística revelou correlação melhorada entre as probabilidades extraídas e as mudanças reais verificadas em cada uma das classes, em comparação com um modelo de previsão que se baseia apenas em dados anteriores. Neste caso, os autores tentam prever a localização das alterações (em que classe elas vão suceder), contrariamente ao que estamos a propor no nosso estudo.

MacDonell et al. realizaram um estudo com o objetivo de explorar o impacto da utilização dos dados provenientes das fases de levantamento de requisitos, de desenho e de implementação na estimação de esforços de previsão na fase posterior. Os autores pretenderam mostrar os benefícios da re-estimação recorrendo a dados de fases concluídas do projeto. Usaram um método que consistia na aplicação de um pequeno número de dados industriais de dezasseis projetos de *software* de uma única organização para comparar modelos de previsão através de uma abordagem conscienciosa, baseada numa análise mais tradicional denominada “*Leave one out*”. Concluíram que a falha de não tomar em consideração a natureza temporal dos dados conduz a estimativas não fiáveis da eficácia da sua previsão e que os dados obtidos antes da fase de esforço podem ser usados para melhorar a gestão de tarefas do processo seguinte [19].

Fuentetaja et al. apresentaram em [3] uma abordagem inovadora que consistiu na aplicação de técnicas de análise de séries temporais para dados históricos do crescimento de sistemas de *software*, tais como: *r/s statistic*¹⁶ e DFA (*Detrended Fluctuation Analysis*). Os resultados preliminares obtidos acusaram a presença de anti correlações nos dados, que validaram algumas leis da evolução de *software* (3ª e 8ª leis). Os autores consideraram também que um *software* em produção está sujeito a forças internas e externas que exigirão uma sucessão interminável de correções, melhorias e extensões de funcionalidade do sistema.

Importa ainda salientar que, para que a evolução dos pedidos de evolução de *software* seja compreendida mais facilmente não podemos esquecer dos diversos métodos para auxiliar na tarefa de previsão de séries temporais, como por exemplo, modelos de: suavização exponencial, auto-regressivos (AR), médias móveis (MA) e modelos ARIMA [3, 14, 18, 27-29]. De seguida são apresentados alguns estudos sobre o assunto.

Goulão *et al.* apresentaram em [18] um estudo acerca de um modelo de previsão de pedidos de evolução de um *software open-source* designado *Eclipse*. Os autores procederam à construção de um modelo de previsão ARIMA, que incluiu a identificação de padrões sazonais e tendências. Este modelo foi validado através da evolução de pedidos de alteração para os doze meses seguintes. Concluíram que o uso da informação sazonal melhorou significativamente a capacidade de estimação do modelo quando comparado com outros modelos ARIMA. De referir que, a nossa investigação tem uma base de trabalho semelhante. A motivação para a primeira parte do trabalho é que a construção do modelo de previsão deve ser simplificada. Como é relativamente simples automatizar a construção dos modelos e a comparação das métricas é muito mais fácil do que o reconhecimento de padrões nos gráficos, simplesmente por mera observação, queremos tentar tornar o processo de construção de modelos acessível a pessoas com menor *background* de

¹⁶ Aplicada ao conjunto de dados a partir de medições temporais. Aplicação do modelo empírico Lei da Potência (relação entre a variação do intervalo de dados com escalas de tempo diferentes).

estatística, ao mesmo tempo que garantimos que no final o modelo é tão bom ou melhor do que o que seria encontrado por um perito.

Raja et al apresentaram um estudo de previsões de defeitos de *software* em séries temporais, para lidar com a natureza temporal de ocorrências dos mesmos. Os autores focaram-se em modelos simples de usar e menos dispendiosos em termos de requisitos e de dados. Recorreram também à técnica ARIMA. Os autores afirmaram que a autocorrelação significativa existente entre o número de defeitos reportados durante a evolução do *software* viola a suposição de independência crítica para modelos de previsão causais. Como tal, a sua pesquisa corrigiu a autocorrelação, evitando assim a distorção dos resultados dos modelos de previsão de defeitos de evolução do *software* [22]. Outra variação importante no nosso trabalho é que enquanto que nos estudos anteriores é sempre feita uma previsão num determinado momento, no nosso é criada uma série temporal de previsões, o que muda radicalmente a análise que é feita a seguir. Em vez de se olhar para a melhor série num determinado instante, estamos a olhar para a evolução dos diferentes modelos ao longo do tempo.

PARTE II – ESTUDO EMPÍRICO



Metodologia da Investigação

Neste capítulo são descritas as opções metodológicas subjacentes ao estudo empírico realizado.

4.1. Motivação

4.1.1. Contexto e descrição do problema

O contexto desta investigação traduz-se na importância da tomada de decisões por parte de um gestor de projetos, a qual influenciará o processo de desenvolvimento do *software* no futuro, em termos de recursos humanos, financeiros e materiais. Assim sendo, são muitos os investigadores que têm apresentado estudos sobre técnicas, metodologias e ferramentas a utilizar no âmbito da análise e previsão da evolução do *software*, com a intenção de promover uma gestão do processo de evolução de *software* mais eficiente.

O problema que motiva esta investigação consiste, então, na necessidade de previsão da distribuição dos pedidos de evolução que serão recebidos ao longo de um período relativamente alargado (até um ano), de modo a poder planear atempadamente os recursos necessários para fazer face a esses pedidos.

4.1.2. Objetivos do trabalho

Este trabalho de investigação teve como grande objetivo geral a discussão do processo de escolha e construção do melhor modelo de previsão de distribuição de pedidos de evolução de *software* ao longo do tempo. É normal que, à medida que novos dados vão ficando disponíveis, queiramos verificar se há necessidade de substituir, ou re-calibrar, os nossos modelos de previsão. *Boehm* sugeriu que, há medida que vão sendo conhecidos mais dados sobre um projeto e ele se aproxima da sua conclusão (a proximidade da conclusão era medida como a percentagem do esforço no projeto que faltava realizar), a capacidade de previsão sobre o mesmo vai aumentando [39]. Este

fenômeno, conhecido como “cone de incerteza”, foi extensivamente discutido em [40, 41], sendo detetáveis fenômenos como enviesamentos sistemáticos de sobrestimação, ou subestimação. Nesta dissertação o cenário foi diferente, dado que tratámos de previsões para pedidos de evolução em projetos para os quais não existe uma data final prevista. Isto implica que o elemento dominante na capacidade de previsão foi a informação acumulada, e não a percentagem de completude do projeto global, que era sempre desconhecida nos nossos casos de estudo. Nos casos aqui estudados, a atualização de informação teve uma periodicidade mensal. Ou seja, a cada mês passado, reavaliámos os modelos de previsão. Avaliámos até que ponto eles eram voláteis (levando à necessidade de os atualizar), ou estáveis (que, pelo contrário, nos permitiria relaxar o ritmo de atualização das previsões). A sua pertinência é justificada pelos constrangimentos vividos pelos gestores de projetos e pelos restantes membros da equipa de análise e desenvolvimento de *software* no cumprimento de prazos e de tarefas de evolução de *software*.

4.1.3. Natureza da investigação

A investigação que dá corpo a este trabalho configura um estudo experimental, descritivo e interpretativo, tendo como fim identificar o melhor modelo de previsão referente ao número de pedidos de evolução de *software*, permitindo assim enunciar pontos fortes e fracos associados ao processo.

Analisámos modelos de previsão gerados para oito projetos. Por uma questão de economia de espaço, apresentamos com maior detalhe a análise referente ao *software Eclipse JDT*, sendo os dados referentes aos restantes sete projetos apresentados de modo mais sumário. O processo aqui descrito foi completamente replicado nos restantes casos. Para atingir os objetivos deste estudo, construímos modelos de previsão da distribuição de pedidos de evolução para um período relativamente alargado (12 meses).

A abordagem realizada teve também como meta estudar que tipos de modelos são mais adaptáveis, tendo em conta o volume de dados histórico existente, correspondente ao repositório selecionado. A abordagem foi experimentada com repositórios de sistemas reais complexos com vários anos de evolução, a saber:

- *Eclipse JDT (Java Development Tooling)*: ferramentas de desenvolvimento *Java*, que inclui três componentes essenciais: *JDT Core*¹⁷, *JDT UI*¹⁸, *Debug JDT*¹⁹.
- *Eclipse Platform*²⁰: inclui o conjunto de estruturas e de serviços comuns que constituem a infraestrutura principal do *Eclipse*.

¹⁷ org.eclipse.jdt.core

¹⁸ org.eclipse.jdt.ui

¹⁹ org.eclipse.jdt.launching

- *Eclipse Android*²¹: Plug-in para o IDE *Eclipse*²² (ADT - *Android Development Tools*) dedicado ao desenvolvimento de aplicações para o sistema operativo móvel *Android*.
- *Eclipse*²³: inclui todas aplicações associadas ao ciclo de desenvolvimento de *software*, aos sistemas de gestão de conteúdos, entre outras.
- *Mozilla*²⁴ (*Core e FireFox*): é um navegador *web* gratuito.
- *GNOME*²⁵ (*Evolution e GStreamer*): consiste numa plataforma de desenvolvimento de *software* livre.

Os dados extraídos destes repositórios de *software* que serviram de base a este estudo já se encontravam agregados, à exceção dos dados do *Eclipse* [18]. O trabalho prático assentou no estudo da evolução dos modelos, que consistiu na deteção de eventuais padrões que pudessem emergir da comparação dos diferentes modelos, para cada um dos repositórios estudados.

4.1.4. Questões de investigação

Neste trabalho foi utilizada a abordagem *Goal Question Metric* (GQM). Trata-se de uma abordagem orientada por objetivos para medição de produtos e processos de *software*. É realizada em duas partes: a primeira inclui uma definição *top-down* de objetivos, de questões de investigação correspondentes e de métricas adequadas; a segunda implica uma análise e interpretação *bottom-up* dos resultados obtidos referentes às métricas definidas, para dar resposta às questões de investigação, a fim de se atingirem os objetivos [36, 37, 38]. Na Figura 4.1 são exibidos os objetivos, geral e específicos, deste estudo, assim como as questões de investigação.

²⁰ <http://wiki.eclipse.org/Platform>

²¹ www.android.com

²² <http://www.ibm.com/developerworks/opensource/tutorials/os-eclipse-androidwidget/index.html?ca=dgr-eclipse-1>

²³ <http://www.eclipse.org/projects/listofprojects.php>

²⁴ www.mozilla.org

²⁵ *GNU Network Object Model Environment* - <http://www.gnome.org>

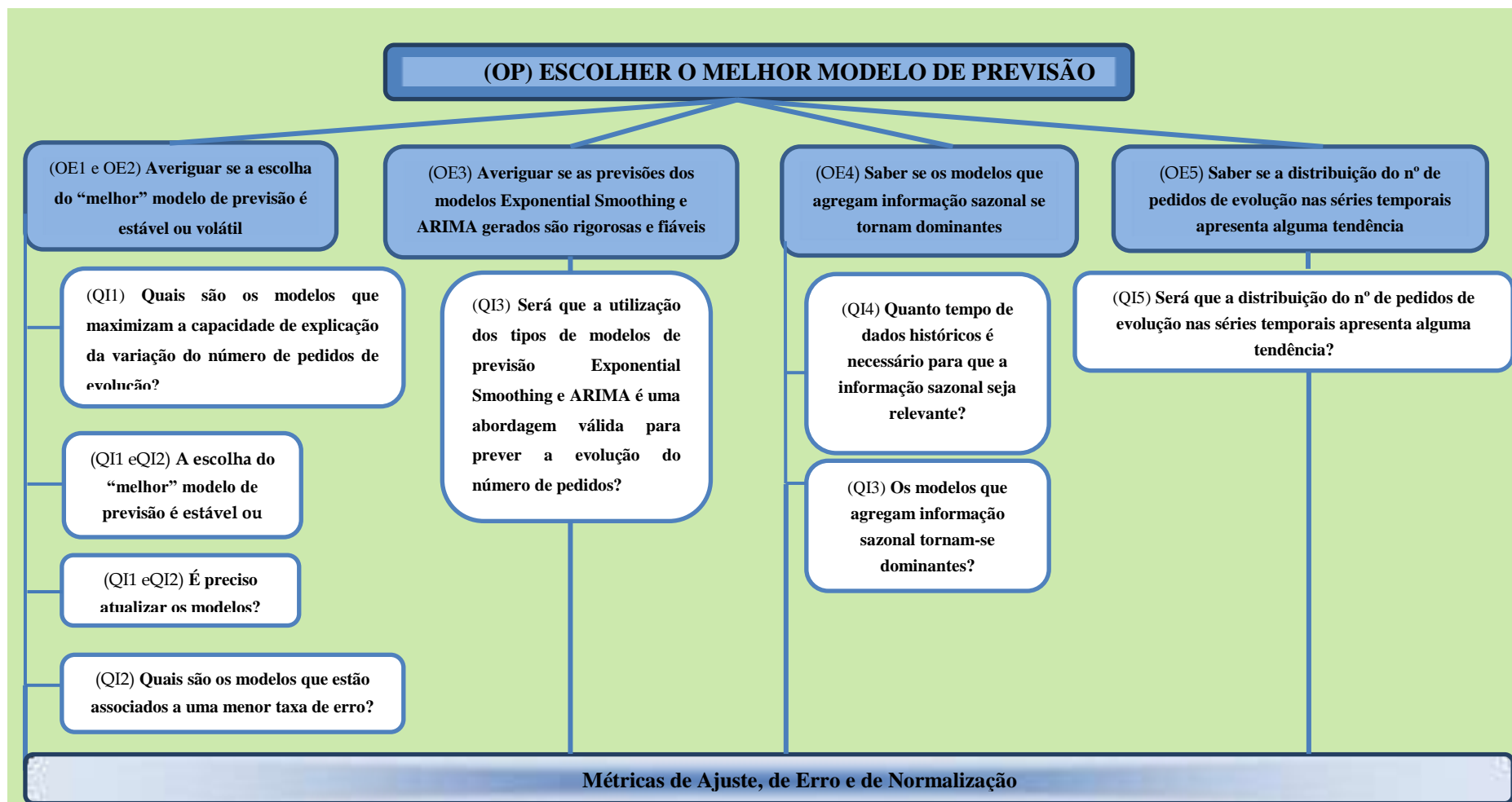


Figura 4.1 – Padrão GQM aplicado ao problema

No Quadro 4.1 são apresentadas as questões e as métricas que serviram de base ao objetivo principal desta investigação.

OP/ QP	Objetivo/ Questão principal	Escolher o melhor modelo de previsão de distribuição de pedidos de evolução de <i>software</i> ao longo do tempo. Qual é o tipo de modelo que melhor se adequa à previsão dos pedidos de evolução?
QI1	Questão de investigação 1	Quais são os modelos que maximizam a capacidade de explicação da variação do número de pedidos de evolução? Serão realizados cálculos para averiguar as medidas de ajuste apresentadas pelos modelos de previsão. Na verdade, deseja-se saber se a escolha do “melhor” modelo de previsão é estável ou volátil. No caso de a escolha ser volátil será necessário atualizar os modelos.
	Métricas	Medidas de ajuste (Quadro 8)
QI2	Questão de investigação 2	Quais são os modelos que estão associados a uma menor taxa de erro? Serão realizados cálculos para averiguar as taxas de erro exibidas pelos modelos de previsão.
	Métricas	Medidas de erros (Quadro 9).
QI3	Questão de investigação 3	Será que a utilização dos tipos de modelos de previsão <i>Exponential Smoothing</i> e <i>ARIMA</i> é uma abordagem válida para prever a evolução do número de pedidos? Serão analisados os pedidos de evolução nestes tipos de modelos e serão realizados cálculos para averiguar a capacidade de explicação dos modelos, o erro por eles cometido e o ajuste do modelo, pesado com a sua parcimónia.
	Métricas	Medidas de ajuste, de erro e de normalização (Quadros 8, 9, 10).
QI4	Questão de investigação 4	Quanto tempo de dados históricos é necessário para que a informação sazonal seja relevante? Será observado se a série temporal apresenta algum padrão sazonal e, em caso afirmativo, identificar a causa do mesmo. Deseja-se ainda saber se os modelos que agregam informação sazonal se tornam dominantes.
	Métricas	Medidas de ajuste, de erro e de normalização (Quadros 8, 9, 10).
QI5	Questão de investigação 5	Será que a distribuição do número de pedidos de evolução nas séries temporais estudadas apresenta alguma tendência específica? Será analisado se a série temporal apresenta alguma tendência.
	Métricas	Medidas de ajuste, de erro e de normalização (Quadros 8, 9, 10).

Quadro 4.1 – Modelo GQM – Questões e métricas de investigação

4.2. Planeamento Experimental

4.2.1. Objetivos experimentais

Os objetivos experimentais abaixo apresentados, etiquetados como OE_i, são uma redefinição dos objetivos de investigação expostos na subsecção 4.1.2. Cada objetivo OE_i corresponde à respetiva questão de investigação Q_i, proposta na subsecção 4.1.4.

- **OE1:** É nosso objetivo identificar quais os modelos de previsão que maximizam a capacidade de explicação do número de pedidos de evolução para cada sistema, com a finalidade de nos permitir analisar a evolução desses modelos nesses sistemas, em busca de padrões recorrentes. Na realidade, pretende-se saber se a escolha do “melhor” modelo de previsão é estável ou volátil.
- **OE2:** É nosso objetivo verificar quais os modelos de previsão que apresentam uma menor taxa de erro do número de pedidos de evolução para cada sistema, com a finalidade de nos permitir analisar a evolução desses modelos nesses sistemas, em busca de padrões recorrentes.
- **OE3:** É nosso objetivo averiguar se as previsões dos modelos *Exponential Smoothing* e ARIMA gerados são rigorosas e fiáveis, com a finalidade de credibilizar os resultados obtidos e as técnicas utilizadas.
- **OE4:** É nosso objetivo examinar o período temporal de dados históricos com a finalidade de caracterizar a sua evolução, procurando padrões sazonais relevantes. Pretende-se também saber se os modelos que agregam informação sazonal se tornam dominantes.
- **OE5:** É nosso objetivo avaliar a distribuição do número de pedidos de evolução com a finalidade de caracterizar a sua evolução, procurando tendências específicas.

4.2.2. Unidades experimentais

No nosso trabalho experimental, usámos dados históricos recolhidos de sistemas de rastreio de pedidos de evolução (e.g. *Bugzilla*) referentes aos 8 *softwares* apresentados na subsecção 4.1.3. Os pedidos de evolução foram agregados com uma granularidade mensal.

4.2.3. Material experimental

Neste estudo utilizámos material experimental que corresponde a 8 conjuntos de dados históricos sobre pedidos de evolução de cada um dos seguintes projetos, respeitantes aos seguintes períodos de tempo:

- *Eclipse JDT*: de outubro de 2001 a setembro de 2007;
- *Eclipse Platform*: de outubro de 2001 a agosto de 2008;
- *Eclipse*: de novembro de 2001 a março de 2010;
- *Eclipse Android*: de novembro de 2007 a setembro de 2011;
- *Gnome Evolution*: de maio de 2000 a julho de 2008;
- *Gnome GStreamer*: de abril de 2002 a agosto de 2008.
- *Mozilla Core*: de outubro de 2001 a setembro de 2007;
- *Mozilla Firefox*: de abril de 2001 a julho de 2008.

4.2.4. Tarefas

As tarefas que foram realizadas neste trabalho são apresentadas no diagrama de atividades exibido na Figura 4.2. Este diagrama obedece à notação do UML 2.0 (*Unified Modeling Language*) no que diz respeito à forma de representação das atividades e dos dados consumidos na execução das mesmas.

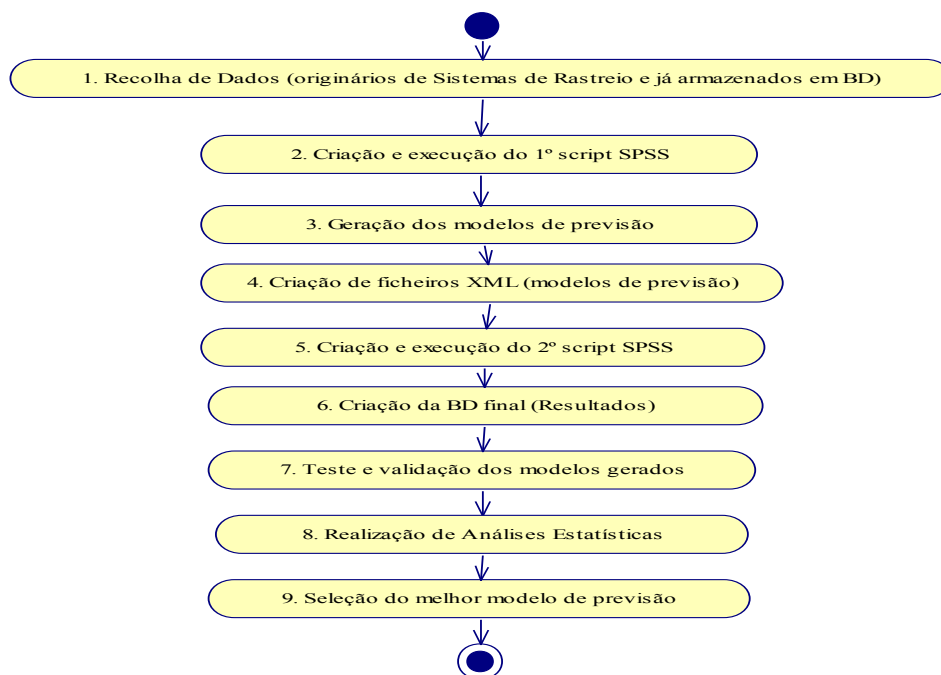


Figura 4.2 – Diagrama de atividades da solução proposta (UML)

4.2.5. Hipóteses e variáveis

Após a recolha de dados históricos sobre os pedidos de evolução, foram criadas tabelas de dados para cada um dos projetos caracterizados na subsecção 4.1.3. Os dados foram armazenados em ficheiros para posterior análise com o *software* de análise estatística SPSS. Essencialmente, os ficheiros de dados (.SAV) (Apêndice I, Figura 1) criados consistem numa tabela com 5 campos (variáveis): *MonthYear*, *ChangeRequests*, *Year_*, *Month_*, *Date_*. Cada linha da tabela corresponde aos dados de pedidos de evolução agregados mensalmente (ou seja, cada caso na tabela corresponde a um mês num determinado ano). Vejamos um exemplo na Tabela 4.1.

MonthYear	ChangeRequests	YEAR_	MONTH_	DATE_
Oct 2001	223	2001	10	OCT 2001
Nov 2001	140	2001	11	NOV 2001
Dec 2001	91	2001	12	DEC 2001

Tabela 4.1 – Tabela de dados parciais do *Software Eclipse JDT*

A variável *MonthYear* (variável independente) assume um carácter temporal e foi utilizada para agregar a informação indispensável à criação das séries temporais com uma granularidade mensal. A variável *ChangeRequests* (variável dependente) representa o número total de pedidos de evolução de *software*, que será utilizado para a criação dos modelos de previsão. As restantes variáveis são de carácter temporal e foram utilizadas no tratamento de dados necessários ao desenvolvimento de *scripts* SPSS.

Tendo como referência os objetivos apresentados na subsecção 4.2.1. foram formuladas cinco hipóteses (Quadro 4.2). Cada hipótese H_i corresponde à respetiva questão de investigação Q_i , proposta na subsecção 4.1.4.

H1	H1 ₀	A escolha do “melhor” modelo de previsão é estável.
	H1 ₁	A escolha do “melhor” modelo de previsão é volátil.
H2	H2 ₀	O modelo de previsão apresenta a menor taxa de erro.
	H2 ₁	O modelo de previsão não apresenta a menor taxa de erro.
H3	H3 ₀	Prever a evolução do número de pedidos de evolução com modelos <i>Exponential Smoothing</i> e ARIMA é uma abordagem válida.
	H3 ₁	Prever a evolução do número de pedidos de evolução com modelos <i>Exponential Smoothing</i> e ARIMA não é uma abordagem válida.
H4	H4 ₀	A série temporal do número de pedidos de evolução apresenta um padrão sazonal.
	H4 ₁	A série temporal do número de pedidos de evolução não apresenta um padrão sazonal.
H5	H5 ₀	A série temporal do número de pedidos de evolução apresenta uma tendência.
	H5 ₁	A série temporal do número de pedidos de evolução não apresenta uma tendência.

Quadro 4.2 – Hipóteses testadas

Na verdade, ambicionámos saber, com estas hipóteses, se cada modelo explicava, muito ou pouco, a variação apresentada e se errava mais ou menos que os restantes modelos; se as metodologias

Exponential Smoothing (subsecção 2.3.2) e *ARIMA* (subsecção 2.3.3) são aproximações válidas para prever o número de pedidos de evolução; e, por fim, pretendemos ainda avaliar se o número de pedidos de evolução ao longo do tempo apresentava algum padrão sazonal ou uma tendência.

4.2.6. Desenho

Neste trabalho experimental utilizámos o desenho de um estudo longitudinal, em que as observações são as contagens dos pedidos de evolução, e o tratamento é a reavaliação de modelos realizada mensalmente. Isto para a construção do modelo em si, porque para a seleção e observação é efetuada a reavaliação dos valores das métricas escolhidas.

Na Tabela 4.2 é apresentado o número total de observações, para cada um dos sistemas analisados, quer na série temporal dos pedidos, quer nas séries de métricas de “mérito” dos modelos. O nosso estudo vai incidir precisamente sobre as séries de métricas de “mérito” dos modelos (subsecção 4.2.8.).

Nº de observações		
Sistemas	Série temporal dos pedidos	Séries de métricas de “mérito” dos modelos
<i>Eclipse JDT</i>	72	48
<i>Eclipse Platform</i>	83	59
<i>Eclipse Android</i>	47	23
<i>Eclipse</i>	101	77
<i>Mozilla Core</i>	72	48
<i>Mozilla Firefox</i>	88	64
<i>Gnome Evolution</i>	99	75
<i>Gnome GStreamer</i>	77	53

Tabela 4.2 – Número total de observações

4.2.7. Procedimento

Como ponto de partida para esta experiência, procedemos à obtenção dos pedidos de evolução para cada *software*, que foram extraídos de um sistema de rastreio de pedidos de evolução, *Bugzilla*, no caso do *software Eclipse* [18], e disponibilizados num artigo para os restantes *softwares* [33].

As ferramentas utilizadas neste trabalho foram o *Eclipse SDK*, versão 4.2.1, o *Portable IBM SPSS Statistics*, versão 19 e o *Microsoft Excel 2010*. O *Eclipse SDK* (*Software Development Kit*) permitiu-nos o desenvolvimento de dois *scripts* SPSS, em linguagem de programação *Java*, para os executarmos na aplicação SPSS, a fim de podermos gerar os modelos de previsão e obtermos os

resultados das métricas estatísticas correspondentes. O *Portable IBM SPSS Statistics* foi a ferramenta utilizada para a geração de modelos de séries temporais e para a realização de análises estatísticas. O *Microsoft Excel* foi uma ferramenta que assegurou a interoperabilidade com a aplicação SPSS. Os ficheiros .CSV foram convertidos em .XLXS, para que os resultados finais pudessem ser importados pela aplicação SPSS.

Na Figura 4.3 é apresentado um diagrama de sequência em UML (*Unified Modeling Language*), que inclui as atividades e os procedimentos essenciais para a execução deste trabalho. Neste diagrama estão presentes as ferramentas descritas anteriormente e os resultados das operações realizadas. Os diversos componentes comunicam entre si através de mensagens (setas direcionais), que representam a identificação de eventos. O investigador é considerado o único ator e é responsável pela recolha dos dados e pela execução/configuração de todos os processos indispensáveis nesta experiência. Este diagrama inclui ainda a numeração de todas as etapas importantes, descritas a seguir. Todas as etapas são repetidas para cada sistema. Para cada sistema existe uma base de dados em que se representam os dados históricos da evolução do número de pedidos, agregados com uma granularidade mensal.

Na etapa 1 o investigador necessitou de consultar a base de dados na aplicação *Portable IBM SPSS Statistics* para ter conhecimento da constituição da mesma (campos da tabela) e dos respetivos dados (intervalo temporal referente ao número de pedidos de evolução) (Tabela 4.1). Nesta experiência cada sistema envolveu um determinado número de observações (Tabela 4.2) e de pedidos de evolução (Tabela 4.3).

Sistemas	Nº de pedidos de evolução
<i>Eclipse JDT</i>	282 335
<i>Eclipse Platform</i>	7 779
<i>Eclipse Android</i>	20 169
<i>Eclipse</i>	282 335
<i>Mozilla Core</i>	10 813
<i>Mozilla Firefox</i>	8899
<i>Gnome Evolution</i>	13 458
<i>Gnome GStreamer</i>	3604

Tabela 4.3 – Nº de pedidos de evolução dos sistemas

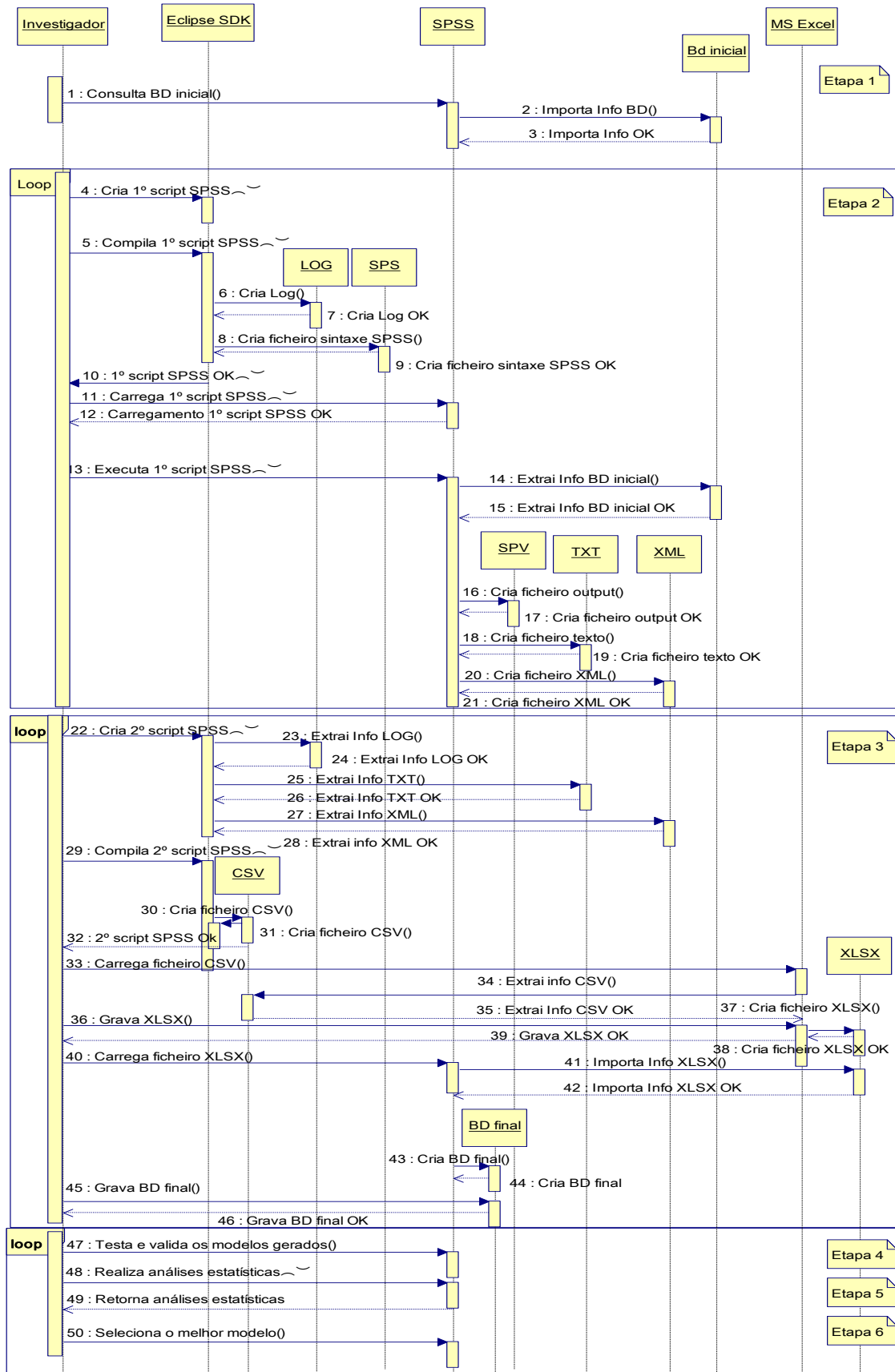


Figura 4.3 – Diagrama de seqüência de atividades em UML

Posteriormente, foi gerado automaticamente um ficheiro *script* SPSS que serviu para automatizar a geração dos modelos de suporte à análise e previsão de séries temporais. Portanto, este *script* usou os dados registados na tabela anterior para gerar os modelos de previsão. Deste último processo resultou um conjunto de modelos, cada um armazenado num ficheiro XML (*Extensible Markup Language*). Trata-se de um documento com dados organizados de forma hierárquica. Foram utilizados dois tipos de modelos de previsão, a saber: *Exponential Smoothing* (subsecção 2.3.2.) e *ARIMA*²⁶ (com e sem sazonalidade) (subsecção 2.3.3.). No modelo *ARIMA* foram utilizados valores entre 0 e 5 para cada elemento (p, d, q), pois num pré-teste todos os modelos seleccionados apareciam dentro destes valores. Estes modelos são usados normalmente em análise de séries temporais, sendo os disponibilizados pela ferramenta de análise de dados usada neste estudo (*Portable IBM SPSS Statistics*).

Assim sendo, na etapa 2, com base nas bases de dados iniciais (Tabela 4.1), foi desenvolvida uma aplicação em *Java* (*1º script SPSS*) que permitiu a geração de ficheiros de registo de operações (.LOG) e de ficheiros de sintaxe (.SPS) para cada *software* em estudo. Os ficheiros .LOG (Apêndice I, Figura 2) foram criados, não só, para nos facilitar no controlo das operações realizadas, como também para podermos extrair algumas informações (sistema, modelo, início e fim de treino, início e fim da previsão) que necessitámos para a segunda aplicação, igualmente desenvolvida em *Java*. Os ficheiros .SPS (Apêndice I, Figura 3) têm comandos numa linguagem específica do SPSS, que lhe podem ser dados para que sejam realizadas operações de estatística (por exemplo, um comando pode mandar gerar um modelo *ARIMA* sobre uma determinada variável). Para o efeito são produzidos dois ficheiros de resultados. Um formato de *output* para visualização na ferramenta SPSS (.SPV), outro em XML, para utilização do próprio SPSS, e suporte a interoperabilidade com outras ferramentas. Os ficheiros .SPV (Apêndice I, Figura 4) são ficheiros de *output* e neste caso, em particular, as informações apresentadas são tabelas e gráficos referentes a métricas estatísticas, de qualidade e de erro, correspondentes aos modelos de previsão gerados, segundo os critérios de seleção (secção 3.2.8.) associados ao tipo de modelo aplicado (*Expert Modeler*). Neste tipo de modelo foi imposta a seleção dos cinco melhores modelos de previsão (*ARIMA* ou *Exponential Smoothing*) para cada série temporal. A informação de cada ficheiro .SPV foi exportada para um respetivo ficheiro .TXT (Apêndice I, Figura 5), pois foi necessário aceder à identificação de cada modelo gerado para criar o *2º script SPSS*. Os ficheiros .XML (Apêndice II, Figura 6) apresentam sobretudo os valores correspondentes às métricas de qualidade, de erro e de normalização referentes a cada um dos modelos gerados.

²⁶http://publib.boulder.ibm.com/infocenter/spsstat/v20r0m0/index.jsp?topic=%2Fcom.ibm.spss.statistics.help%2Fsyn_ts_apply_fit_measures.htm

Na etapa 3 procedeu-se ao desenvolvimento da segunda aplicação desenvolvida em *Java* (2^o script *SPSS*), que teve como base de trabalho os dados provenientes dos ficheiros .LOG, .TXT e .XML de cada *software* em estudo. Estes dados foram devidamente agrupados e armazenados num ficheiro .CSV (Apêndice II, Figura 7). Este ficheiro depois de convertido para .XLSX (Apêndice II, Figura 8) foi carregado no programa *SPSS* e gravado como base de dados final (.SAV). Esta base de dados inclui a identificação do sistema, os períodos de estimação e de previsão e as métricas de erro, de ajuste e de normalização disponíveis na aplicação *SPSS* (Apêndice II, Figura 9).

A etapa 4 consistiu na operação de teste e validação dos modelos gerados com base nos dados históricos existentes em cada sistema.

Na etapa 5, foram analisados estatisticamente os resultados produzidos, para posterior seleção do melhor modelo em cada sistema (etapa 6).

Na tabela 4.5 apresentamos a correspondência entre as atividades e a sequência necessária para a realização das mesmas nesta experiência (ambas identificadas pelo respetivo número existente nas Figuras 4.2 e 4.3).

Diagramas	
Atividade	Sequência
1	1 à 3
2	4 à 13
3	14 à 19
4	20 à 21
5	22 à 39
6	40 à 46
7	47
8	48 à 49
9	50

Tabela 4.5 – Correspondência entre os diagramas de atividade e de sequência

4.2.8. Procedimento de análise

Para cada sistema sob análise, simulámos a reconstrução mensal dos modelos de estimação, ao longo do projeto, usando toda a informação que estava disponível até à data para prever os 12 meses seguintes. Em particular, testámos a hipótese de que, a partir de um determinado volume de dados, os modelos de previsão que tiram partido de informação sazonal são melhores do que os que não usam. Em termos objetivos, foram efetuados vários testes, com períodos de treino que abrangeram sempre doze ou mais meses e tiveram todos a mesma referência temporal inicial, que correspondeu ao primeiro mês de dados disponível. A opção por incluir sempre os dados mais antigos, por oposição a ir “esquecendo” esses dados visou tirar o melhor partido possível da componente de sazonalidade, na qual um maior volume de dados permite capturar com maior

precisão os eventuais efeitos dessa sazonalidade [18]. Os períodos de previsão foram todos de doze meses. Isso permitiu-nos identificar, mensalmente, quais os “melhores” modelos disponíveis à data, considerando três critérios: a) a capacidade de explicação do modelo (a maximizar, medida pela métrica *RSQUARE*), b) o erro do modelo face aos dados reais (a minimizar, medida pela métrica *MAPE*) e c) o ajuste do modelo face aos dados, privilegiando o princípio da parcimónia (em termos simples, procura-se atingir o melhor equilíbrio possível entre o ajuste do modelo, que se quer maximizar, e a complexidade do modelo, que se quer minimizar, medida pela métrica *normalized BIC*²⁷). Para cada modelo que, de acordo com algum dos critérios usados, em algum dos momentos se revelou como o “melhor”, a avaliação foi feita ao longo de toda a série temporal. Deste modo, no final da simulação, temos a evolução completa de todos os modelos que, em algum momento, foram considerados os “melhores” à luz de pelo menos um dos critérios usados. Isto permite-nos analisar a evolução dos modelos em vários sistemas, em busca de padrões recorrentes. Para se compreender melhor todo o processo, vejamos o exemplo apresentado no Quadro 4.3. O primeiro teste realizado conta com um período de 12 meses de treino e 12 de previsão. O segundo teste conta com um período de 13 meses de treino e 12 de previsão. A cada novo teste, vai sendo acrescentado mais um mês de treino, mantendo-se os 12 de previsão. E assim sucessivamente.

²⁷ *Bayesian Information Criterion.*

1º Teste	Período (meses)	Início	Fim	Ago-02			Jul-03	Ago-03			Jul-04							
Treino	12	Ago-02	Jul-03															
Previsão	12	Ago-03	Jul-04															
2º Teste									Set-03			Ago-04						
Treino	13	Ago-02	Ago-03															
Previsão	12	Set-03	Ago-04															
...																		
Último Teste	Período	Início	Fim												...	Jul-08	Ago-08	Jul-09
Treino	6 anos	Ago-02	Set-08															
Previsão	12 meses	Out-08	Set-09															

Quadro 4.3 - Exemplo de planeamento de testes

A análise estatística de séries temporais foi realizada com base em tabelas de dados e gráficos denominados de correlogramas, com o objetivo de identificar na série temporal prováveis padrões sazonais ou tendências. Estes gráficos apresentam a correlação entre o número de defeitos em intervalos de tempo consecutivos. Os correlogramas foram também utilizados para comparar, em termos das métricas de ajuste, de erro e de normalização, todos os modelos de previsão gerados, segundo cada critério de seleção.

Os critérios de seleção acima descritos foram representados respetivamente pelas métricas: *RSQUARE* (Quadro 4.5), *MAPE* (Quadro 4.6) e *normalized BIC* (Quadro 4.7). Estas 3 métricas foram escolhidas porque permitem fazer uma melhor distinção entre os diferentes modelos de previsão e identificar o modelo mais fiável. Podemos considerar um bom modelo se o erro for ruído branco (e relativamente pequeno). O ruído branco está associado a uma sequência de erros (ou choques) aleatórios, sempre que esta tiver média e variância constante e sem autocorrelação (medida que informa o quanto um valor de uma variável aleatória é capaz de influenciar os seus vizinhos. Oscila entre os valores -1 e um 1. O valor 0 significa total ausência de correlação).

É considerado temporalmente homogéneo, estacionário e sem dependência temporal.

Em suma, os modelos gerados foram analisados de forma sistemática, tendo em conta a sua capacidade de explicação da variação das séries temporais de pedidos de evolução (através da métrica *RSQUARE*), o erro nas suas previsões (através da métrica *MAPE*) e o nível de ajuste dos modelos, com o *Normalized BIC*. Naturalmente, pretendeu-se identificar, por um lado, os modelos que explicavam uma maior percentagem da variação do número de pedidos de evolução e que minimizavam o erro feito nas previsões e, por outro, os que melhor balanceavam o ajuste do modelo com a sua parcimónia.

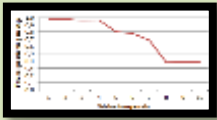
Esta nossa análise sistemática incidiu sobre a evolução do “mérito relativo” dos modelos de previsão, quando fizemos variar o período de treino. Na realidade, quisemos saber se cada modelo explicava, muito ou pouco, a variação apresentada e se errava mais ou menos que os outros modelos. Para o efeito, tivemos diferentes métricas alternativas e estudámos quais as mais eficazes para a nossa análise. Abaixo encontram-se descritas todas as métricas de ajuste, de erro e de normalização disponíveis na aplicação SPSS, que foram usadas neste trabalho, para que se pudesse fazer uma análise mais objetiva e fiável. No Quadro 4.4 apresentamos a explicação das incógnitas usadas nas fórmulas dessas métricas.

Representação	Descrição
N	Número de previsões efetuadas
K	Número de variáveis independentes (explicativas)
t	Período de tempo
Y(t), X _i	Valor da série temporal no período t
$\hat{Y}(t)$	Valor ajustado da série temporal no período t
\bar{Y}	Média de valores da série temporal
μ	Valor esperado (média) da variável aleatória x
SSQ	(Residual sum of squares) Soma dos quadrados dos resíduos (erros de previsão)
SSQ'	(Adjusted Residual sum of squares) Soma ajustada dos quadrados dos resíduos
$\hat{\sigma}_a^2$	(Estimated residual variance) Variância residual estimada
N_p	Número de parâmetros no modelo estatístico
L	Valor maximizado da função de probabilidade

Quadro 4.4 – Descrição de incógnitas usadas nas fórmulas das métricas

4.2.8.1. Métricas de ajuste

As métricas de ajuste (Quadro 4.5) possibilitam-nos saber se o modelo em causa está ajustado à série temporal. Os valores das métricas de ajuste devem ser maximizados.

Designação da métrica	RSQUARE (R-Square) ²⁸
Definição informal	O coeficiente de explicação R^2 é uma estimativa da proporção da variação total na série que se explica pelo modelo.
Definição formal	$R^2 = \left(1 - \frac{\sum_{t=1}^N (Y(t) - \hat{Y}(t))^2}{\sum_{t=1}^N (Y(t) - \bar{Y})^2} \right) \times 100$
Comentário	É uma métrica calculada através da comparação do erro do modelo e a variação dos dados da série dependente à volta da sua média. Está associada à capacidade de explicação do modelo de previsão. Esta métrica é particularmente adequada a séries estacionárias.
Perfil	 <p>Podem assumir qualquer valor entre 0 e 100%. Quanto maior for o seu valor, melhor será a explicação do modelo.</p>

²⁸ http://www2.dbd.puc-rio.br/pergamum/tesesabertas/0712965_10_cap_04.pdf

Designação da métrica	SRSQUARE (Stationary R-Square)²⁹
Definição informal	A raiz quadrada estacionária é uma medida que compara a parte estacionária do modelo com um modelo de média simples.
Definição formal	$R^2_s = 1 - \left(\frac{N-1}{N-k} \right) (1 - R^2)$
Comentário	É uma métrica que está ligada à capacidade de explicação do modelo de previsão. Esta métrica é preferível à raiz quadrada comum, quando há uma tendência ou padrão sazonal. Trata-se de uma medida de correção ao R^2 , pois penaliza a inclusão de variáveis independentes pouco explicativas, beneficiando assim o princípio da parcimónia.
Perfil	Análogo à métrica <i>R-Square</i> .

Quadro 4.5 – Métricas de Ajuste

4.2.8.2. Métricas de erro

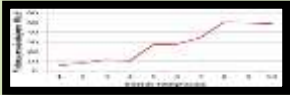
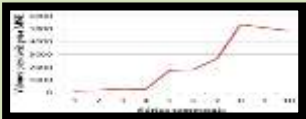
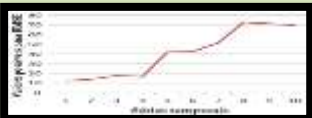
As métricas de erro (Quadro 4.6) mostram-nos as percentagens de erro do modelo. Os valores destas métricas devem ser minimizados. Os erros de previsão são resultantes da diferença entre o valor real e a previsão do mesmo. É relevante proceder à sua monitorização para determinar se estão dentro de limites razoáveis³⁰ (intervalo de confiança), aquando da realização de previsões periódicas.

Designação da métrica	ErrVariance (Error Variance)³¹
Definição informal	O erro de variância é uma medida de dispersão estatística, isto é, indica geralmente quão longe os seus valores se encontram do valor esperado.
Definição formal	$\sigma^2 = \frac{\sum_{i=1}^n (x_i - \mu)^2}{n}$
Comentário	Representa a média do quadrado da distância de cada ponto até à média, ou seja, é a média do quadrado dos desvios. Ou seja, é o valor esperado do quadrado do desvio dos valores da variável relativamente à sua própria média. A definição apresentada pode ser usada para variáveis aleatórias discretas e contínuas.
Perfil	Os valores devem oscilar num intervalo de confiança de 95%.

²⁹ <http://www.bertolo.pro.br/AdminFin/AnalInvest/NotasSobreRegressao.pdf>

³⁰ Intervalo estimado de um parâmetro estatístico, utilizado para assinalar a sua credibilidade.

³¹ <http://www.dpi.ufv.br/~peternelli/inf162.www.16032004/materiais/CAPITULO4.pdf>
<http://pessoal.utfpr.edu.br/eustaquio/arquivos/Apostila%20de%20Estatistica.pdf>

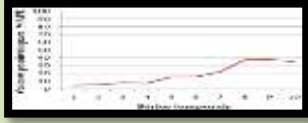
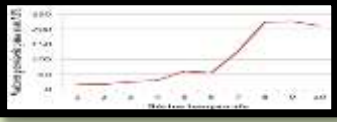
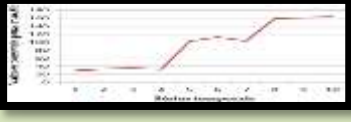
Designação da métrica	MAE (Mean Absolute Error)³²
Definição informal	O erro médio absoluto ³³ representa a média dos erros cometidos pelo modelo de previsão ao longo de uma série de períodos de tempo.
Definição formal	$MAE = \frac{\sum_{t=1}^N Y(t) - \hat{Y}(t) }{N}$
Comentário	É uma das métricas mais frequentes de erro de previsão, que não tem em consideração se um erro foi sobrestimado ou subestimado.
Perfil 	Os valores devem oscilar num intervalo de confiança, com um nível de 95%. Esta métrica encontra-se nas mesmas unidades que a sua série.
Designação da métrica	MSE (Mean Square Error)³⁴
Definição informal	O erro quadrático médio resulta da média de todos os erros (elevados ao quadrado) associados ao modelo de previsão nas séries temporais em causa.
Definição formal	$MSE = \frac{\sum_{t=1}^N (Y(t) - \hat{Y}(t))^2}{N}$
Comentário	Esta métrica tem como vantagem estar na mesma escala que os dados.
Perfil 	Os valores devem oscilar num intervalo de confiança de 95%.
Designação da métrica	RMSE (Root Mean Square Error)³⁵
Definição informal	A raiz quadrada do erro quadrático médio (ou média quadrática) representa o desvio quadrático médio.
Definição formal	$RMSE = \sqrt{MSE}$
Comentário	É uma boa medida de precisão de previsões, capaz de indicar a magnitude do erro.
Perfil 	Os valores devem oscilar num intervalo de confiança de 95%.

³² <http://www.eps.ufsc.br/disserta96/mueller/cap3/cap3.htm>

³³ Ou módulo de um número representa o próprio número, que se obtém ao eliminar o sinal que eventualmente o anteceda. Ex: $|-1| = |1| = 1$.

³⁴ <http://www.mudancasabruptas.com.br/MetQuant5.pdf>

³⁵ http://pic.dhe.ibm.com/infocenter/spssstat/v20r0m0/index.jsp?topic=%2Fcom.ibm.spss.statistics.help%2Fsyn_tsmode_l_autooutlier.htm

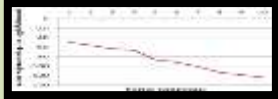
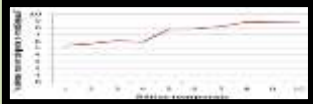
Designação da métrica	MAPE (Mean Absolute Percentage Error)³²
Definição informal	Trata-se da média da diferença absoluta entre os valores previstos e os valores atuais, expressa em percentagem.
Definição formal	$MAPE = \frac{\sum_{t=1}^N \left \frac{Y(t) - \hat{Y}(t)}{Y(t)} \right }{N} \times 100$
Comentário	O erro médio percentual absoluto é um bom indicador de precisão. Quanto menor for o seu valor, melhor será a adequação do modelo analisado e, consequentemente, melhor será a previsão.
Perfil	 <p>É uma métrica fácil de se entender, pois os seus resultados são expressos em %. No entanto, se o valor real for pequeno, qualquer discrepância na previsão fará disparar o valor desta métrica.</p>
Designação da métrica	MAXAPE (Maximum Absolute Percentage Error)³⁴
Definição informal	É o maior erro previsto, expresso em percentagem.
Definição formal	$MaxAPE = 100 \max((Y(t) - \hat{Y}(t)) / Y(t))$
Comentário	Trata-se de uma medida estatística útil para conceber o pior cenário de previsões.
Perfil	 <p>Os valores devem oscilar num intervalo de confiança de 95%.</p>
Designação da métrica	MAXAE (Maximum Absolute Error)³⁴
Definição informal	É o maior erro de previsão.
Definição formal	$MaxAE = \max(Y(t) - \hat{Y}(t))$
Comentário	Tal como o MAXAPE, é profícuo para idealizar o pior cenário para as previsões.
Perfil	 <p>Os valores devem oscilar num intervalo de confiança de 95%.</p>

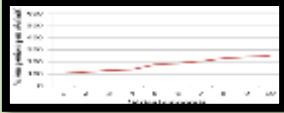
Quadro 4.6 – Métricas de Erro

Depois de pesadas todas as características das métricas de erro a que tivemos acesso na aplicação SPSS, optámos por escolher usar a métrica MAPE, de modo a tirar partido das suas características de precisão e de simplicidade, que eram mais importantes para o nosso contexto.

4.2.8.3. Métricas de normalização

No processo de análise de dados é muito importante a seleção do modelo adequado, do ponto de vista estatístico. Procura-se, de facto, o modelo mais parcimonioso, isto é, o modelo que inclua o mínimo de parâmetros possíveis a serem estimados e que explique bem o comportamento da variável dependente. Neste estudo utilizámos os seguintes critérios de informação para seleção de modelos: *Log - Likelihood*, BIC (*Bayes Information Criterion*) e o AIC (*Akaike Information Criterion*), apresentados no Quadro 4.7. Num cenário ideal estes critérios deverão ser os menores possíveis podendo assumir valores negativos, sendo que ambos medem o quanto o modelo estimado se ajusta aos dados.

Designação da métrica	logLikelihood (<i>Log - Likelihood</i>)³⁴
Definição informal	A máxima verosimilhança é o método de estimação dos parâmetros de um modelo estatístico.
Definição formal	$L = -N \ln \left(\hat{\sigma}_a \right) - \frac{SSQ'}{2 \hat{\sigma}_a^2} - \frac{N \ln(2\pi)}{2}$
Comentário	Estima os diferentes parâmetros de um modelo estatístico, de forma a maximizar a probabilidade dos valores observados.
Perfil	Os valores devem oscilar num intervalo de confiança de 95%. 
Designação da métrica	NORMBIC (<i>Normalized Bayes Information Criterion</i>)³⁴
Definição informal	É um critério de informação normalizada Bayesiano (BIC).
Definição formal	$\text{Normalized BIC} = \ln(MSE) + k \frac{\ln(n)}{n}$
Comentário	É uma pontuação baseada no erro quadrático médio e inclui uma penalização para o número de parâmetros do modelo e do comprimento da série. A penalização remove a vantagem de modelos com mais parâmetros, facilitando a comparação entre diferentes modelos para a mesma série.
Perfil	Os valores devem oscilar num intervalo de confiança de 95%. 

Designação da métrica	AIC (<i>Akaike Information Criterion</i>) ³⁴
Definição informal	O critério de informação de <i>Akaike</i> é uma medida para a seleção e comparação de modelos.
Definição formal	$AIC = -2L + 2N_p$
Comentário	Os valores menores indicam melhores modelos. A AICC (<i>Corrected Akaike Information Criterion</i>) corrige o critério de informação de <i>Akaike</i> para pequenos tamanhos de amostra.
Perfil	Os valores devem oscilar num intervalo de confiança de 95%. 

Quadro 4.7 – Métricas de Normalização

4.3. Execução

Na execução deste trabalho experimental foram cumpridos todos os objetivos, utilizadas as unidades experimentais mencionadas na seção 3.2.2., com recurso aos materiais necessários, e realizadas todas as tarefas estipuladas para o efeito.

Em suma, o número de pedidos de evolução variou consoante o sistema em causa (Tabela 4.3, subsecção 4.2.7.). Na criação dos modelos de séries temporais foi levado em conta a agregação de dados ordenados por data (mês e ano) e com o número de pedidos de evolução em cada mês. As análises estatísticas foram realizadas sobre as bases de dados completas (inicial e final), de acordo com as amostras disponibilizadas (secção 3.2.3.).



Resultados

Tendo como referência as técnicas estatísticas descritas no capítulo anterior, apresentaremos de seguida a descrição dos resultados obtidos, à luz dos objetivos definidos para este trabalho de investigação (subsecção 4.2.1.). Para a concretização dos mesmos foi elaborado um conjunto de questões de investigação (subsecção 4.1.4.). Para obtenção das respetivas respostas foram realizados cálculos para averiguar as medidas de ajuste (subsecção 4.2.8.1), de erro (subsecção 4.2.8.2) e de normalização (subsecção 4.2.8.3) associadas aos modelos de previsão.

O nosso estudo envolve oito sistemas, identificados na subsecção 3.2.3., seguindo sempre o mesmo protocolo. Para ilustrar o protocolo, apresentamos os resultados de um dos sistemas, nomeadamente o *Eclipse JDT*, sumariando no final os resultados dos restantes sistemas cobertos nesta investigação.

5.1. Apresentação de resultados

No sistema *Eclipse JDT* tivemos como base de trabalho os seus dados históricos compreendidos entre outubro de 2001 a setembro de 2007. A geração do 1º *script* SPSS foi feita de acordo com as três perspetivas (critérios de seleção *RSQUARE*, *MAPE* e *NORMBIC*) explicadas na subsecção 3.2.8. Após a execução deste *script* SPSS foram gerados quatro modelos de previsão, a saber:

- *Exponential Smoothing - Simple Seasonal* (EXSS);
- *Exponential Smoothing – Winters’ Additive* (EXWA);
- *ARIMA (1,0,0) (0,0,0)* (ANS100);
- *ARIMA (0,0,0) (0,0,0)* (ANS000).

A razão pela qual estes modelos foram gerados deve-se ao facto de, em determinadas séries temporais do sistema em causa, apresentarem os melhores resultados, segundo o critério selecionado. No caso do modelo EXSS, que foi selecionado pelos três critérios, podemos concluir que se trata de um modelo com uma boa capacidade de explicação acerca da evolução do número de pedidos, com uma menor percentagem de erros em relação aos restantes modelos e que, conseqüentemente, privilegia o princípio da parcimónia. O modelo ANS100 foi gerado segundo o critério *Normalized BIC (Normalized Bayesian Information Criterion)*, o que significa que em algumas séries temporais atingiu o melhor equilíbrio possível entre o ajuste e a complexidade do modelo. O modelo ANS000 foi gerado conforme os critérios *RSQUARE* e *MAPE*, apenas na primeira série temporal, o que denota que à falta de melhor informação foi o modelo considerado menos mau na realidade. É interessante salientar que os critérios *RSQUARE* e *MAPE* apresentaram precisamente os mesmos resultados, não só neste sistema como também nos restantes.

De seguida, apresentaremos um conjunto de instrumentos (gráficos e tabelas) que nos permitiram simplificar o processo de análise e avaliação dos resultados, visto que não é muito fácil identificar por mera observação o melhor modelo de previsão.

As Figuras de 5.1 a 5.4 estão associadas a um período de observação compreendido entre outubro de 2001 a outubro de 2006. O período de previsão está associado aos 12 meses seguintes. Na leitura dos gráficos deve-se ter em conta que o eixo das abcissas representa o período temporal, referente ao período de observação e ao período de previsão, e o eixo das ordenadas diz respeito ao número de pedidos de evolução de *software*. O modelo de treino encontra-se representado através de uma linha vermelha, enquanto que o modelo de ajuste está associado a uma linha azul fina e os valores previstos estão representados por uma linha azul espessa. A linha a tracejado superior representa o limite superior de confiança (**UCL** – *Upper Confidence Limit*) e a linha a tracejado inferior representa o limite inferior de confiança (**LCL** – *Lower Confidence Limit*).

A tabela 5.1 exhibe os valores das métricas de ajuste (*Stationary R-squared* e *R-squared*), de erro (RMSE, MAPE, MAE, MaxAPE) e de normalização (*Normalized Bayesian Information Criterion*). Incluem ainda o teste *Ljung-Box Q* que através do valor Sig. pode indicar se o modelo de previsão é ajustado e adequado à série temporal. As figuras e as tabelas expostas representam apenas um *snapshot* dos modelos, das dezenas que foram construídos. Para uniformizar o processo de análise foi escolhida, para cada modelo, a representação da última série temporal do sistema em estudo.

5.1.1. Eclipse JDT – Modelo de Previsão EXSS

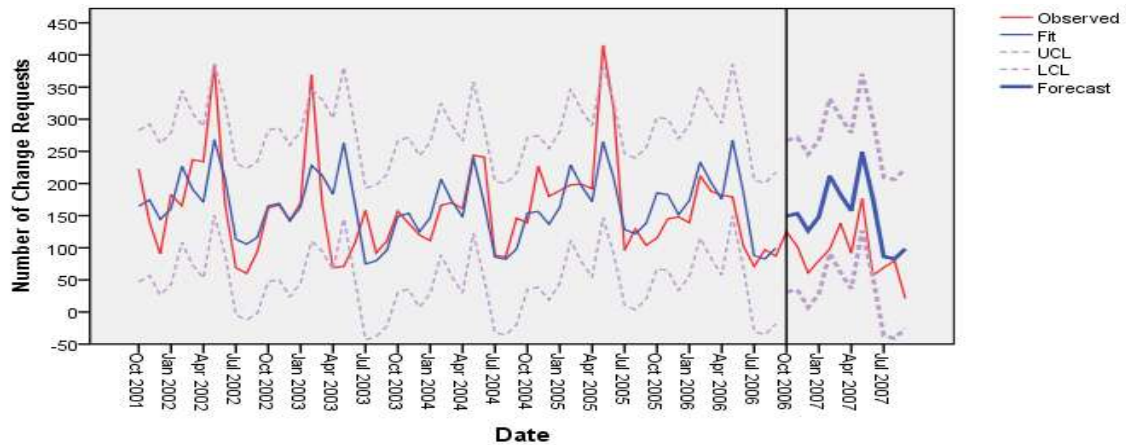


Figura 5.1 – Modelo de Previsão EXSS

Na Figura 5.1 encontra-se representado o modelo de previsão *Exponential Smoothing Simple Seasonal*, o que significa que esta série temporal não tem tendência, mas tem sazonalidade [26]. Podemos verificar, no geral, que os valores observados acerca do número de pedidos de evolução de *software* se encontram dentro dos limites de um intervalo de confiança de 95%. Os valores de previsão, apesar de serem sobrestimados, também se encontram dentro dos limites de confiança. Em sumário, quando comparados os dados observados com o modelo de ajuste percebe-se que o modelo se encontra ajustado o suficiente para se realizar previsões acerca de períodos futuros. Observando os valores previstos e comparando-os com os valores dos dados observados, podemos inferir que o modelo faz previsões dentro de um nível de confiança aceitável, apesar de se confirmar uma ligeira sobrestimação do número de pedidos de evolução de *software* previstos, no período de previsão.

5.1.2. Eclipse JDT – Modelo de Previsão EXWA

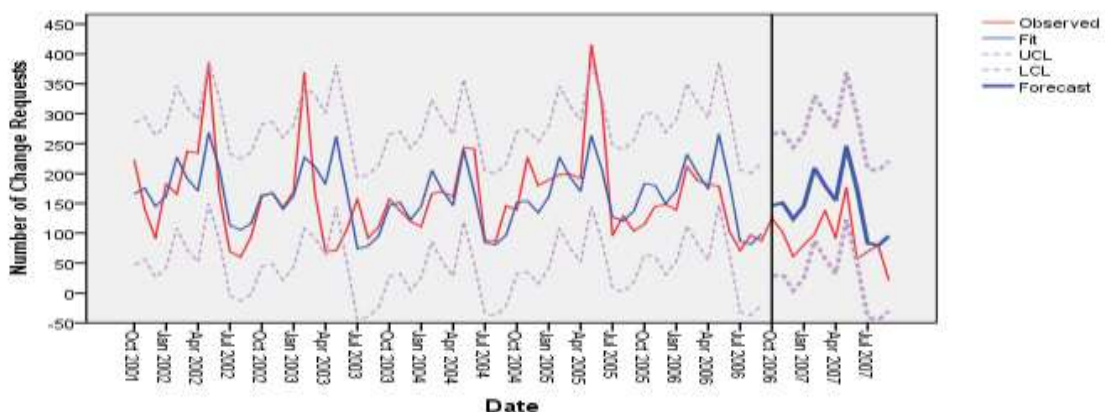


Figura 5.2 – Modelo de Previsão EXWA

Na Figura 5.2 encontra-se representado o modelo de previsão *Exponential Smoothing Winter's Additive*, o que implica que esta série temporal tem tendência e sazonalidade [26]. Podemos averiguar, no geral, que os valores observados se encontram dentro dos limites de um intervalo de confiança de 95%. Os valores de previsão apesar de serem sobrestimados também se encontram dentro dos limites credíveis. Em resumo, quando comparados os dados observados com o modelo de ajuste percebe-se que o modelo se encontra ajustado razoavelmente para se fazer previsões sobre períodos futuros. Examinando os valores previstos e comparando-os com os valores dos dados observados, podemos aferir que o modelo faz previsões dentro de um nível de confiança aceitável, apesar de se notar uma ligeira sobrestimação na previsão do número de pedidos de evolução de *software*, no período de previsão.

5.1.3. Eclipse JDT – Modelo de Previsão ANS100

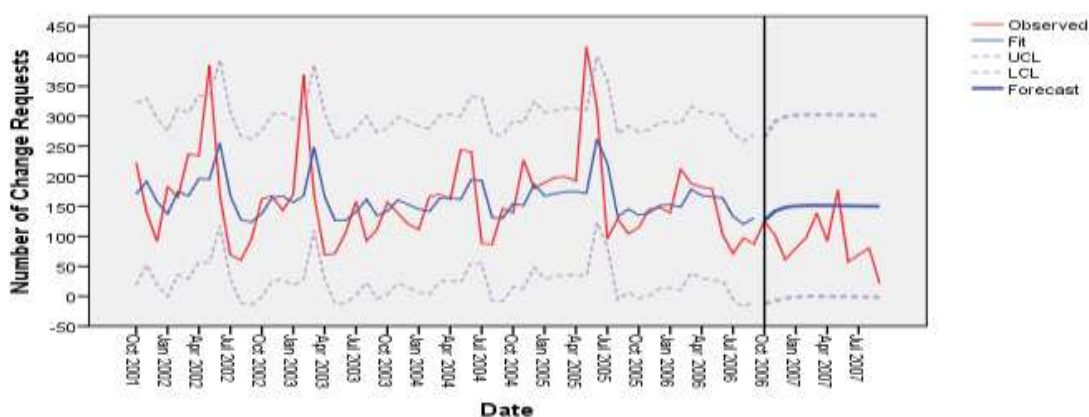


Figura 5.3 - Modelo de Previsão ANS100

Na Figura 5.3 encontra-se representado o modelo de previsão ARIMA (1,0,0)(0,0,0), cujos parâmetros indicam que se trata de um modelo sem características sazonais e com apenas um parâmetro auto regressivo, sem diferenciação de série e com média móvel nula. Isto significa que uma determinada observação num período corrente é gerada por uma média ponderada de observações passadas, sendo que 1 período anterior será incluído na série para além do erro do período corrente. Podemos apurar, no geral, que os valores observados se encontram dentro dos limites do intervalo de confiança de 95%. Quando comparados os dados observados com o modelo de ajuste percebe-se que o modelo se encontra ajustado o suficiente para efetuar previsões sobre períodos futuros. Confirma-se que os valores de ajuste atingem os seus máximos e mínimos 1 série mais tarde do que os valores reais. Os valores de previsão assumem um comportamento linear estacionário. Observando os valores previstos e comparando-os com os valores dos dados observados, podemos concluir que o modelo faz previsões dentro de um nível de confiança

aceitável, apesar de se verificar uma ligeira sobrestimação na previsão, na maioria dos casos, do número de pedidos de evolução de *software*, no período de previsão. No entanto, em Maio de 2007 o valor é subestimado. Ao contrário do que acontecia com os modelos anteriores, neste caso o modelo não tenta “antecipar” as flutuações mensais, apontando simplesmente para a média móvel (que neste caso parece manter-se praticamente ao longo de todo o período).

5.1.4. Eclipse JDT – Modelo de Previsão ANS000

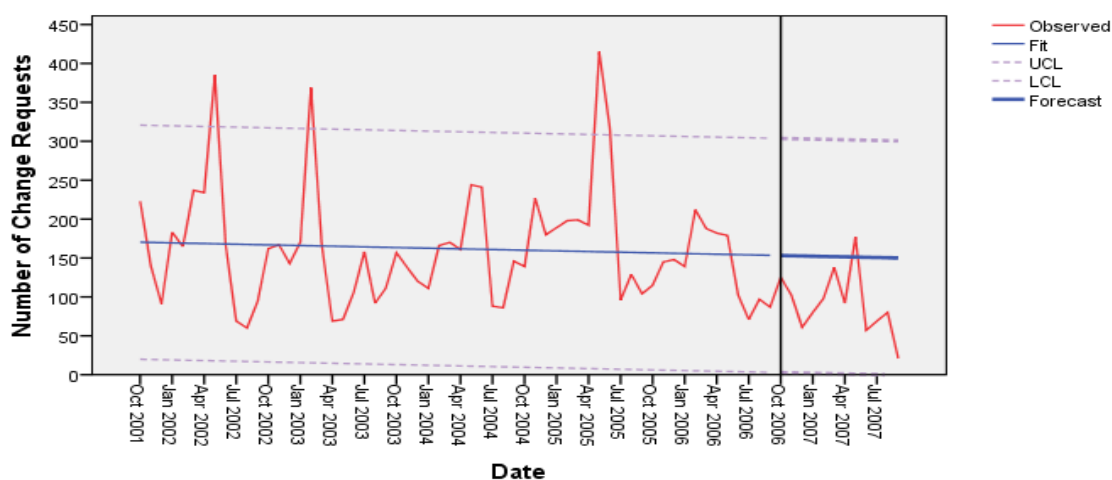


Figura 5.4 - Modelo de Previsão ANS000

Na Figura 5.4 está representado o modelo de previsão ARIMA (0,0,0)(0,0,0), cujos parâmetros mostram que se trata de um modelo sem características sazonais e sem parâmetros auto regressivos, sem diferenciação de série e com média móvel nula. Este modelo ARIMA (0,0,0)(0,0,0) é considerado um modelo de ruído branco³⁶, ou seja, é composto por uma sequência de valores aleatórios independentes. Podemos constatar que alguns valores observados se encontram fora dos limites do intervalo de confiança de 95%, nomeadamente nos meses de maio de 2002, fevereiro de 2003 e maio de 2005. Os valores de previsão assumem um comportamento linear estacionário, com um suave declive, tal como os valores do modelo de ajuste. Quando comparados os dados observados com o modelo de ajuste percebe-se que assumem comportamentos distintos, embora ambos respeitem os limites de confiança, na generalidade dos casos. Observando os valores previstos e comparando-os com os valores dos dados observados, podemos concluir que o modelo faz previsões dentro de um nível de confiança aceitável, apesar de se verificar uma sobrestimação na previsão do número de pedidos de evolução de *software*, no período de previsão. Salienta-se o

³⁶http://pt.wikibooks.org/wiki/Log%C3%ADstica/T%C3%A9cnicas_de_previs%C3%A3o/M%C3%A9todos_ARIMA_de_Box-Jenkins

facto que este gráfico é apenas um *snapshot*, tirado num momento em que este modelo na realidade não é adequado, mas que em algum momento o modelo foi o melhor. Tal como o modelo ANS100, neste caso o modelo não tenta “antecipar” as flutuações mensais, apontando simplesmente para a média móvel (baixa um pouco).

5.1.5. Eclipse JDT – Métricas estatísticas dos modelos gerados

Podemos constatar que a simples observação das Figuras de 5.1 a 5.4 não torna suficientemente simples a escolha dos melhores modelos de previsão. Como tal, para validar a precisão do modelo é essencial avaliar os resultados das métricas estatísticas. A Tabela 5.1 exhibe os valores obtidos na última série temporal do sistema em causa (como exemplo) referentes aos quatro modelos gerados, para que possamos discutir os resultados todos em conjunto.

Modelo	Model Fit statistics								Ljung-Box Q(18)		
	Stationary R-squared	R-squared	RMSE	MAPE	MAE	MaxAPE	MaxAE	Normalized BIC	Statistics	DF	Sig.
	EXSS	,698	,388	58,920	30,018	41,405	270,571	192,106	8,289	38,275	16
EXWA	,698	,388	59,411	29,857	41,305	269,339	191,231	8,374	38,343	15	,001
ANS100	,169	,169	69,237	34,581	47,662	142,944	243,433	8,680	15,240	17	,578
ANS000	,004	,004	75,147	39,367	52,921	179,068	256,956	8,775	28,002	18	,062

Tabela 5.1- Métricas estatísticas dos modelos gerados

Podemos verificar que os dois primeiros modelos (EXSS e EXWA) conseguem atingir valores muito próximos e francamente melhores que os dois últimos modelos (ANS100 e ANS000). Relativamente às medidas de ajuste, de erro e de normalização, incluindo o teste *Ljung-Box*, os modelos EXSS e EXWA exibem valores muito similares, o que não simplifica o processo de seleção do “melhor” modelo. Quanto aos modelos ARIMA apresentam valores das métricas de erro mais elevados e nas métricas de ajuste exibem valores consideravelmente baixos, quando comparado com os outros modelos. Quanto à medida de normalização, *Normalized BIC* (*Bayesian*

Information Criterion), estes modelos ostentam um valor superior aos restantes modelos. O modelo ANS100 foi selecionado segundo o critério *Normalized BIC*, daí que, apesar de apresentar valores das métricas pouco credíveis, o que é certo, é que na métrica *Normalized BIC* o seu valor é aproximado aos restantes. O modelo ANS000 apresenta valores das métricas de erro, de ajuste e de normalização francamente piores que os outros modelos e consideravelmente baixos.

5.1.6. Eclipse JDT – Comparação do número de modelos gerados segundo os critérios *RSQUARE*, *MAPE* e *Normalized BIC*

Nesta subsecção pretendemos, essencialmente, mostrar quantas vezes cada um dos modelos foi considerado o “melhor”, à luz de cada uma das 3 métricas de seleção. Nas figuras 5.5 e 5.6, expomos, por critério de seleção *RSQUARE*, *MAPE* e *Normalized BIC* (*Bayesian Information Criterion*), os modelos de previsão gerados para o *software Eclipse JDT*, segundo o modelo *Expert Modeler*. Este projeto compreendia 48 séries temporais, correspondentes ao período de outubro de 2001 a setembro de 2007.

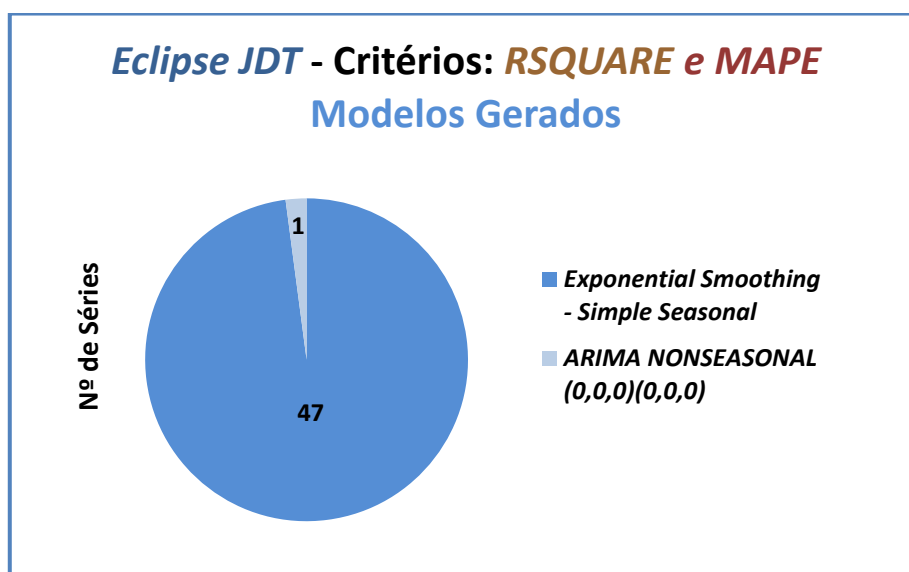


Figura 3.5 - Modelos gerados segundo o critério *RSQUARE* e *MAPE* (*Eclipse JDT*)

Na Figura 5.5 podemos verificar que o modelo *Exponential Smoothing - Simple Seasonal* (EXSS) se destaca novamente em relação ao outro modelo gerado, a saber: *ARIMA (0,0,0) (0,0,0)* (ANS000).

O modelo EXSS foi selecionado em 47 séries temporais, da 2ª à 48ª série. Por seu turno, o modelo ANS000 surgiu apenas na 1ª série temporal. Apesar de ser um modelo com valores estatísticos muito fracos, à falta de melhor informação foi escolhido o modelo considerado menos mau.

Enquanto existe pouca informação disponível, a capacidade dos modelos será em geral baixa. Neste caso, temos a opção menos má num modelo de média móvel o mais simples possível.

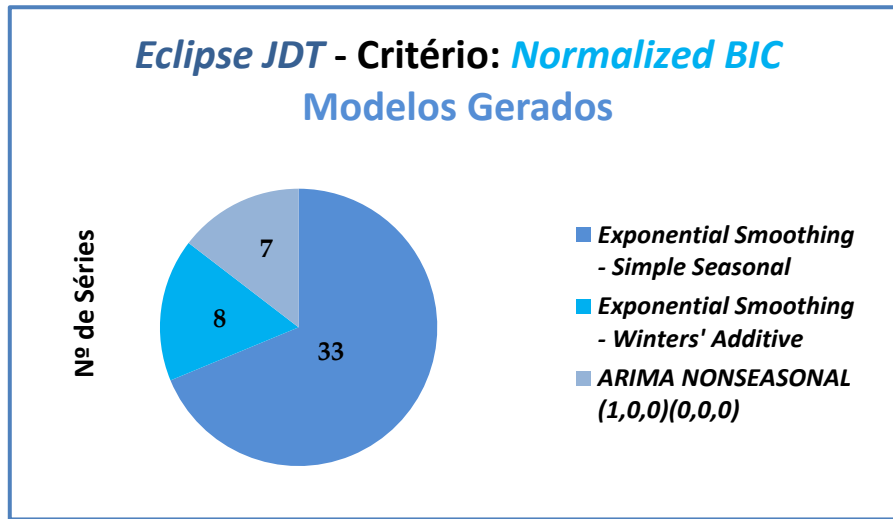


Figura 5.4 – Modelos gerados segundo o critério *Normalized BIC* (Eclipse JDT)

Na Figura 5.6 podemos verificar que o modelo *Exponential Smoothing - Simple Seasonal* (EXSS) se evidencia face aos restantes modelos gerados, nomeadamente: *Exponential Smoothing - Winters' Additive* (EXWA) e *ARIMA (1,0,0) (0,0,0)* (ANS). Das 48 séries temporais implícitas neste estudo, 33 correspondem ao modelo EXSS, 8 ao modelo EXWA e 7 ao modelo ANS. O modelo de previsão EXSS surge assim da 7ª à 36ª série temporal com novo registo na 46ª à 48ª série. O modelo EXWA pode observar-se não só da 1ª à 6ª série temporal, como também da 37ª à 38ª série. Por último, o modelo ANS (1,0,0) (0,0,0) foi selecionado da 39ª à 45ª série temporal.

5.1.7. Eclipse JDT – Comparação das métricas de ajuste, de erro e de normalização segundo os critérios *RSQUARE*, *MAPE* e *Normalized BIC*

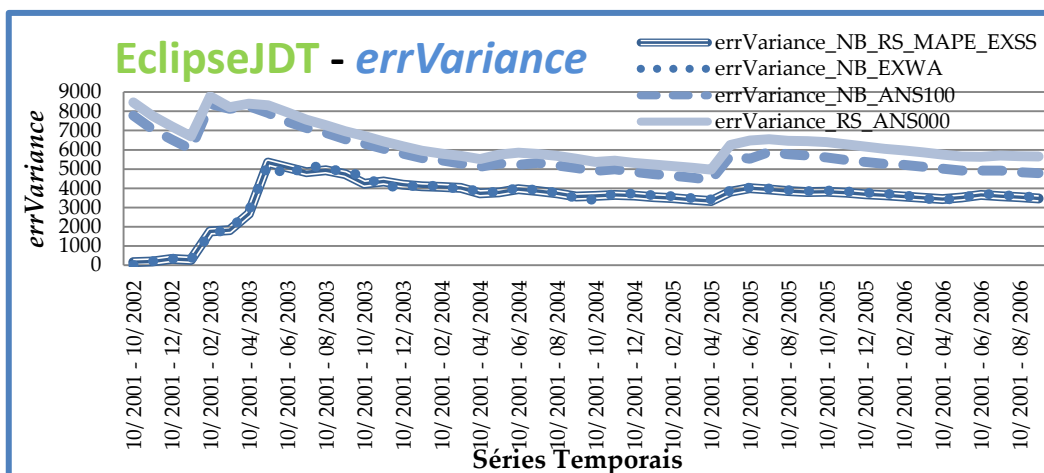


Figura 5.5 – Métrica de Erro *errVariance*

Pela análise da métrica de variância em cada um dos modelos selecionados (Figura 5.7) pode observar-se que os modelos de previsão EXSS e EXWA apresentam um comportamento muito semelhante, destacando-se negativamente os modelos ANS100 e ANS000. Pela consulta da tabela de *Excel* podemos constatar que o modelo **EXSS** apresenta uma média de valores (3476,806) inferior ao modelo EXWA (3506,035), o que torna **mais favorável**. A partir da série treinada com dados de 10/2001 a 5/2003 estes modelos apresentam uma tendência estável. Aliás, o erro vai aumentando até essa altura, começando, a partir daí, a decair muito lentamente.

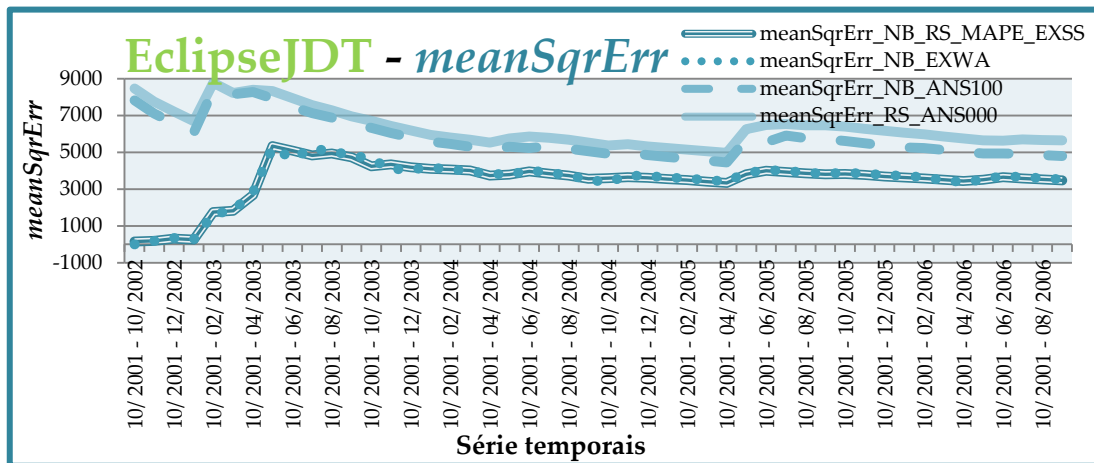


Figura 5.6 – Métrica de Erro *meanSqrErr*

Na Figura 5.8 a análise da métrica de erro quadrático médio em cada um dos modelos selecionados confirma que os modelos de previsão EXSS e EXWA apresentam um comportamento muito idêntico, evidenciando-se negativamente os modelos ANS100 e ANS000. Pela consulta da tabela de *Excel* podemos constatar que o modelo **EXSS** apresenta uma média de valores (3476,806) inferior ao modelo EXWA (3506,035), o que torna **mais benéfico**. A partir da série treinada com dados de 10/2001 a 5/2003 estes modelos apresentam uma tendência estável. Aliás, o erro vai aumentando até essa altura, começando, a partir daí, a decair muito lentamente.

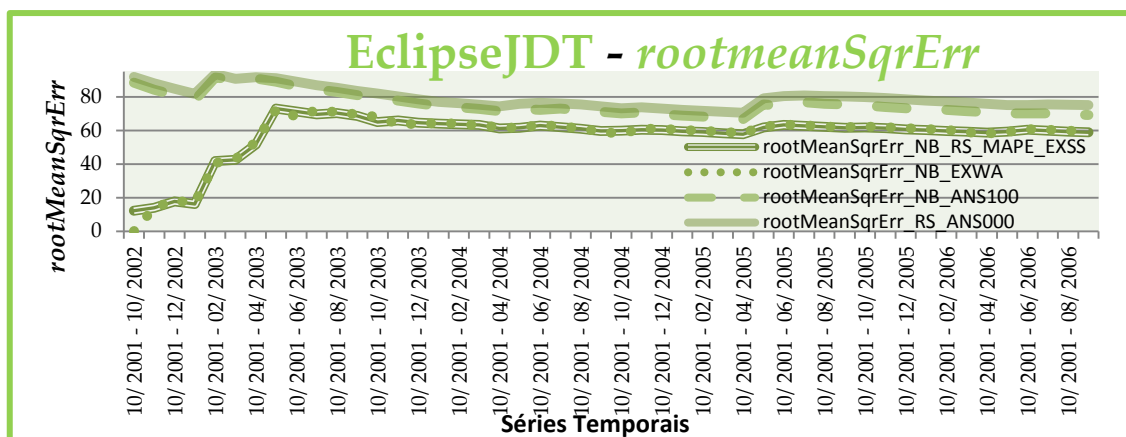


Figura 5.7 – Métrica de Erro *rootmeanSqrErr*

Como se pode observar na Figura 5.9 a análise da métrica de raiz quadrada do erro quadrático médio em cada um dos modelos selecionados atesta que os modelos de previsão EXSS e EXWA exibem novamente um comportamento análogo, comprovando-se que os modelos ANS100 e ANS000 apresentam mais vez um valor desfavorável. Pela consulta da tabela de *Excel* podemos verificar que o modelo **EXSS** apresenta uma média de valores (57,312) inferior ao modelo EXWA (57,337). Deste ponto de vista, estes modelos são essencialmente semelhantes. A partir da série treinada com dados de 10/2001 a 5/2003 estes modelos apresentam uma tendência estável. Aliás, o erro vai aumentando até essa altura, começando, a partir daí, a decair muito lentamente.

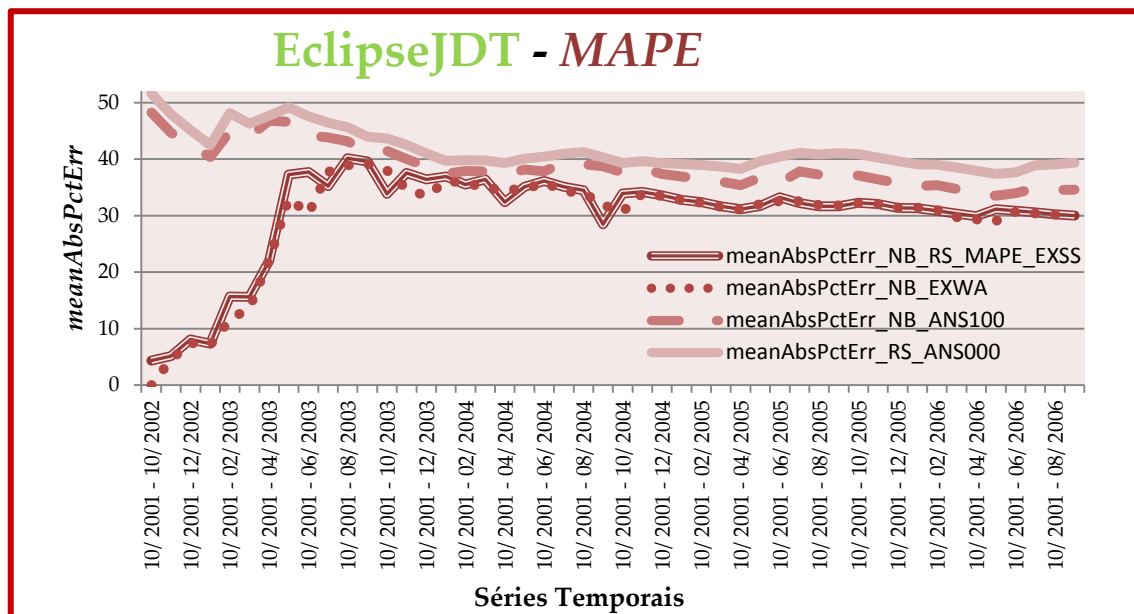


Figura 5.8 – Métrica de Erro MAPE

Como se pode observar na Figura 5.10 a métrica de erro médio percentual absoluto (MAPE³⁷) em cada um dos modelos gerados atesta que os modelos de previsão EXSS e EXWA exibem novamente um comportamento análogo, comprovando-se que os modelos ANS100 e ANS000 exibem mais vez um valor mais elevado. Pela consulta da tabela de *Excel* podemos verificar que o modelo **EXWA** apresenta uma média de valores (29,536) inferior ao modelo EXSS (30,154). Deste ponto de vista, estes modelos são essencialmente semelhantes. A partir da série treinada com dados de 10/2001 a 5/2003 estes modelos apresentam uma tendência estável. Aliás, o erro vai aumentando até essa altura, começando, a partir daí, a decair muito lentamente.

³⁷ MeanAbsPctErr.

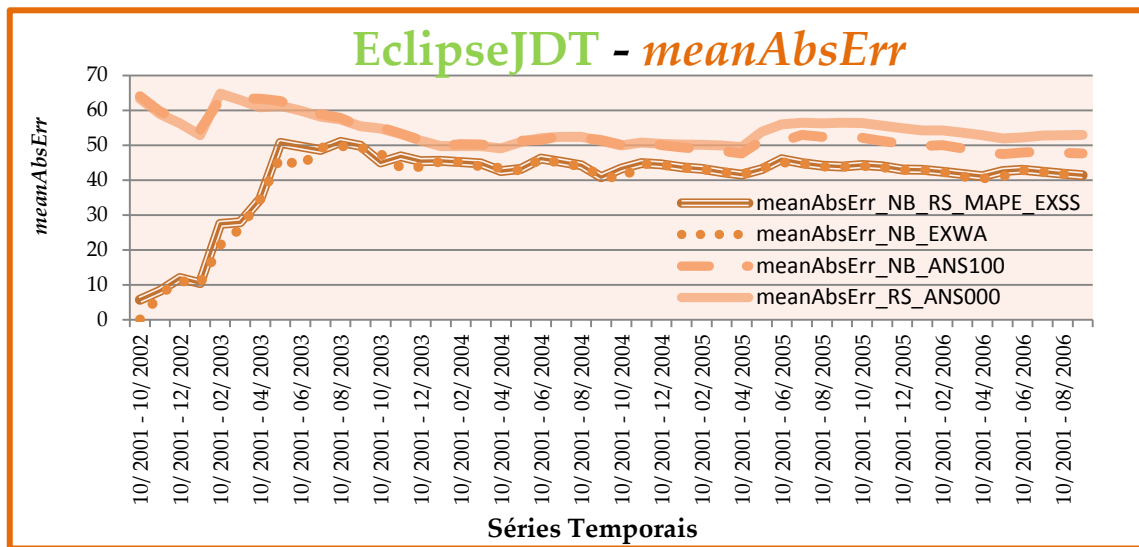


Figura 5.9 – Métrica de Erro *meanAbsErr*

A representação gráfica da métrica de erro médio absoluto em cada um dos modelos gerados (Figura 5.11) indica-nos que os modelos de previsão EXSS e EXWA apresentam novamente um comportamento similar, confirmando-se que os modelos ANS100 e ANS000 expõem novamente um valor mais elevado. Pela consulta da tabela de *Excel* podemos atestar que o modelo **EXWA** apresenta uma média de valores (39,775) inferior ao modelo EXSS (40,574). Deste ponto de vista, estes modelos são essencialmente semelhantes. A partir da série treinada com dados de 10/2001 a 5/2003 estes modelos apresentam uma tendência estável. Aliás, o erro vai aumentando até essa altura, começando, a partir daí, a decair muito lentamente.

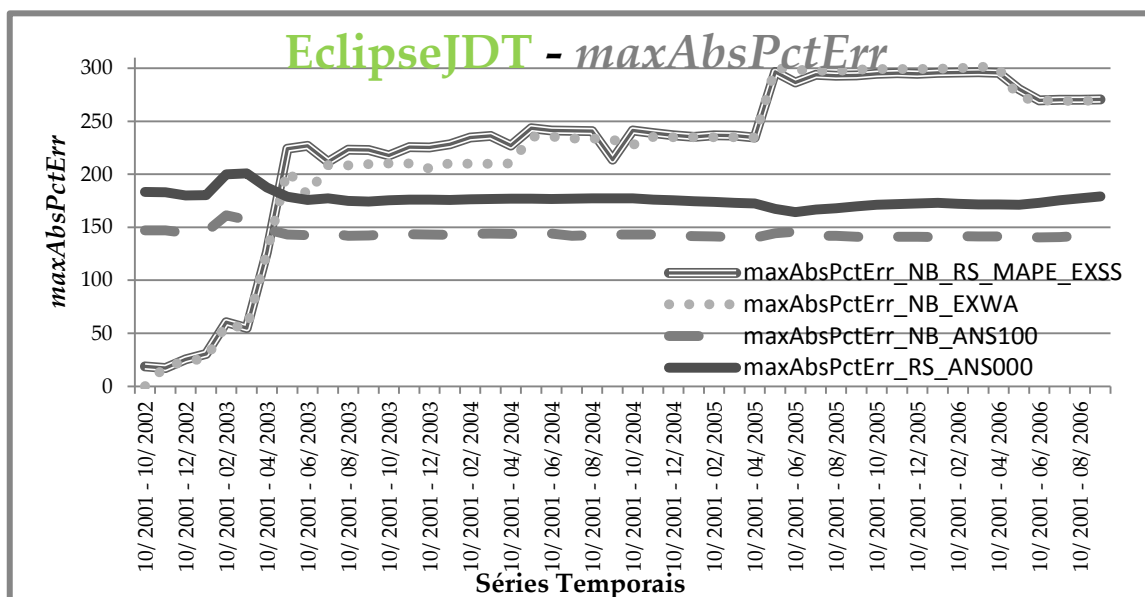


Figura 5.10 – Métrica de Erro *maxAbsPctErr*

Como se pode analisar na Figura 5.12 a métrica de erro percentual máximo absoluto em cada um dos modelos gerados demonstra que os modelos de previsão EXSS e EXWA exibem novamente um comportamento análogo, mas desta vez pior que os modelos ANS100 e ANS000. Pela consulta da tabela de *Excel* podemos verificar que o modelo **ANS100** obtém uma **média** de valores (143,514) **inferior aos restantes modelos**. Os modelos ANS100 e ANS000 apresentam uma tendência estável ao longo das séries temporais. Nos restantes modelos, a partir da série treinada com dados de 10/2001 a 5/2003, o erro vai aumentando até essa altura, começando, a partir daí, a apresentar um crescimento mais lento até à série treinada com dados de 10/2001 a 5/2005. A partir dessa série verifica-se uma tendência de estagnação até 5/2006, passando a decair lentamente a partir daí.

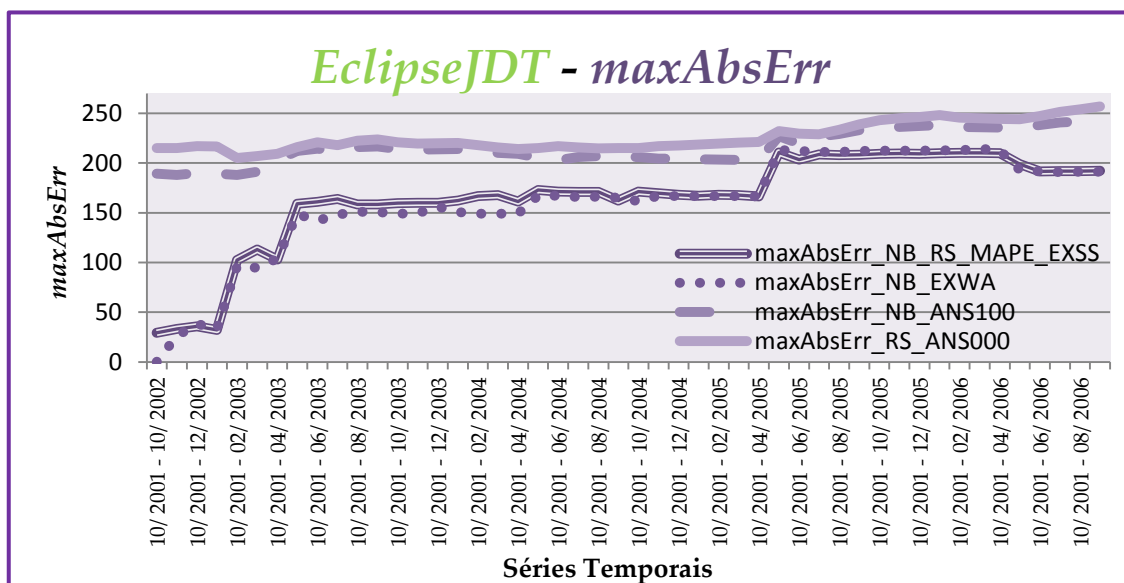


Figura 5.11 – Métrica de Erro $maxAbsErr$

Como se pode analisar na Figura 5.13 a métrica de erro máximo absoluto em cada um dos modelos gerados explica que os modelos de previsão EXSS e EXWA mostram novamente um comportamento análogo, melhor que os modelos ANS100 e ANS000. Pela consulta da tabela de *Excel* podemos verificar que o modelo EXWA apresenta uma média de valores (160,174) inferior ao modelo EXSS (164,354), o que torna mais favorável. A partir da série treinada com dados de 10/2001 a 5/2003 estes modelos apresentam uma tendência estável até à série temporal de 10/2001 a 4/2005. A partir daí, o erro vai aumentando lentamente, mantendo-se estável até à série de 10/2001 a 4/2006. A seguir, decai lentamente.

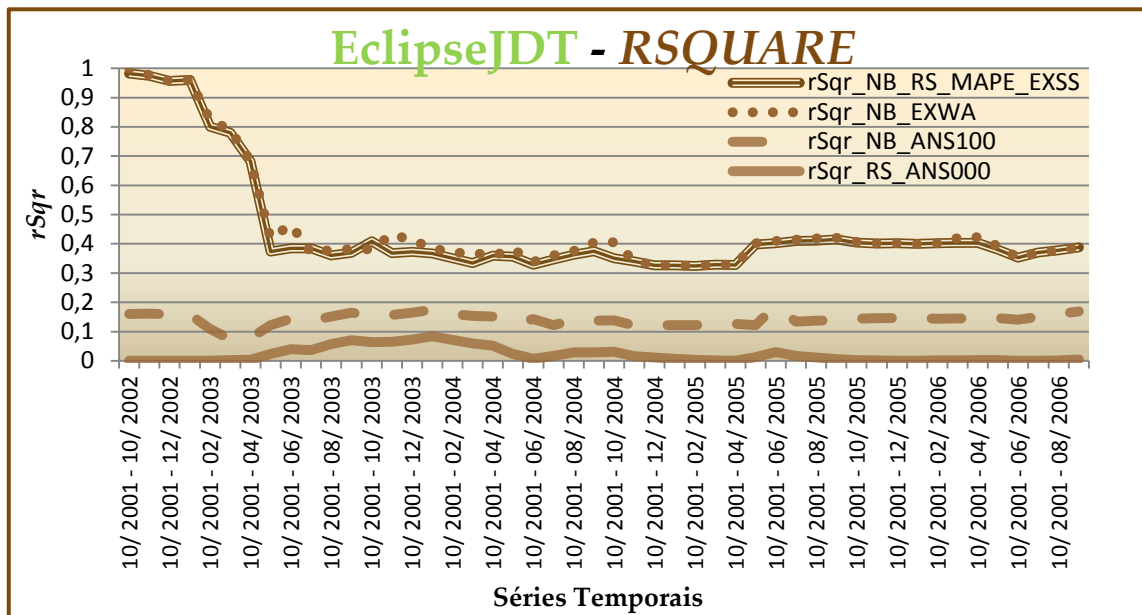


Figura 5.12 – Métrica de ajuste *RSQUARE*

Na Figura 5.14 a métrica de ajuste raiz quadrada em cada um dos modelos gerados explana que os modelos de previsão EXSS e EXWA apresentam novamente um comportamento análogo, melhor que os modelos ANS100 e ANS000. Pela consulta da tabela de *Excel* podemos verificar que o modelo **EXWA** apresenta uma média de valores (0,458) superior ao modelo EXSS (0,446), o que torna melhor, **apesar dos resultados serem muito próximos e consideravelmente fracos**. Os restantes modelos assumem resultados muito baixos, cerca de 0,141 para o modelo ANS100 e 0,020 para o modelo ANS000. A partir da série treinada com dados de 10/2001 a 5/2003 os modelos EXWA e EXSS apresentam uma tendência estável. Até então, o erro vai diminuindo.

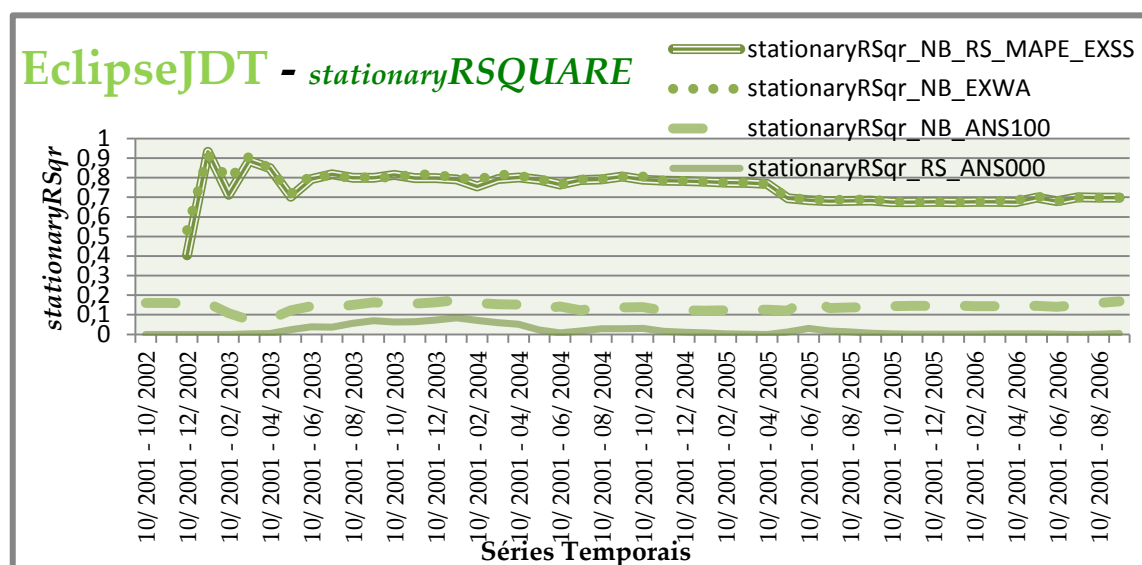


Figura 5.13 – Métrica de ajuste *stationaryRSQUARE*

Na Figura 5.15 a métrica de ajuste raiz quadrada estacionária em cada um dos modelos gerados elucidada que os modelos de previsão EXSS e EXWA apresentam um comportamento idêntico, melhor que os modelos ANS100 e ANS000. Pela consulta da tabela de *Excel* podemos verificar que o modelo **EXWA** apresenta uma média de valores (0,754) superior ao modelo EXSS (0,745), o que torna melhor, **apesar dos resultados serem muito próximos**. Os restantes modelos voltaram a exibir valores muito fracos. A partir da série treinada com dados de 10/2001 a 06/2003 os modelos EXWA e EXSS apresentam uma tendência estável. Aliás, o erro vai oscilando entre aumentos e decréscimos até essa altura.

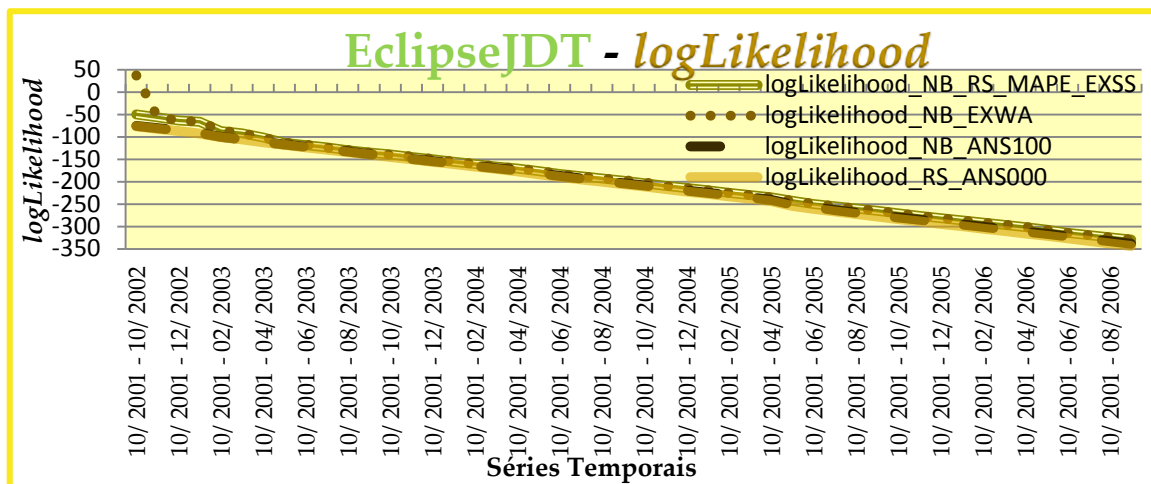


Figura 5.14 – Métrica *logLikelihood*

Na Figura 5.16 a métrica *log Likelihood* em cada um dos modelos gerados explica que os modelos de previsão apresentam um comportamento similar. Pela consulta da tabela de *Excel* podemos verificar que o modelo **EXWA** apresenta uma média de valores (-196,863), ligeiramente superior aos restantes modelos. A partir da série treinada com dados de 10/2001 a 10/2002 todos modelos exibem uma tendência linear decrescente.

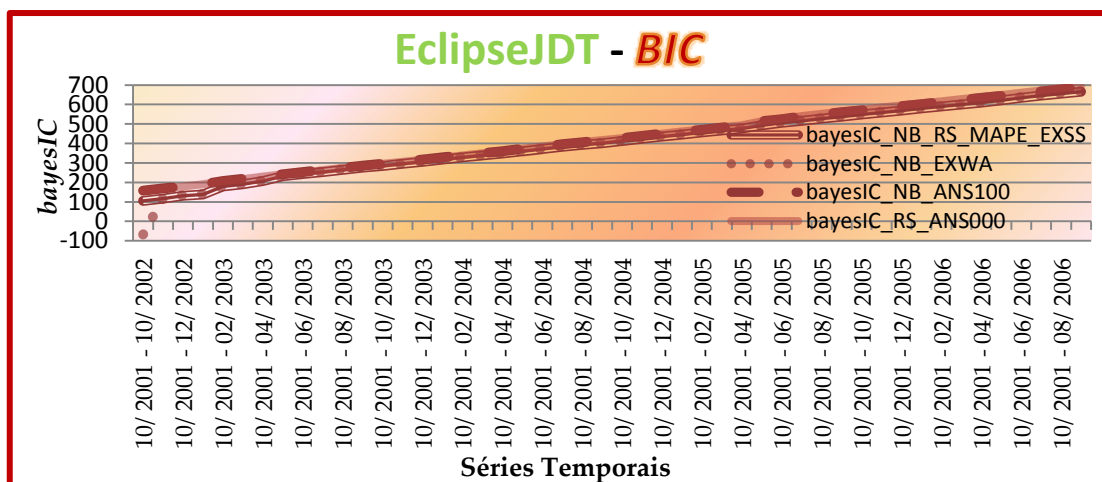


Figura 5.15 – Métrica *BIC*

Na Figura 5.17 o critério de informação *BIC* (*Bayesian Information Criterion*) em cada um dos modelos gerados esclarece que os modelos de previsão apresentam um comportamento idêntico. Pela consulta da tabela de *Excel* podemos verificar que o modelo **EXWA** apresenta uma média de valores (404,267) ligeiramente inferior aos restantes modelos, o que torna mais favorável. A partir da série treinada com dados de 10/2001 a 10/2002 todos modelos exibem uma tendência crescente.

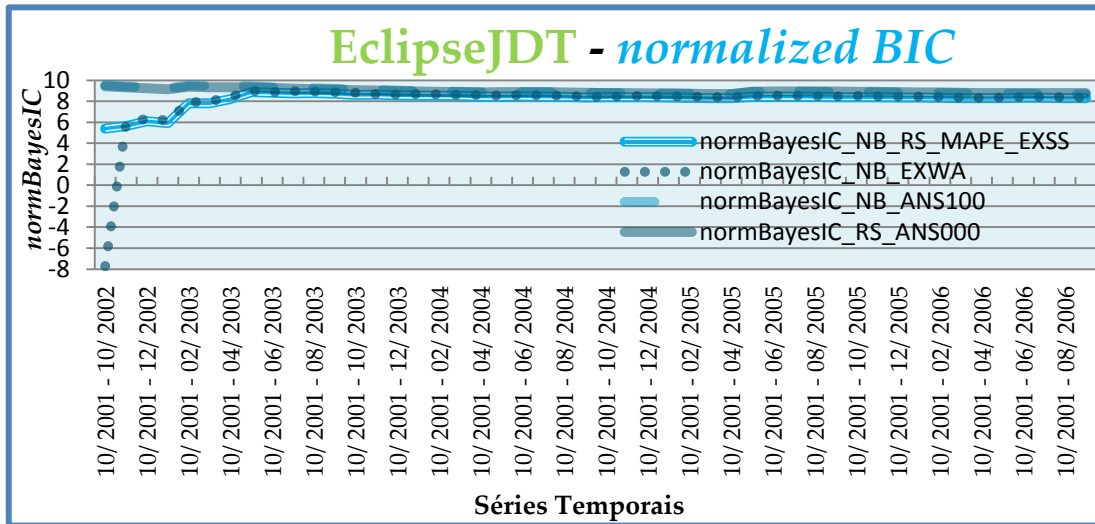


Figura 5.16 – Métrica *normalized BIC*

Na Figura 5.18 o critério de informação normalizado Bayesiano em cada um dos modelos gerados pelo critério *Normalized BIC* explica que os modelos de previsão apresentam um comportamento similar. Pela consulta da tabela de *Excel* podemos verificar que o modelo **EXWA** apresenta uma média de valores (8,04) inferior aos restantes modelos, o que torna **mais adequado**.

A partir da série treinada com dados de 10/2001 a 5/2003 todos modelos exibem uma tendência estável. Aliás, o erro vai aumentando até essa altura, nos modelos EXSS e EXWA.

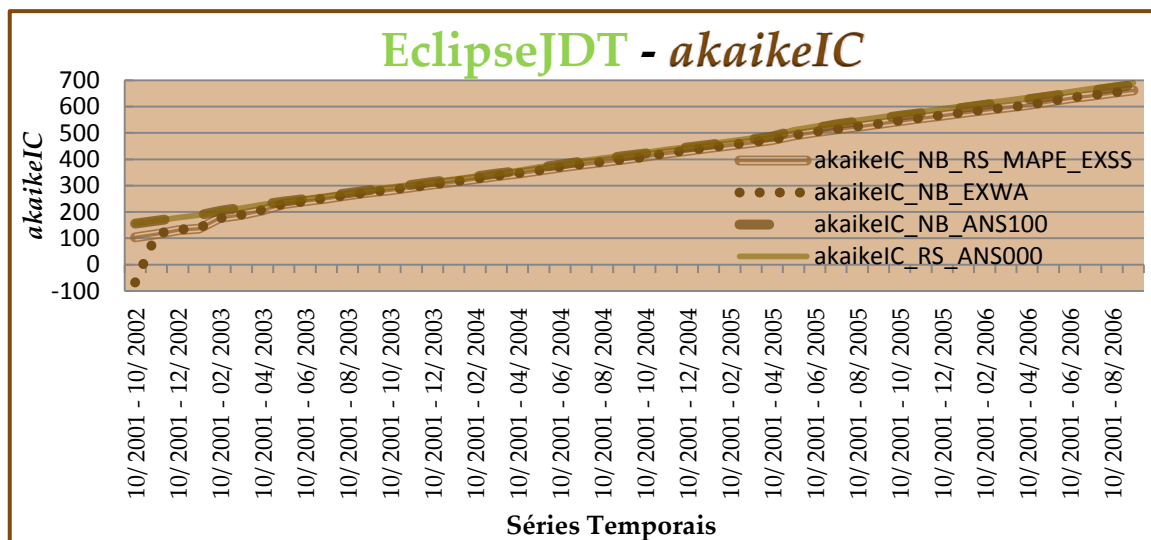


Figura 5.17 – Métrica *akaikeIC*

Na Figura 5.19 o critério de informação *akaike* em cada um dos modelos gerados elucidada que os modelos de previsão apresentam um comportamento semelhante. Pela consulta da tabela de *Excel* podemos verificar que o modelo **EXWA** apresenta uma média de valores (399,727) ligeiramente inferior aos restantes modelos, o que torna **mais satisfatório**.

A partir da série treinada com dados de 10/2001 a 10/2002 todos modelos exibem uma tendência crescente.

5.1.8. Eclipse JDT – Avaliação global dos modelos selecionados

Para fazermos a avaliação global dos modelos selecionados decidimos criar dois diagramas *Kiviat*³⁸, dado que neste tipo de gráficos é conveniente que todos os eixos meçam más ou boas características para que não se faça uma análise enganosa. Na Figura 5.20 estão representadas as métricas de erro e de normalização utilizadas no trabalho experimental, obedecendo à escala de 0 a 100%. Neste diagrama quanto maiores forem os valores apresentados pelas métricas referidas pior será a avaliação dos modelos de previsão. Na Figura 5.21 encontram-se representadas as métricas de ajuste e uma métrica de normalização também usadas na experiência, obedecendo à escala de 0 a 100%. Já neste diagrama quanto maiores forem os valores apresentados pelas métricas referenciadas melhor será a avaliação dos modelos de previsão. Pela consulta dos diagramas de *Kiviat* (Figura 5.20 e 5.21) podemos constatar que os modelos de previsão selecionados exibem um comportamento semelhante nas diferentes métricas estatísticas. No entanto, os modelos EXSS e EXWA conseguem evidenciar-se, sobretudo por assumirem valores de erros mais reduzidos e valores de ajuste mais elevados do que os restantes modelos. Estes dois modelos são, de facto, muito similares.

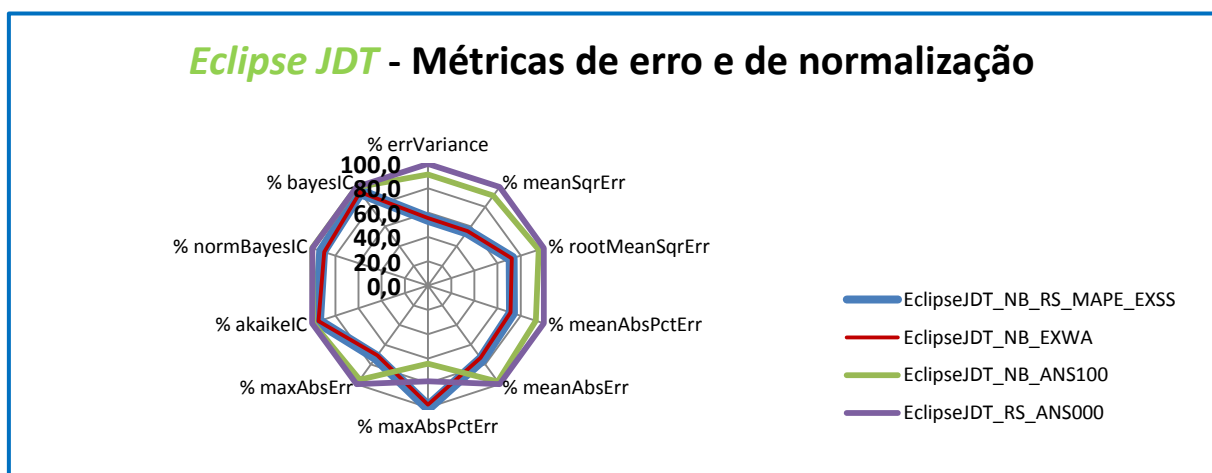


Figura 5.18 – Diagrama de *Kiviat* – Métricas de erro e de normalização

³⁸ <http://it.toolbox.com/blogs/enterprise-solutions/better-kiviat-diagrams-19868>

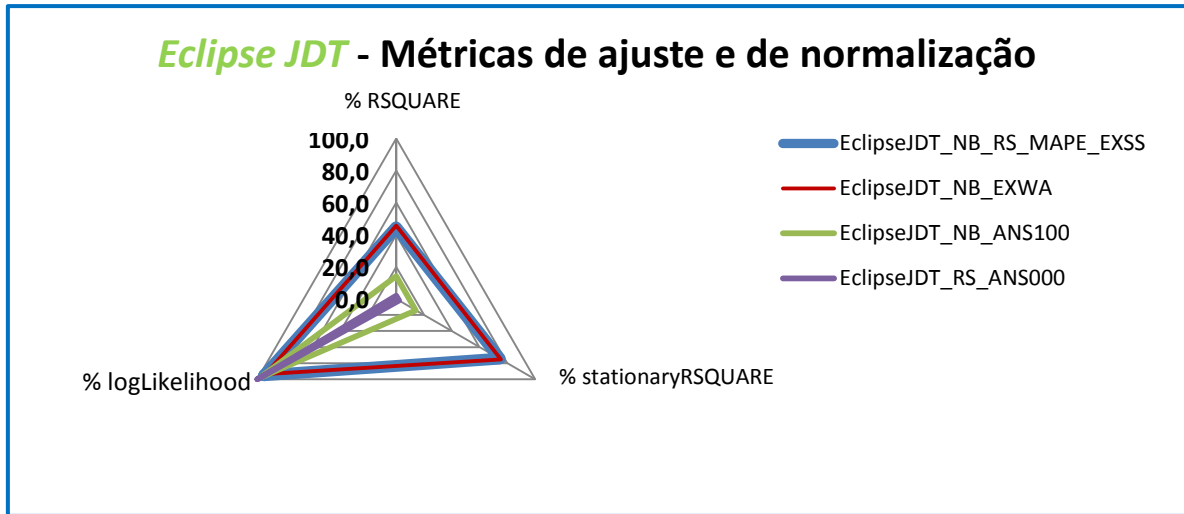


Figura 5.19 – Diagrama de Kiviat – Métricas de ajuste e de normalização

5.1.9. Eclipse JDT – Modelo emergente

Na Figura 5.22 apresentamos uma comparação da precisão das previsões, usando uma abordagem semelhante ao conhecido “cone de incerteza” de *Boehm* [38], mas que difere deste no sentido em que o final do projeto não tem data prevista. Podemos verificar que, em geral, existe uma sobrevalorização dos modelos seleccionados para este projeto.

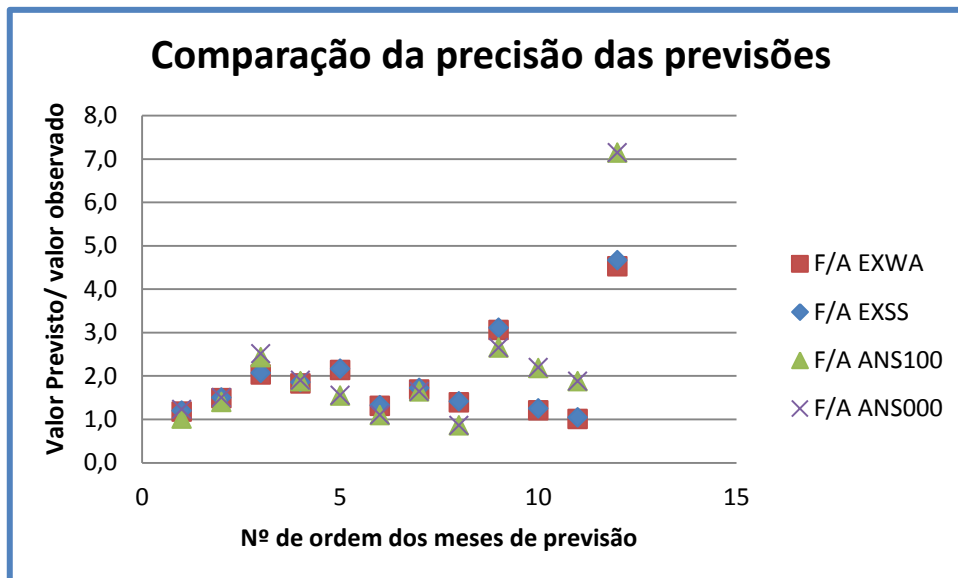


Figura 5.20 – Comparação da precisão das previsões

Neste projeto podemos confirmar que não existe propriamente um modelo emergente. Os modelos de previsão EXSS e EXWA tornam-se os mais satisfatórios face ao histórico de dados disponíveis,

mas a sua capacidade de explicação apenas se aproxima dos 45% e 46%, respetivamente. Apresentam ainda uma percentagem de erros consideravelmente elevada. Os valores das métricas de normalização também não são favoráveis.

5.1.10. Outros Projetos

Procede-se agora à apresentação da análise descritiva dos resultados obtidos para os restantes projetos. Importa referir que, todos os critérios de seleção (RSQUARE, MAPE e *Normalized BIC*) geraram os mesmos modelos de previsão para cada sistema.

5.1.10.1. Eclipse Platform

No sistema *Eclipse Platform* foram considerados os seus dados históricos abrangidos de outubro de 2001 a agosto de 2008. Da interpretação dos gráficos apresentados no Apêndice III e na Figura 5.23 podemos constatar que foram **gerados dois modelos de previsão**, nomeadamente: **EXWA** (*Exponential Winter's Additive*) e **EXWM** (*Exponential Smoothing – Winters' Multiplicative*). Tal facto, implica que neste sistema existe tendência e sazonalidade (subsecção 2.3.2.). Nestes gráficos apenas se encontra representada uma série, com um período de treino compreendido entre 10/2001 e 08/2007 e um período de previsão de 08/2007 a 08/2008. Os modelos exibiram uma oscilação **patente de subestimação e sobrestimação dos valores previstos, no período de previsão**. Pela análise realizada a partir dos gráficos de métricas de ajuste (Apêndice X), de erro (Apêndice XI) e de normalização (Apêndice XII) deduz-se que o **modelo EXWM apresenta melhores resultados, apesar de serem muito próximos dos resultados obtidos pelo outro modelo**.

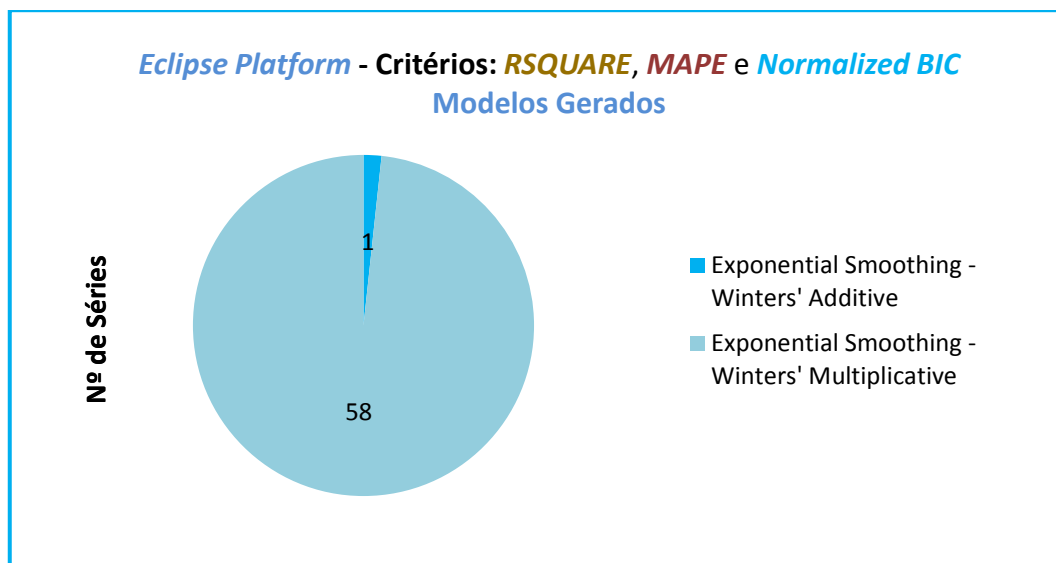


Figura 21.23 - Modelos gerados segundo o critério *RSQUARE* e *normalized BIC* (*Eclipse Platform*)

5.1.10.2. Eclipse

No sistema *Eclipse* foram estudados os seus dados históricos abrangidos de novembro de 2001 a março de 2010. Observa-se através dos gráficos exibidos no Apêndice IV e na Figura 5.24 que foram **gerados quatro modelos de previsão**, designadamente: **EXWA**, **EXSS** (*Exponential Smoothing – Simple Seasonal*), **ANS100** (*ARIMA (1,0,0) (0,0,0)*) e **AS001010** (*ARIMA (0,0,1) (0,1,0)*). Os modelos exibem uma **tendência crescente** na série representada (período de treino: **10/2001 – 3/2009** e período de previsão: **3/2009 – 3/2010**). Nos modelos EXWA, EXSS (subsecção 2.3.2.) e AS001010 (subsecção 2.3.3.) está patente a **existência de sazonalidade** na série. Importa referir que, todos os **modelos exibiram uma sobrestimação dos valores previstos, no período de previsão**. Pela análise realizada a partir dos gráficos de métricas de ajuste (Apêndice XIII), de erro (Apêndice XIV) e de normalização (Apêndice XV) deduz-se que o **modelo EXWA exibe melhores resultados, embora pouco significativos, sobretudo em relação ao modelo EXSS**.

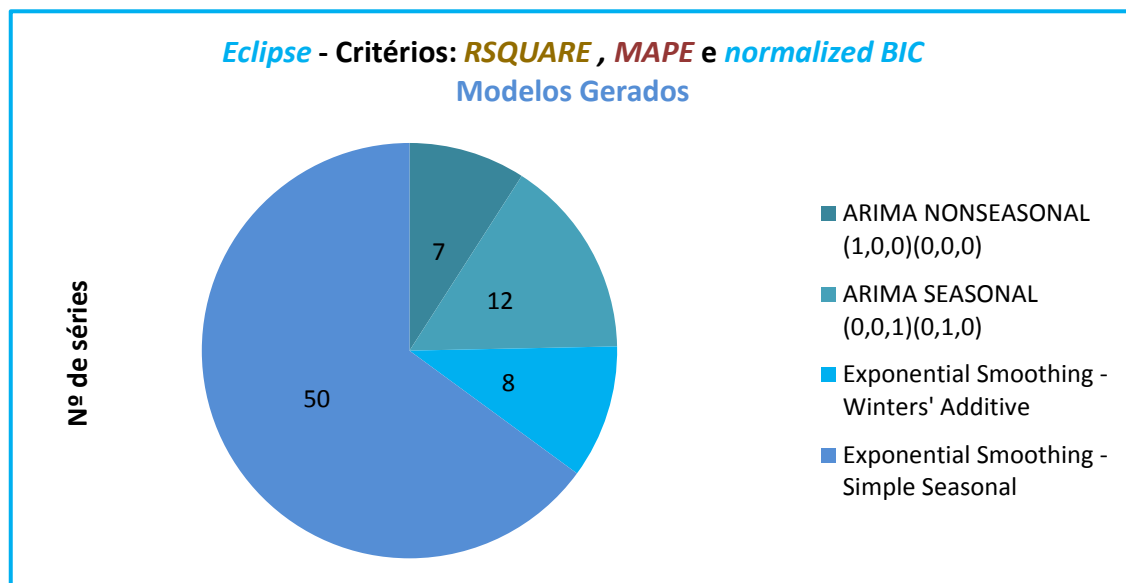


Figura 22.24 - Modelos gerados segundo o critério *RSQUARE*, *MAPE* e *normalized BIC* (*Eclipse*)

5.1.10.3. Eclipse Android

No sistema *Eclipse Android* foram observados os seus dados históricos abrangidos de novembro de 2007 a setembro de 2011. Pode observar-se através dos gráficos exibidos no Apêndice V e na Figura 5.25 que foram **gerados dois modelos de previsão**, designadamente: **EXWA** e **ANS010** (*ARIMA (0,1,0) (0,0,0)*). Todos os modelos exibem uma **tendência crescente** na série representada (período de treino: **11/2007 – 9/2010** e período de previsão: **9/2010 – 9/2011**). Importa referir que, os **modelos exibiram uma sobrestimação dos valores previstos, no período de previsão**. No

modelo EXWA está patente a **existência de sazonalidade** na série. Pela análise realizada a partir dos gráficos de métricas de ajuste (Apêndice XIII), de erro (Apêndice XIV) e de normalização (Apêndice XV) deduz-se que o **modelo EXWA** **exibe resultados mais satisfatórios, embora pouco significativos, sobretudo em relação ao modelo ANS010.**

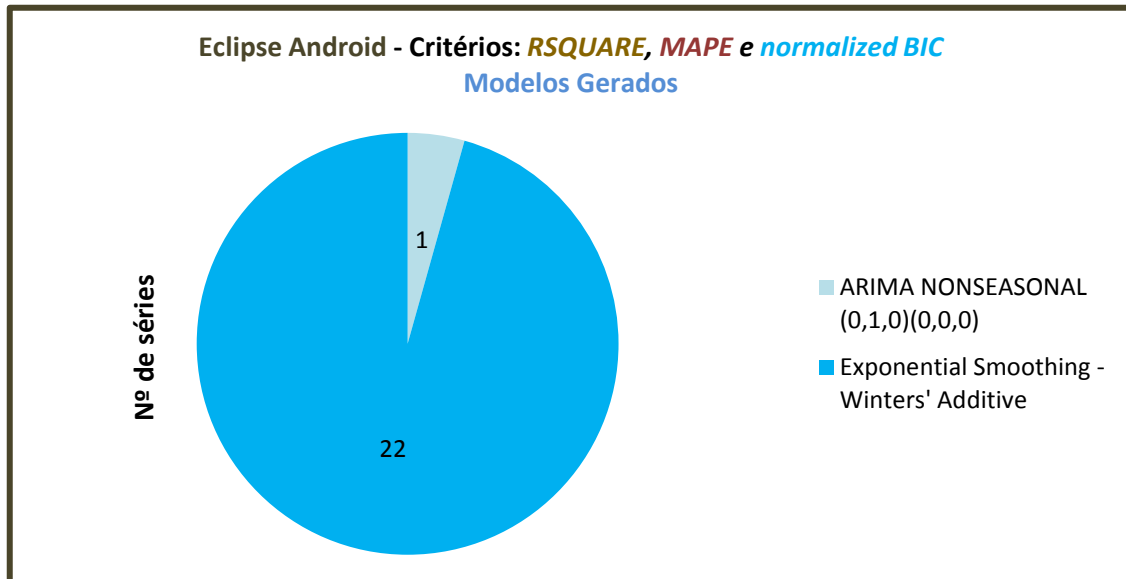


Figura 23.25 - Modelos gerados segundo o critério *RSQUARE*, *MAPE* e *normalized BIC* (Eclipse Android)

5.1.10.4. Gnome Evolution

No sistema *Gnome Evolution* foram analisados os seus dados históricos abrangidos de maio de 2000 a julho de 2008. Pela visualização dos gráficos exibidos no Apêndice VI e na Figura 5.26 que foram **gerados seis modelos de previsão**, a saber: **EXWA**, **EXSS**, **EXS** (*Exponential Smoothing – Simple*), **ANS010** (*ARIMA (0,1,0) (0,0,0)*), **ANS011** (*ARIMA (0,1,1) (0,0,0)*) e **ANS200** (*ARIMA (2,0,0) (0,0,0)*). Todos os modelos exibem uma **tendência linear monótona** (com aumentos e diminuições graduais) na série representada (período de treino: **5/2000 – 7/2007** e período de previsão: **7/2007 – 7/2008**). As previsões são próximas dos valores observados. Nos modelos EXWA e EXSS está manifesta a **existência de sazonalidade** na série. Pela análise realizada a partir dos gráficos de métricas de ajuste (Apêndice XIX), de erro (Apêndice XX) e de normalização (Apêndice XXI) infere-se que o **modelo EXSS** **exibe resultados mais satisfatórios, embora pouco relevantes.**

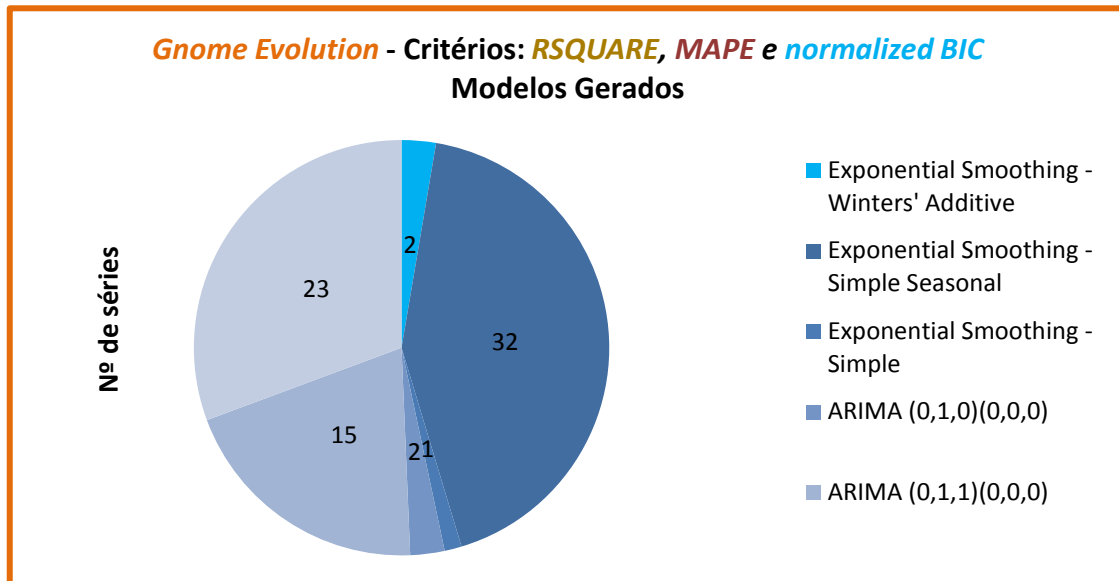


Figura 24.26 - Modelos gerados segundo o critério *RSQUARE*, *MAPE* e *normalized BIC* (*Gnome Evolution*)

5.1.10.5. *Gnome GStreamer*

No sistema *Gnome GStreamer* foram estudados os seus dados históricos compreendidos entre abril de 2002 e agosto de 2008. Observa-se através dos gráficos exibidos no Apêndice VII e na Figura 5.27 que foram **gerados cinco modelos de previsão**, designadamente: **EXWA**, **EXSS**, **ANS000** (*ARIMA* (0,0,0) (0,0,0)), **ANS010** (*ARIMA* (0,1,0) (0,0,0)) e **ANS011** (*ARIMA* (0,1,1) (0,0,0)). Todos os modelos exibem uma **tendência de subestimação e de sobrestimação** ao longo da série representada (período de treino: **4/2002 – 8/2007** e período de previsão: **8/2007 – 8/2008**). Nos modelos EXWA e EXSS é evidente a **existência de sazonalidade** na série. Pela análise efetuada a partir dos gráficos de métricas de ajuste (Apêndice XXII), de erro (Apêndice XXIII) e de normalização (Apêndice XXIV) depreende-se que o **modelo EXWA apresenta melhores resultados, embora pouco significativos, principalmente em relação ao modelo EXSS**.

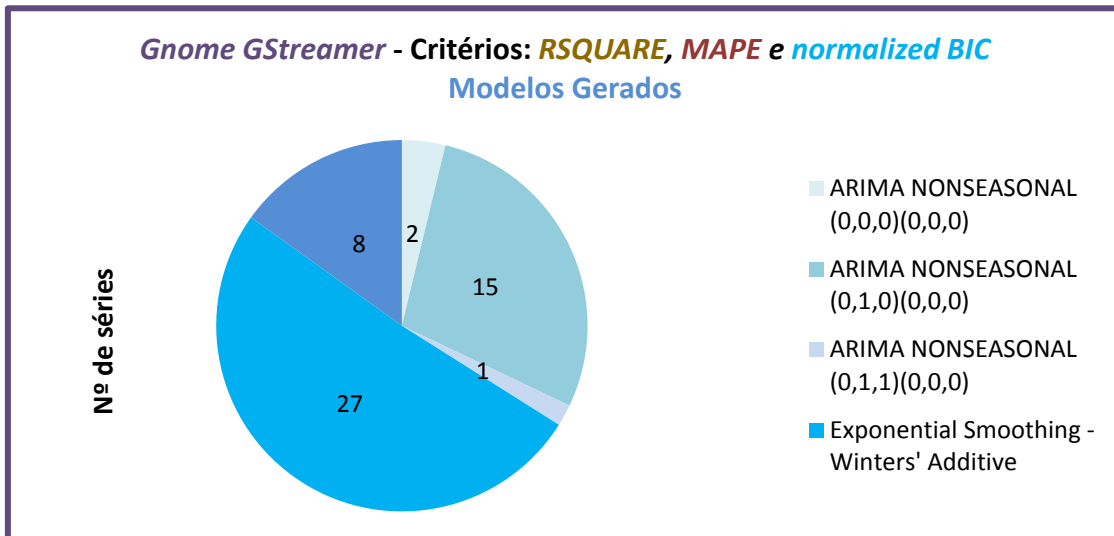


Figura 25.27 - Modelos gerados segundo o critério *RSQUARE*, *MAPE* e *normalized BIC* (Gnome GStreamer)

5.1.10.6. Mozilla Core

No sistema *Mozilla Core* foram analisados os seus dados históricos abrangidos de outubro de 2001 a setembro de 2007. Pode comprovar-se que através dos gráficos exibidos no Apêndice VIII e na Figura 5.28 que foram **gerados dois modelos de previsão**, designadamente: **EXSS** e **ANS000** (*ARIMA (0,0,0) (0,0,0)*). Os dois modelos exibem uma **sobrestimação** no período de previsão de **9/2006 a 9/2007**. No modelo EXWA verifica-se a **existência de sazonalidade** na série. Pela análise realizada a partir dos gráficos de métricas de ajuste (Apêndice XXV), de erro (Apêndice XXVI) e de normalização (Apêndice XXVII) conclui-se que o **modelo EXSS** **exibe melhores resultados, contudo pouco significativos**.

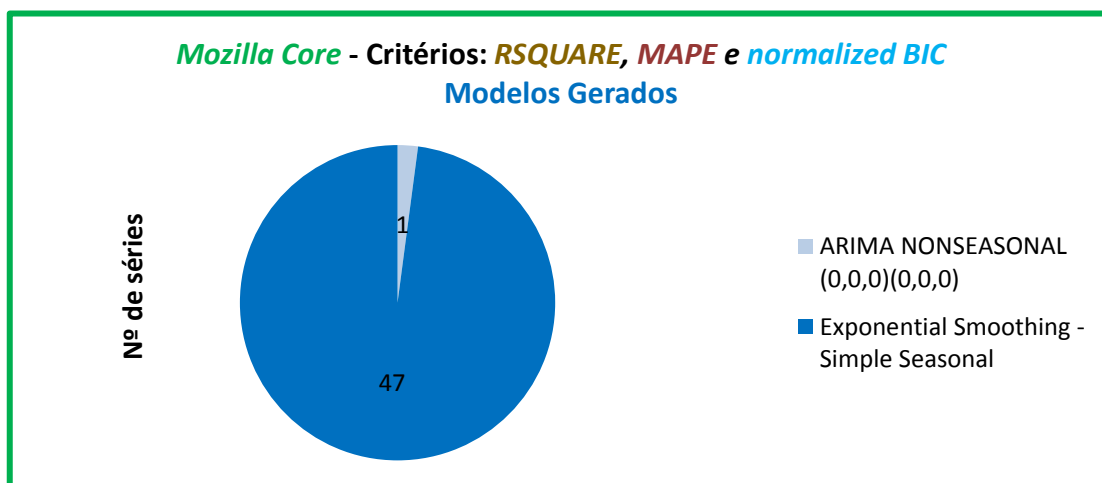


Figura 26.28 - Modelos gerados segundo o critério *RSQUARE*, *MAPE* e *normalized BIC* (Mozilla Core)

5.1.10.7. *Mozilla Firefox*

No sistema *Mozilla Firefox* foram analisados os seus dados históricos abrangidos de abril de 2001 a julho de 2008. Observa-se através dos gráficos exibidos no Apêndice IX e na Figura 5.29 que foram gerados seis modelos de previsão, designadamente: **EXWA**, **EXSS**, **ANS000** (*ARIMA (0,0,0) (0,0,0)*), **ANS010** (*ARIMA (0,1,0) (0,0,0)*), **ANS020** (*ARIMA (0,2,0) (0,0,0)*) e **ANS030** (*ARIMA (0,3,0) (0,0,0)*). Todos os modelos exibem uma **subestimação do número de pedidos de evolução previsto**, no período de previsão compreendido entre 7/2007 e 7/2008. Nos modelos EXWA e EXSS confirma-se a **existência de sazonalidade** na série. Pela análise feita a partir dos gráficos de métricas de ajuste (Apêndice XXVIII), de erro (Apêndice XXIX) e de normalização (Apêndice XXX) deduz-se que o **modelo EXWA** **exibe melhores resultados, embora pouco reveladores, sobretudo em relação ao modelo EXSS**.

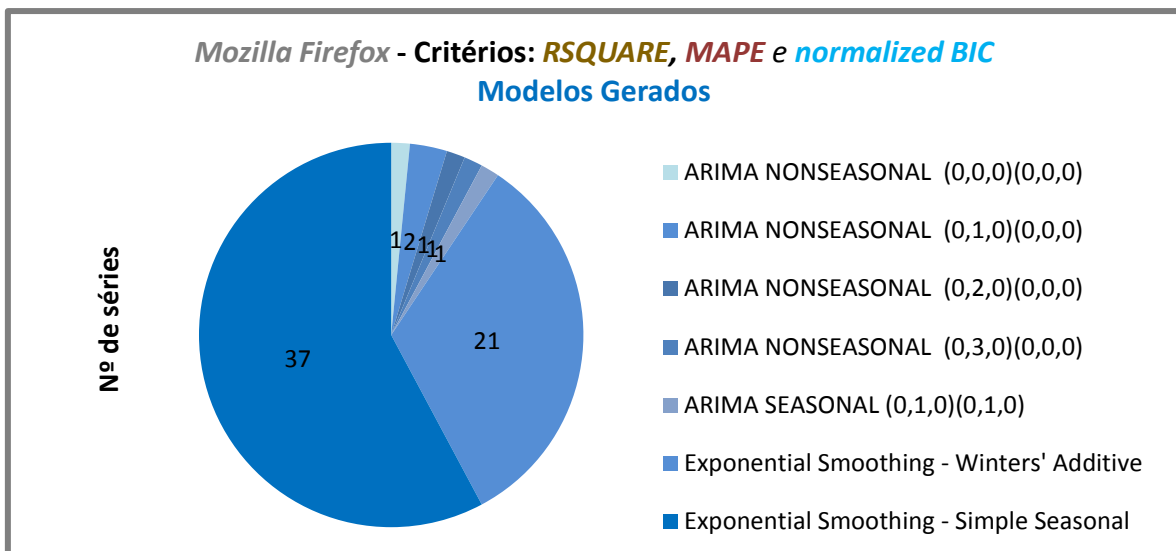


Figura 27.29 - Modelos gerados segundo o critério *RSQUARE*, *MAPE* e *normalized BIC* (*Mozilla Firefox*)



Interpretação de resultados

Neste capítulo apresentamos uma síntese reflexiva sobre os resultados dos dados recolhidos, implícitos no nosso objetivo geral que se traduz na escolha do melhor modelo de previsão de distribuição de pedidos de evolução de *software*, em cada momento, para cada projeto. Este estudo emergiu das dificuldades e dos constrangimentos experimentados pelos gestores de projetos e restantes membros das equipas de desenvolvimento de *software*, no que concerne à afetação apropriada de todos os recursos humanos, materiais, tecnológicos e financeiros aos pedidos de evolução de *software*, a cumprir num período de tempo definido.

6.1. Avaliação dos resultados e implicações

Tendo como referência os objetivos que nortearam a investigação (subsecção 4.2.1.), apresentamos a seguir os principais resultados alcançados e que consideramos pertinente evidenciar neste contexto em jeito de conclusão.

Para responder à nossa primeira questão de investigação (**Q1**) (subsecção 4.1.4.) podemos mencionar que os modelos de previsão que maximizam a capacidade de explicação do número de pedidos de evolução para cada sistema, com a finalidade de nos permitir analisar a evolução desses modelos nesses sistemas, em busca de padrões recorrentes, são os que constam da Tabela 6.1. Importa referir que os valores apresentados nesta tabela são os valores médios das métricas RSQUARE e *Stationary* RSQUARE. No projeto *Eclipse Android* a métrica *Stationary* RSQUARE apresenta um valor muito baixo no modelo ANS010, pois a mesma só é preferível à raiz quadrada comum, quando há uma tendência ou padrão sazonal, o que não se verifica neste caso.

Modelos de Previsão	RSQUARE	Stationary RSQUARE	Sistema	Escolha do Melhor Modelo
EXWA	45,8%	75,4%	Eclipse JDT	Estável
EXSS	44,6%	74,5%		Estável
EXWM	72,4%	79,9%	Eclipse Platform	Estável
EXWA	56,9%	80,7%		Estável
EXWA	74,5%	66,3%	Eclipse	Estável
EXSS	73,5%	64,9%		Estável
EXWA	85,2%	74,2%	Eclipse Android	Estável
ANS010	64,9%	0,7%		Volátil
EXSS	74,4%	61,5%	Gnome Evolution	Estável
EXWA	74,3%	62,2%		Estável
EXWA	73,1%	70,3%	Gnome GStreamer	Estável
EXSS	66,9%	66,8%		Estável
EXSS	37,9%	74,6%	Mozilla Core	Estável
EXWA	75,1%	65,1%	Mozilla Firefox	Estável
EXSS	72,6%	61,7%		Estável
ANS000	51,7%	51,7%		Estável

Tabela 6.1 – Modelos que maximizam a capacidade de explicação do nº de pedidos de evolução

Na realidade, podemos concluir que a escolha do “melhor” modelo de previsão é predominantemente estável para cada sistema, quando o critério usado é o da capacidade de explicação da variação do número de pedidos de evolução.

No que diz respeito à segunda questão (Q2), os modelos de previsão que apresentam uma menor taxa de erro do número de pedidos de evolução para cada sistema, com a finalidade de nos permitir analisar a evolução desses modelos nesses sistemas, em busca de padrões recorrentes são os indicados na Tabela 6.2. A escolha destes modelos também é predominantemente estável para cada sistema, quando o critério usado é o do erro do modelo face aos dados reais.

Modelos de Previsão	MeanAbsPctErr (MAPE)	Sistema	Escolha do Melhor Modelo
EXWA	29,54%	Eclipse JDT	Estável
EXSS	30,15%		Estável
EXWM	50,22%	Eclipse Platform	Estável
EXWA	62,99%		Estável
EXWA	17,78%	Eclipse	Estável
EXSS	18,30%		Estável
EXWA	35,21%	Eclipse Android	Estável
ANS010	52,48%		Volátil
EXS	37,48%	Gnome Evolution	Estável
EXSS	47,87%		Estável
EXWA	50,45%		Estável
EXWA	32,02%	Gnome GStreamer	Estável
EXSS	32,26%		Estável
EXSS	62,06%	Mozilla Core	Estável
ANS010	45,00%	Mozilla Firefox	Estável

Tabela 6.2 – Modelos que apresentam uma menor taxa de erro do nº de pedidos de evolução

Podemos verificar ainda que as taxas de erro apresentadas pelos modelos são pouco satisfatórias, à exceção do sistema *Eclipse*. No entanto, representam uma base de trabalho para os elementos das equipas de desenvolvimento de *software* e para os investigadores.

Apesar de estar apresentado na Tabela 6.2 apenas a métrica de erro MAPE (terceiro critério de seleção de modelos), adiantamos que nas restantes métricas de erro os modelos identificados exibem igualmente valores menores em relação aos restantes.

Quanto à terceira questão de investigação (Q3) podemos constatar que, para o histórico de dados disponíveis de cada sistema, as previsões dos modelos *Exponential Smoothing* gerados são mais rigorosas e fiáveis do que as dos modelos ARIMA, tornando-se assim mais credíveis os seus resultados e as técnicas utilizadas. Face aos três critérios de seleção (subsecção 4.2.8.) que estabelecemos no nosso trabalho experimental, os modelos *Exponential Smoothing* reuniram, de facto, mais condições para se tornarem vencedores em comparação com os modelos ARIMA. Os modelos *Exponential Smoothing* são modelos que estão associados à simplicidade de utilização, a um baixo custo e a uma razoável precisão [25, 26], o que os torna mais favoráveis no nosso estudo. Os modelos ARIMA são, na realidade, modelos de maior complexidade (subsecção 2.3.3.).

Para responder à quarta questão de investigação (Q4) podemos considerar, através da consulta da Tabela 6.1, que os modelos que mais se evidenciaram nos oito projetos são modelos que incluem características de sazonalidade, nomeadamente os modelos: EXSS, EXWA e EXWM. Estes modelos que agregam informação sazonal tornaram-se assim dominantes (Apêndices III ao XXX), em geral, logo a partir do período inicial de treino, que tinha a duração de doze meses. Isto significa que, o primeiro modelo escolhido já incorporava informação sazonal. Ainda assim, a evolução do período temporal de dados históricos de cada sistema foi caracterizada na secção 5.1., e pudemos constatar que foram encontrados padrões sazonais pouco relevantes, apontando assim para um efeito sazonal reduzido.

Tendo em conta a quinta questão de investigação (Q5), podemos deduzir através da avaliação da distribuição do número de pedidos de evolução nos oito sistemas (subsecções de 5.1.1. a 5.1.4., Apêndices III a IX) que, os modelos que apresentam características de tendência específica (crescente e decrescente) são, assinaladamente, os modelos: EXSS, EXWA e EXWM.

Por fim, respondendo à questão principal desta investigação, consideramos que o tipo de modelo que melhor se adequa à previsão dos pedidos de evolução é, em geral, o *Exponential Smoothing*, mais concretamente, o modelo **EXWA** (*Exponential Smoothing – Winters' Additive*). Este modelo apresenta-se em todos os sistemas estudados, exceto no projeto *Mozilla Core*, como sendo um modelo de previsão com melhor capacidade de explicação da evolução do número de pedidos de *software* (Tabela 6.1), atinge também as menores taxas de erro, na maioria dos casos (Tabela 6.2), e os valores alcançados nas métricas de normalização (Apêndices XII, XV, XVIII, XXI, XXIV,

XXX) confirmam não só a conhecida simplicidade deste modelo, como também o ajustamento do modelo às séries temporais em causa. Relativamente ao sistema *Mozilla Core*, o modelo EXSS é o modelo de previsão considerado menos mau, mas os seus valores neste projeto são francamente fracos, apontando assim para um projecto cujos dados históricos são relativamente pouco úteis na previsão dos pedidos de evolução, quando comparado com os restantes. No entanto, o modelo EXSS também se destaca em mais cinco projetos (Tabelas 6.1 e 6.2) com resultados bastante significativos.

6.2. Ameaças à validade

Sendo consensual a necessidade de escolha e construção do melhor modelo de previsão de distribuição de pedidos de evolução de *software* ao longo do tempo, por parte dos gestores de projeto e da restante equipa de desenvolvimento de *software*, podemos constatar que apesar de termos obtido o EXWA como melhor modelo em sete sistemas em estudo e o EXSS noutro sistema, o que é certo é que, se o período de previsão ou o sistema em estudo fossem outros, poderíamos ter obtido outros resultados.

Do estudo aqui apresentado foram sendo sucessivamente discutidos vários aspetos específicos, que quando conjugados, contribuem para potenciar a escolha do melhor modelo. No entanto, temos que ter presente que nenhuma técnica de previsão é perfeita e, como tal, existirão erros no processo de execução. Daí ser importante a realização de revisões contínuas das previsões.

Um aspeto importante e não negligenciável na nossa análise é que os modelos aqui apresentados foram construídos usando exclusivamente informações relativas ao número de pedidos de evolução registados até ao momento em que se iniciava o período de 12 meses de previsão. Assim, não foi usada qualquer informação relativa a outros aspetos que serão, muito provavelmente, relevantes para ajudar a prever a evolução dos pedidos. Por exemplo, não foi usada qualquer informação relativa ao calendário de apresentação de novas versões dos sistemas estudados, ou à evolução da dimensão da comunidade de utilizadores destes sistemas. Qualquer dos factores poderia, de forma plausível, afetar os resultados. Foi observado, em [18], que existe um incremento de pedidos de evolução associado aos momentos em que são feitas distribuições de novas versões. Dado que a escolha desses momentos não é necessariamente guiada por critérios de sazonalidade, é natural que os modelos com base em dados históricos tenham dificuldade em identificar um padrão associado às entregas de novas versões. Por outro lado, é razoável de admitir que uma flutuação significativa da base de utilizadores de um sistema tenha impacto no número de pedidos de evolução recebidos. Factores como estes ajudam a compreender os motivos pelos quais a taxa de explicação dos modelos não é mais alta, ou o erro médio não é mais baixo. Assim, uma das conclusões a tirar é que existe

informação relevante que os dados históricos do número de pedidos pura e simplesmente não capturam. No entanto, este estudo não deixa de ser uma referência para os elementos das equipas de desenvolvimento de *software* e para os investigadores, pois é sempre preferível ter uma base de trabalho, que possa ainda ser conjugada com outros tipos de informação para uma se chegar a uma solução mais fiável.

6.3. Inferências

Neste trabalho experimental pudemos identificar e caracterizar a distribuição do número de pedidos de evolução de oito sistemas (subsecção 4.2.3.), por meses, o que nos permitiu identificar um padrão recorrente na escolha do modelo de previsão. O modelo EXWA foi, de facto, um modelo dominante em determinadas fases do processo monitorizado. Outro dado importante a retirar deste estudo é que embora uma parte significativa da evolução do número de pedidos de evolução possa ser explicada com base nos dados históricos dos pedidos de evolução, existe informação relevante para a previsão que não é capturada por esses dados, como discutido na secção 6.2.

6.4. Aprendizagens

No que respeita à conceção metodológica deste estudo, a utilização de ferramentas (subsecção 4.2.7.) tornou-se bastante útil, não só na automatização de alguns processos como também na aprendizagem dos seus benefícios e limitações (secção 6.5.). A interoperabilidade entre ferramentas (e.g. java – SPSS, e vice-versa; SPSS – Excel, e vice-versa) foi fundamental e contribuiu bastante para agilizar o desenvolvimento do trabalho experimental.

6.5. Limitações do estudo

Este ponto resume as principais limitações metodológicas com que nos confrontámos no decorrer da realização deste trabalho.

Numa fase inicial foram contemplados na primeira aplicação desenvolvida em *Java* todos os modelos de previsão passíveis de análise no *software* SPSS, a saber: **ExpertModeler**, **ARIMA** (com seis parâmetros, cujos valores variaram de 0 a 5) e **Exponential Smoothing** (*Simple, Holts's linear Trend, Brown's linear trend, Damped Trend, Simple Seasonal, Winter's additive, Winter's multiplicative*). No decorrer do processo de execução dos ficheiros de sintaxe, deparámo-nos com um problema referente ao tipo de modelo ARIMA, pois os ficheiros .SPS correspondentes não foram possíveis de carregar no *software* SPSS. Cada ficheiro .SPS correspondeu a uma série

temporal. Cada *software* estudado incluiu um conjunto de várias séries temporais, entre 48 a 75. De facto, para cada série temporal, foram gerados 46.656 (6^6) modelos ARIMA diferentes. Por outras palavras, um ficheiro de dados com 46656 linhas e 13 colunas de métricas de qualidade e de erro, mais as colunas a identificar o sistema, o modelo, os períodos de treino e de previsão, os parâmetros do modelo, para cada série temporal, representariam entre 2.239.488 a 3.499.200 linhas para todas estas colunas, dependendo do *software* em causa. Perante este cenário, podemos constatar que o *software* SPSS não aguenta tantos dados. Para resolver o problema acima enunciado, decidimos trabalhar apenas com o tipo de modelo *Expert Modeler* (Tipo: *ARIMA* e *EXSMOOTH*).

Todavia, deparámo-nos com outra limitação que teve a ver com a ativação do subcomando *AUTOOUTLIER*, que permite detetar automaticamente valores atípicos, ou extremos na série temporal e excluir esses valores na geração do modelo. Este comando era ignorado com um aviso, se fizesse parte de um bloco do modelo que contém um subcomando *EXPERTMODELER* ou *EXSMOOTH*, portanto, na prática, não o podíamos usar com alguns dos modelos. A implicação desta limitação foi que não foi usada a técnica de exclusão de valores atípicos (*outliers*), ou extremos, em nenhuma das nossas séries temporais. Tipicamente, a exclusão destes valores melhora o desempenho dos modelos de estimação, dado que eles se afastam bastante dos valores “normais” da respetiva série, funcionando como um factor de “confusão” para a previsão. No entanto, não havendo motivos para suspeitar da qualidade dos dados que tínhamos à nossa disposição, assumimos que, atípicos ou não, os dados disponíveis deveriam ser considerados de modo consistente para todos os modelos. Assim sendo, a uma limitação prática juntou-se uma decisão técnica, no sentido de tornar a comparação entre modelos de previsão o mais justa possível. Deste modo, nenhum dos modelos estudados excluiu valores atípicos, ou extremos.



Considerações Finais

Neste capítulo apresentamos as principais conclusões do estudo empírico e algumas recomendações para estudos futuros.

7.1. Conclusões

A principal temática discutida ao longo da presente dissertação, escolha e construção do melhor modelo de previsão de distribuição de pedidos de evolução de *software* ao longo do tempo, pelos benefícios que lhe estão associados, apresenta-se como sendo um dos temas de discussão recorrentes em evolução de *software* (secção 3.1.).

Dado o enfoque definido e delimitado do presente estudo, consideramos que os nossos objetivos foram cumpridos (secção 6.1.) e que as contribuições previstas foram concretizadas, pois criámos um suporte de ferramentas para gerar uma análise semiautomática, construímos modelos de previsão para a distribuição de pedidos de evolução, com e sem informação sazonal, realizámos uma análise comparativa de diferentes tipos de previsão (*Exponential Smoothing* e *ARIMA*). Sendo o tipo de modelo *Exponential Smoothing* o mais adequado para os sistemas estudados, certificámo-nos que os dados mais recentes foram os mais importantes (subsecção 2.3.2.).

No que respeita à escolha do “melhor” modelo de previsão podemos constatar que é predominantemente estável para cada sistema, o que significa que na prática os gestores de projetos não têm que estar a fazer a seleção dos modelos com frequência, embora devam ir atualizando os dados disponíveis para os modelos selecionados. Verificámos ainda que os modelos EXWA e EXSS foram, de facto, dominantes na maioria dos sistemas (secção 6.1.), em geral, logo a partir do período inicial de treino, que tinha a duração de doze meses. Na verdade, o primeiro modelo escolhido na maioria dos sistemas já incorporava informação sazonal. Ainda assim, a evolução do período temporal de dados históricos de cada sistema foi caracterizada na secção 5.1., e pudemos verificar que foram encontrados padrões sazonais pouco relevantes, apontando assim para um efeito sazonal reduzido.

Por fim, importa salientar que as métricas estatísticas associadas aos três critérios de seleção dos modelos de previsão gerados (que, como vimos na subsecção 4.2.8., são a capacidade de explicação, a taxa de erro de previsão e o ajuste do modelo), apresentaram precisamente os mesmos resultados em sete projetos, e num deles ainda exibiram resultados parciais satisfatórios. Isto significa que a informação é consistente entre as métricas de seleção, e que para a escolha do melhor modelo de previsão seria suficiente apenas a utilização de um dos três critérios.

7.2. Trabalho futuro

Considerando as conclusões, as limitações e a pertinência do estudo realizado, é nosso entender que do mesmo poderão surgir novas linhas de investigação, das quais destacamos:

- Estudo específico para se averiguar se a sazonalidade que permite melhorar as nossas previsões será influenciada pelo agendamento de *releases*, mais do que pelo calendário. Se fôr, ao “distorcer” o tempo de modo a fazer a agregação de pedidos de evolução alinhar com o calendário de *releases*, em vez de alinhar com os meses do ano, conseguiremos previsões mais precisas. Se, pelo contrário, tal não acontecer, isso significa que as flutuações dependem mais de fatores externos ao projeto do que ao seu calendário de *releases*. Esta “distorção temporal” será, na prática, uma normalização dos períodos a usar na série temporal com base nas datas de publicação de novas versões.
- Estudo específico para avaliar o impacto da duração do período de previsão (por exemplo, de um ano para um mês) na seleção do modelo de previsão. Na realidade, seria natural que o modelo a escolher fosse outro, visto que o próprio prazo de previsão pode ter um impacto não negligenciável na escolha do modelo.
- Estudo que inclua novas variáveis explicativas, como a evolução da dimensão da comunidade de utilizadores do sistema para o qual se realiza a previsão.
- Alargamento deste estudo empírico a projetos comerciais.

Bibliografia

- [1] D'AMBROS, Marco. "Supporting software evolution analysis with historical dependencies and defect information," *Software Maintenance, 2008. ICSM 2008, IEEE International Conference*, pp.412-415, Sept. 28 2008-Oct. 4 2008.
- [2] FISCHER, Michael; GALL, Harald. "EvoGraph: A Lightweight Approach to Evolutionary and Structural Analysis of Large Software Systems," *Reverse Engineering, 2006. WCRE '06. 13th Working Conference*, pp.179-188, Oct. 2006.
- [3] FUENTETAJA, E.; Bagert, D.J. "Software evolution from a time-series perspective," *Software Maintenance, 2002. Proceedings. International Conference on*, pp. 226- 229, 2002.
- [4] PRESSMAN, Roger S.; Engenharia de Software, 5ª Edição, McGraw-Hill, 2002.
- [5] PARNAS, David Lorge. Software aging. In *Proceedings of the 16th international conference on Software engineering (ICSE '94)*, IEEE Computer Society Press, Los Alamitos, CA, USA, 279-287, 1994.
- [6] HASSAN, A.E.; XIE, T. "Software Intelligence: Future of Mining Software Engineering Data", In: Workshop on the Future of Software Engineering Research, 2010.
- [7] HASSAN, A.E. "Mining Software Repositories to Assist Developers and Support Managers", *Software Maintenance, ICSM '06, 22nd IEEE International Conference on*, pp.339-342, 24-27 Sept. 2006.
- [8] BHATT, Ganesh D. "Knowledge management in organizations: examining the interaction between technologies, techniques, and people", *Journal of Knowledge Management*, Vol. 5 Iss: 1, pp.68 – 75, 2001.
- [9] CHAPIN, N. et al. Types of software evolution and software maintenance, *Journal of Software Maintenance And Evolution*, vol:13 pg:3 -3 0, 2001.
- [10] LEHMAN, M.M. "Laws of software Evolution Revisited," Department of Computing, *Imperial College*, London, UK, 1996.
- [11] VOINEA, Lucian; TELEA, Alexandru. Visual data mining and analysis of software repositories, *Computers & Graphics*, Volume 31, Issue 3, Pages 410-428, June 2007.
- [12] VOINEA, Lucian; TELEA, Alexandru. Mining software repositories with CVSgrab, In *Proceedings of the 2006 international workshop on Mining software repositories (MSR '06)*, ACM, New York, NY, USA, 167-168, 2006.

- [13] WERMELINGER, Michel; YU, Yijun. Analyzing the evolution of eclipse plugins. In Proceedings of the 2008 international working conference on Mining software repositories (MSR '08). ACM, New York, NY, USA, 133-136, 2008.
- [14] SIY, H. et al. "Discovering Dynamic Developer Relationships from Software Version Histories by Time Series Segmentation," *Software Maintenance, 2007. ICSM 2007. IEEE International Conference on*, pp.415-424, 2-5 Oct. 2007.
- [15] TSANTALIS, N.; CHATZIGEORGIOU, A.; STEPHANIDES, G. "Predicting the probability of change in object-oriented systems," *Software Engineering, IEEE Transactions on*, vol.31, no.7, pp. 601- 614, July 2005.
- [16] PINZGER, Martin et al. Visualizing multiple evolution metrics. In *Proceedings of the 2005 ACM symposium on Software visualization (SoftVis '05)*. ACM, New York, NY, USA, 67-75. 2005.
- [17] NAGAPPAN, Nachiappan; BALL, Thomas, ZELLER, Andreas. Mining metrics to predict component failures. In Proceedings of the 28th international conference on Software engineering (ICSE '06). ACM, New York, NY, USA, 452-461. 2006.
- [18] GOULÃO, M. et al. "Software Evolution Prediction Using Seasonal Time Analysis: A Comparative Study," *Software Maintenance and Reengineering (CSMR), 2012 16th European Conference*, pp.213-222, 27-30 March 2012.
- [19] MACDONELL, Stephen G.; SHEPPERD, Martin. Data accumulation and software effort prediction. In *Proceedings of the 2010 ACM-IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM '10)*. ACM, New York, NY, USA, Article 31, 4 pages. 2010.
- [20] ZIMMERMANN, T.; PREMRAJ, R.; Zeller, A. "Predicting Defects for Eclipse," Predictor Models in Software Engineering, 2007. PROMISE'07: ICSE Workshops 2007. International Workshop, pp.9, 20-26 May 2007.
- [21] MOSER, R.; PEDRYCZ, W.; SUCCI, G. "A comparative analysis of the efficiency of change metrics and static code attributes for defect prediction," *Software Engineering, 2008. ICSE '08. ACM/IEEE 30th International Conference*, pp.181-190, 10-18 May 2008.
- [22] RAJA, U.; HALE, D. P.; HALE, J. E. *Modeling software evolution defects: a time series approach*. J. Softw. Maint. Evol., 21(1):49–71, 2009.
- [23] VILLA, Luis. "Large Free Software Projects and Bugzilla", Ximian, Inc., Canada, July 23th–26th, 2003.

- [24] FISCHER, M.; PINZGER, M.; GALL, H. "Analyzing and Relating Bug Report Data for Feature", Technical University of Vienna, Information Systems Institute, June 16, 2003.
- [25] KALEKAR, Prajakta S. Time series Forecasting using Holt-Winters Exponential Smoothing, Kanwal Rekhi School of Information Technology, December 6, 2004.
- [26] FOMBY, Thomas B. "Exponential Smoothing Models", Department of Economics, Southern Methodist University, Dallas, TX, June 2008.
- [27] BORNIA, Antonio Cezar; BARBETTA, Pedro Alberto; REIS, Marcelo Menezes. Estatística para Cursos de Engenharia e Informática. 2.ed. Rio de Janeiro: Atlas, 2008.
- [28] GONÇALVES, Esmeralda; LOPES, Nazaré Mendes. Séries Temporais – Modelações Lineares e Não Lineares, 2.ed. Sociedade Portuguesa de Estatística, INE, I.P., 2008.
- [29] KENDALL, Maurice; STUART, Alan. The Advanced theory of statistics – Design and Analysis, and Time-Series. Vol. 3. 3ª ed., Griffin, 1976.
- [30] D'AMBROS, Marco; LANZA, Michele. "BugCrawler: Visualizing Evolving Software Systems," *Software Maintenance and Reengineering, 2007. CSMR '07. 11th European Conference on*, pp.333-334, 21-23 March 2007.
- [31] D'AMBROS, Marco; LANZA, Michele. Visual software evolution reconstruction. *J. Softw. Maint. Evol.: Res. Pract.*, 21: 217–232, 2009.
- [32] CUBRANIC, D. et al. "Hipikat: a project memory for software development," *Software Engineering, IEEE Transactions on* , vol.31, no.6, pp. 446- 465, June 2005.
- [33] GIGER, Emanuel; PINZGER, Martin; GALL, Harald. "Predicting the fix time of bugs." *Proceedings of the 2nd International Workshop on Recommendation Systems for Software Engineering*. ACM, 2010.
- [34] LEHMAN, M. M.; RAMIL, Juan F. Rules and Tools for Software Evolution Planning and Management, *Ann. Softw. Eng.* 11, 15-44, November 2001.
- [35] HALLORAN, T.; SCHERLIS, W. "High Quality and Open Source Software Practices", In: Workshop on Open Source Software Engineering, pp. 19-25, 2002.
- [36] SOLINGEN, R. V.; BERGHOUT, E. The Goal/Question/Metric Method: a Practical Guide for Quality Improvement of Software Development, McGraw Hill, England, 1999.
- [37] ESTEVES, J.; PASTOR, J.; CASANOVAS, J. "Measuring Sustained Management Support in ERP Implementation Projects: a GQM Approach". 8th Americas Conference on Information Systems (AMCIS), Dallas, USA, 2002, pp. 1381-1389.

- [38] BASILI, Victor R.; CALDIERA, Gianluigi; ROMBACH, H. Dieter. "The Goal Question Metric Approach", Institute for Advanced Computer Studies, University Of Maryland, Maryland, FB Informatik, Universität Kaiserslautern, Germany.
- [39] BOEHM, Barry. Software engineering economics, Prentice Hall PTR, 1981.
- [40] EVELEENS, J. Laurenz; VERHOEF, Chris. Quantifying IT forecast quality, Science of Computer Programming, Volume 74, Issues 11–12, November 2009, Pages 934-988, ISSN 0167-6423.
- [41] EVELEENS, J. Laurenz; VERHOEF, Chris. "The rise and fall of the chaos report figures." IEEE software vol. 27 n. 1 pp 30-36, 2010.

APÊNDICES

Apêndice I – Exemplos de ficheiros manipulados na criação e execução do 1º script SPSS
(excertos)

MonthYear	ChangeRequests	YEAR	MONTH	DATE
Oct 2001	223	2001	10	OCT 2001
Nov 2001	140	2001	11	NOV 2001
Dec 2001	91	2001	12	DEC 2001
Jan 2002	183	2002	1	JAN 2002

Figura 1 – Ficheiro SAV inicial

```

#3 Execution..... Started @ 22:02:04
#5 System: Eclipse30F @ 22:02:04
#6 File W:\SAV\Eclipse30F\ChangeRequests_1.SPS open
#7 Processing Trial Start: syear 2001, month 10 through Trial End: syear 2002, month 10 Started @ 22:03:04
#8 Processing Prediction Start: syear 2002, month 11 through Prediction End: syear 2003, month 10 Started @ 22:03:04
#9 Model : EXPERTMODELER @ 22:03:04
#7 EXPERTMODELER..... Started @ 22:03:04
#8 EXPERTMODELER..... Ended @ 22:03:04
#9 Processing Trial Start: syear 2001, month 10 through Trial End: syear 2002, month 10 Ended @ 22:03:04
#10 Processing Prediction Start: syear 2002, month 11 through Prediction End: syear 2003, month 10 Ended @ 22:03:04
    
```

Figura 2 – Ficheiro LOG

```

GET
FILE='M:\FCT\SPSS\SAV\Eclipse30F\ChangeRequests.sav'.
DATASET NAME DataSet1 WINDOW=FRONT.
FILTER OFF.
USE year 2001 month 10 thru year 2002 month 10.
EXECUTE.
PREDICT THRU YEAR 2003 MONTH 10.
Time Series Modeler
TSMODEL
/MODELSUMMARY PRINT=[MODEL FIT RESIDUAL PREDICTION]
/MODELSTATISTICS DISPLAY=YES MODEL FIT=[R SQUARE RS SQUARE RMSE MAPE MAE MAXAPE MAXAE NORMSIC]
/MODELDETAILS PRINT=[PARAMETERS RESIDUAL PREDICTION FORECAST] PLOT=[RESIDUAL PREDICTION]
/SERIESPLOT OBSERVED FORECAST FIT FORECASTCI FITCI
/OUTPUTFILTER DISPLAY=[IDENTIFY=5] MODEL FIT=[CHRONO]
/SAVE PREDICTED(Predicted) LCL(LCL) UCL(UCL) RESIDUAL(Residual)
/AUXILIARY CILEVEL=95 MAXACFLAGS=24
/MISSING USERMISSING=[EXCLUDE]
/MODEL DEPENDENT=CHANGEREQUESTS INDEPENDENT=MonthYear
/OUTFILE='M:\FCT\SPSS\SAV\Eclipse30F\ChangeRequests_E_1.smf'
/PREFIX=Model
/EXPERTMODELER TYPE=[ARMA ESMOOTH] TRYSEASONAL=YES
/AUTOOUTLIER_DETECT=OFF
    
```

Figura 3 – Ficheiro SPS

Time Series Modeler

[DataSet1] M:\FCT\SPSS\SAV\Eclipse30F\ChangeRequests.sav

Model Description		Model Type									
Model ID	Change Requests	Model_1	Winters' Additive								
Model Summary											
Model Fit											
Fit Statistic	Mean	SD	Minimum	Maximum	5	10	25	50	75	90	95
Observed R-squared											
RMSE	4.039E-11	4.039E-11	4.039E-11	4.039E-11	4.039E-11	4.039E-11	4.039E-11	4.039E-11	4.039E-11	4.039E-11	4.039E-11
MAPE	8.944E-12	8.944E-12	8.944E-12	8.944E-12	8.944E-12	8.944E-12	8.944E-12	8.944E-12	8.944E-12	8.944E-12	8.944E-12
MAE	1.742E-11	1.742E-11	1.742E-11	1.742E-11	1.742E-11	1.742E-11	1.742E-11	1.742E-11	1.742E-11	1.742E-11	1.742E-11
MAAD	1.367E-10	1.367E-10	1.367E-10	1.367E-10	1.367E-10	1.367E-10	1.367E-10	1.367E-10	1.367E-10	1.367E-10	1.367E-10
Normalized RMSE	-47.003	-47.003	-47.003	-47.003	-47.003	-47.003	-47.003	-47.003	-47.003	-47.003	-47.003

Figura 4 – Ficheiro SPV

Model Description			Model Type
Model ID	Change Requests	Model_1	Winters' Additive

Figura 5 – Ficheiro TXT

Apêndice II – Exemplos de ficheiros manipulados na criação e execução do 2º script SPSS
(excertos)

```

<EstimationInfo periodStartIndex="0" periodLength="12" degreesOfFreedom="9">
  <Statistic type="errVariance">2.0716591019407e-021</Statistic>
  <Statistic type="meanSqrErr">2.07485475569601e-021</Statistic>
  <Statistic type="rootMeanSqrErr">4.55505736044675e-011</Statistic>
  <Statistic type="meanAbsPctErr">8.5443556215549e-013</Statistic>
  <Statistic type="meanAbsErr">1.14160532878789e-011</Statistic>
  <Statistic type="maxAbsPctErr">1.02131224918821e-011</Statistic>
  <Statistic type="maxAbsErr">1.36651578941382e-010</Statistic>
  <Statistic type="rSqr">1</Statistic>
  <Statistic type="normBayesIC">-47.0031691364502</Statistic>
  <Statistic type="logLikelihood">270.454453044879</Statistic>
  <Statistic type="akaikeIC">-534.908906089757</Statistic>
  <Statistic type="bayesIC">-533.454186140394</Statistic>
</EstimationInfo>
</ExpSmoothingModel>
    
```

Figura 6 – Ficheiro XML

```

System,Model,StartTrialYear,StartTrialMonth,EndTrialYear,EndTrialMonth,StartPredictionYear,StartPredictionMonth,EndPredictionYear,EndPredictionMonth,vAf
EclipseJDT,Exponential Smoothing - Winters' Additive,2001,10,2002,10,2002,11,2003,10,0,0,0,0,0,2.0716591019407E-21,2.07485475569601E-21,4.55505736044675E
EclipseJDT,Exponential Smoothing - Winters' Additive,2001,10,2002,11,2002,12,2003,11,0,0,0,0,0,0,0.026875501771888,0.0268755017718827,0.163937493490302,0.00
EclipseJDT,Exponential Smoothing - Winters' Additive,2001,10,2002,12,2003,1,2003,12,0,0,0,0,0,0,37.4369632469353,37.4369632469353,6.11857526283164,0.271546
    
```

Figura 7 – Ficheiro CSV

System	Model	StartTrialYear	StartTrialMonth	EndTrialYear	EndTrialMonth	maxAbsPctErr	maxAbsErr	rSqr	normBayesIC	logLikelihood
EclipseJDT	Exponential Smoothing - Winters' Additive	2001	10	2002	10	1.02E-11	1.17E-10	1	-47.00316914	270.454453
EclipseJDT	Exponential Smoothing - Winters' Additive	2001	10	2002	11	0.040003025	0.42283197	0.999999964	-3.024628732	6.76677581
EclipseJDT	Exponential Smoothing - Winters' Additive	2001	10	2002	12	1.856773524	13.1645243	0.999944484	4.188170823	-43.51563484
EclipseJDT	Exponential Smoothing - Winters' Additive	2001	10	2003	1	16.28447823	273.2535447	0.976883271	10.11043713	-91.37670452
EclipseJDT	Exponential Smoothing - Winters' Additive	2001	10	2003	2	17.52130639	307.6778334	0.961116495	10.86743323	-102.2224843
EclipseJDT	Exponential Smoothing - Winters' Additive	2001	10	2003	3	16.58115294	269.2779238	0.960875286	10.6067419	-108.379115
EclipseJDT	Exponential Smoothing - Simple Seasonal	2001	10	2003	4	68.6289609	866.2369789	0.765167134	12.12150384	-130.6840091
EclipseJDT	Exponential Smoothing - Simple Seasonal	2001	10	2003	5	64.63137925	1213.408082	0.518558994	12.81182953	-144.6693252

Figura 8 – Ficheiro XLSX

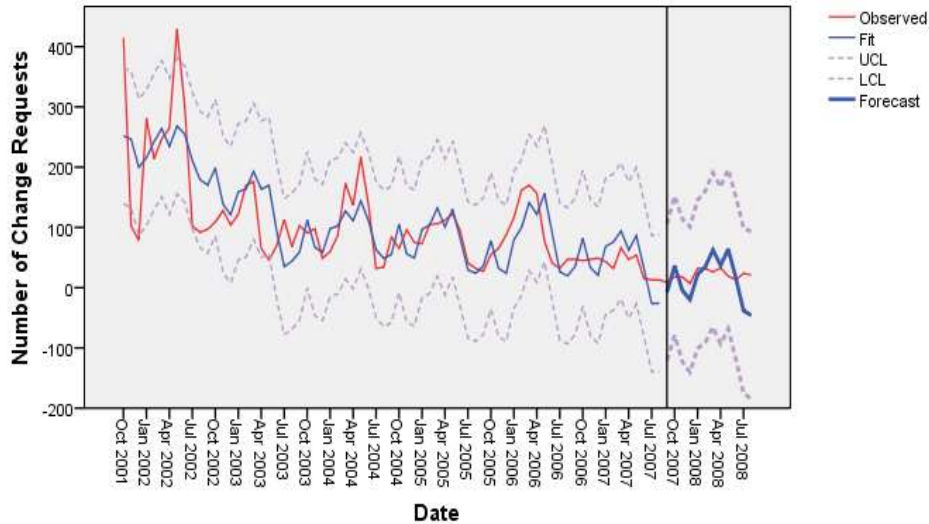
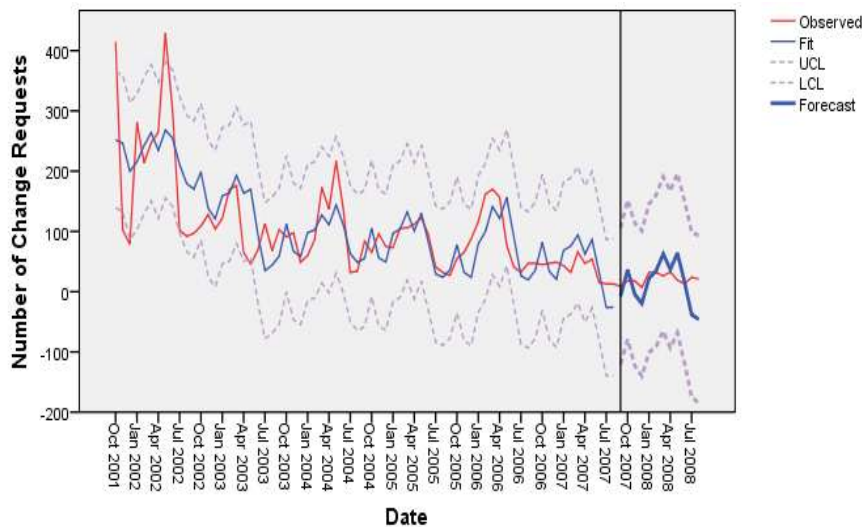
System	Model	StartTrialYear	StartTrialMonth	EndTrialYear	EndTrialMonth	maxAbsPctErr	maxAbsErr	rSqr	normBayesIC
EclipseJDT	Exponential Smoothing - Winters' Additive	2001	10	2002	10	1E-11	1.4E-10	1.0000000000	-47.00316913645
EclipseJDT	Exponential Smoothing - Winters' Additive	2001	10	2002	11	0.04000302463	42283197033	.99999996353	-3.02462873194
EclipseJDT	Exponential Smoothing - Winters' Additive	2001	10	2002	12	1.85677352614	13.16452430033	.99994448423	4.18817082338
EclipseJDT	Exponential Smoothing - Winters' Additive	2001	10	2003	1	16.28447823095	273.25354471531	.97688327123	10.11043712819

Figura 9 – Ficheiro SAV final

Apêndice III - Figuras: Software EclipsePlatform – Modelos de Previsão

Período de Treino: 10/2001 – 08/2007

Período de Previsão: 08/2007 – 08/2008

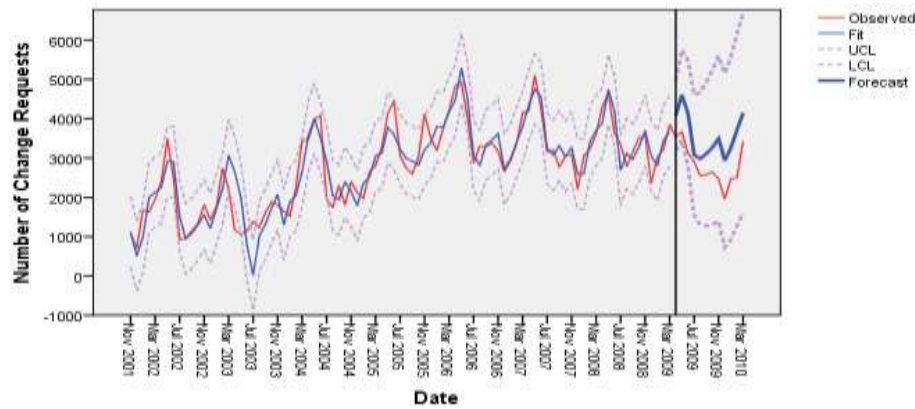
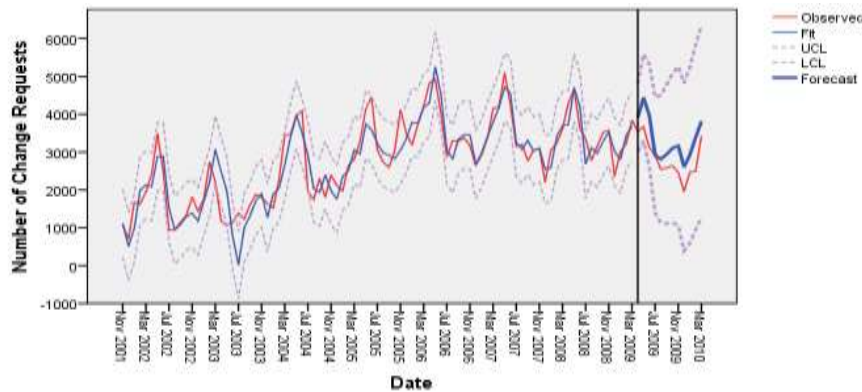
**EXWA - Exponential Smoothing - Winters' Additive****EXWM - Exponential Smoothing – Winter's Multiplicative**

Nas figuras referentes aos modelos mencionados podemos verificar que o seu comportamento é idêntico. Os valores observados acerca do número de pedidos de evolução de *software* encontram-se dentro dos limites do intervalo de confiança de 95%, à exceção dos meses de outubro de 2001 e de maio e junho de 2002. Estes últimos meses referidos precedem geralmente o lançamento de uma nova versão. Os valores de previsão, à exceção dos meses citados, encontram-se dentro dos limites credíveis e muito próximos dos valores observados. O comportamento dos diferentes valores exibidos ao longo do tempo está associado ainda uma tendência linear monótona, pois diminui e aumenta gradualmente, apresentando ainda sazonalidade.

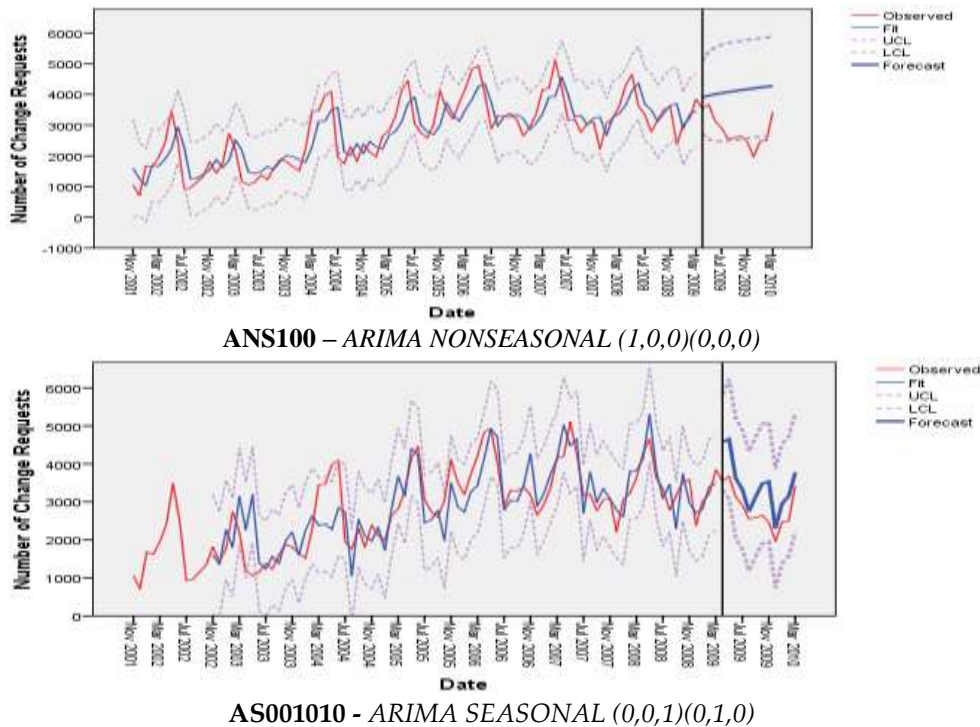
Apêndice IV - Figuras: Software Eclipse – Modelos de Previsão

Período de Treino: 10/2001 – 03/2009

Período de Previsão: 03/2009 – 03/2010

**EXWA - Exponential Smoothing - Winters' Additive****EXWM - Exponential Smoothing – Winter's Multiplicative**

Nas figuras pertencentes aos modelos acima indicados podemos averiguar que o seu comportamento é semelhante. Os valores observados acerca do número de pedidos de evolução de *software* encontram-se dentro dos limites do intervalo de confiança de 95%. Os números de pedidos de evolução de *software* mais elevados verificam-se geralmente nos meses de maio e de junho, pois são meses que antecedem o lançamento de uma nova versão. Os valores de previsão encontram-se dentro dos limites credíveis e muito próximos dos valores observados. O comportamento dos diferentes valores exibidos ao longo do tempo está associado a uma tendência crescente de novembro de 2001 até junho de 2006, mantendo um padrão mais constante a partir dessa data. Apresenta ainda sazonalidade. Em resumo, quando comparados os dados observados (linha vermelha) com o modelo de ajuste (linha azul fina) depreende-se que o modelo se encontra ajustado o suficiente para efetuar previsões sobre períodos futuros. Comparando os valores previstos (linha azul espessa) com os valores dos dados observados, podemos concluir que o modelo realiza previsões dentro de um nível de confiança admissível, apesar de se confirmar uma ligeira sobrestimação do número de pedidos de evolução de *software* na previsão, no período de previsão.

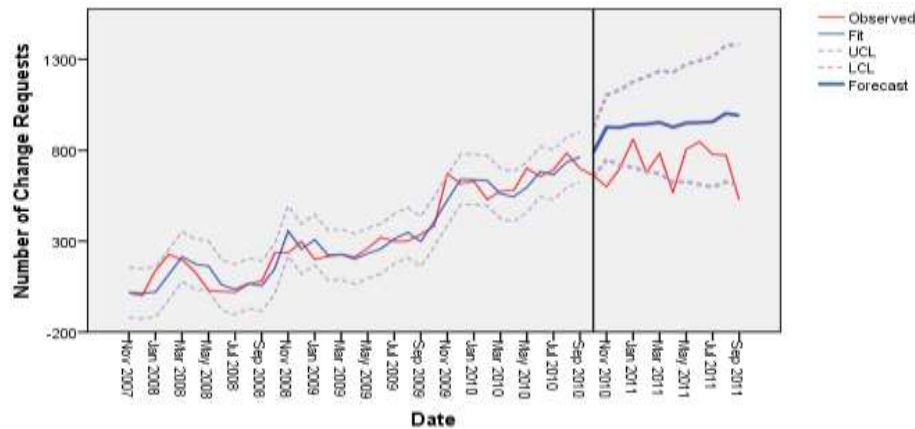
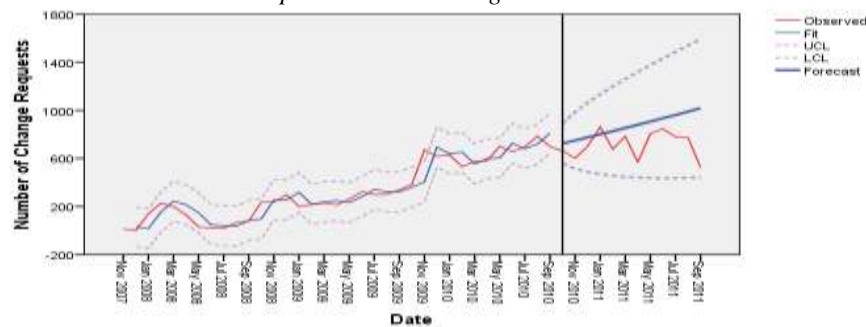


Nas figuras relativas aos modelos apresentados podemos observar que o comportamento dos diferentes valores exibidos ao longo do tempo está associado a uma tendência crescente de novembro de 2001 até junho de 2006, mantendo-se mais constante a partir dessa data. O modelo ANS100 apresenta um parâmetro autorregressivo. A sazonalidade está presente apenas no modelo AS001010, onde a série foi diferenciada uma vez e os valores previstos representam uma combinação linear de um número finito de ruídos (1 parâmetro de Média Móvel). Os valores observados acerca do número de pedidos de evolução de *software* encontram-se dentro dos limites do intervalo de confiança, à exceção dos meses de previsão entre novembro de 2009 e março de 2010 no modelo ANS100. Os números de pedidos de evolução de *software* mais elevados verificam-se geralmente nos meses de maio e de junho, pois são meses que antecedem o lançamento de uma nova versão. Os valores de previsão encontram-se dentro dos limites credíveis e muito próximos dos valores observados, exceto no período de previsão do modelo ANS100, pois enquanto os valores previstos apresentam uma tendência crescente, os valores observados apresentam uma tendência decrescente de junho de 2009 até janeiro de 2010, e depois uma tendência crescente até março de 2010. Em síntese, quando comparados os dados observados (linha vermelha) com o modelo de ajuste (linha azul fina) depreende-se que o modelo se encontra ajustado o suficiente para efetuar previsões sobre períodos futuros. Comparando os valores previstos (linha azul espessa) com os valores dos dados observados, podemos concluir que o modelo realiza previsões dentro de um nível de confiança admissível, apesar de se confirmar uma ligeira sobrestimação do número de pedidos de evolução de *software* na previsão, no período de previsão.

Apêndice V - Figuras: Software Android – Modelos de Previsão

Período de Treino: 11/2007 – 09/2010

Período de Previsão: 09/2010 – 09/2011

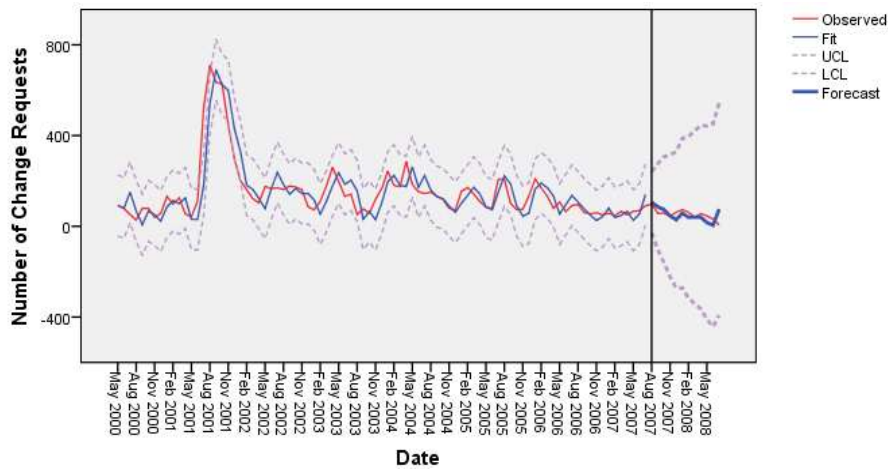
**EXWA - Exponential Smoothing - Winters' Additive****ANS100 – ARIMA NONSEASONAL (1,0,0)(0,0,0)**

Nas figuras respeitantes aos modelos exibidos podemos notar que o comportamento dos diferentes valores apresentados ao longo do tempo está associado geralmente a uma tendência crescente desde novembro de 2007. Os valores observados acerca do número de pedidos de evolução de *software* encontram-se dentro dos limites do intervalo de confiança de 95%, à exceção dos meses de previsão dos meses de setembro de 2010, abril de 2011 e de setembro de 2011 no modelo EXWA. Este modelo indica que existe uma tendência, neste caso crescente, e sazonalidade na série. Os valores de previsão encontram-se dentro dos limites credíveis e muito próximos dos valores observados, exceto no período de previsão do modelo ANS100, pois enquanto os valores previstos apresentam uma tendência crescente, os valores observados apresentam uma tendência decrescente de junho de 2009 até janeiro de 2010, e depois uma tendência crescente até março de 2010. O modelo ANS100 não exhibe sazonalidade e apresenta um parâmetro autorregressivo. Em suma, quando comparados os dados observados (linha vermelha) com o modelo de ajuste (linha azul fina) depreende-se que o modelo se encontra ajustado o suficiente para efetuar previsões sobre períodos futuros. Observando os valores previstos (linha azul espessa) e confrontando-os com os valores dos dados observados, podemos concluir que o modelo faz previsões dentro de um nível de confiança aceitável, apesar de se constatar uma ligeira sobrestimação do número de pedidos de evolução de *software* na previsão, no período de previsão.

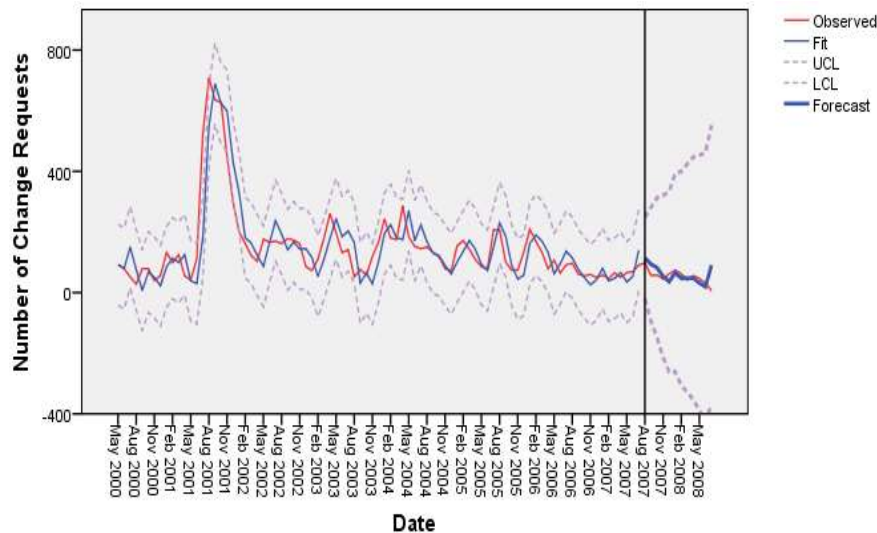
Apêndice VI - Figuras: *Software Gnome Evolution* – Modelos de Previsão

Período de Treino: 05/2000 – 07/2007

Período de Previsão: 07/2007 – 07/2008

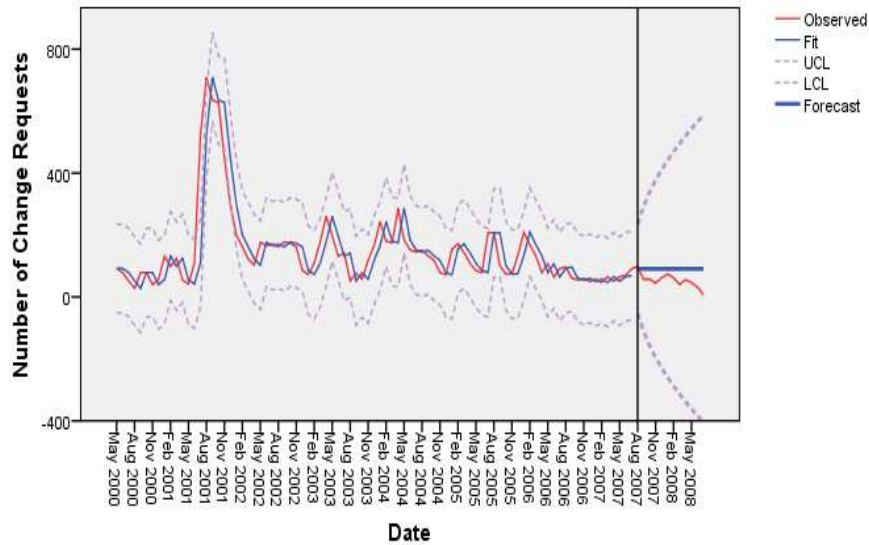


EXWA - Exponential Smoothing - Winters' Additive

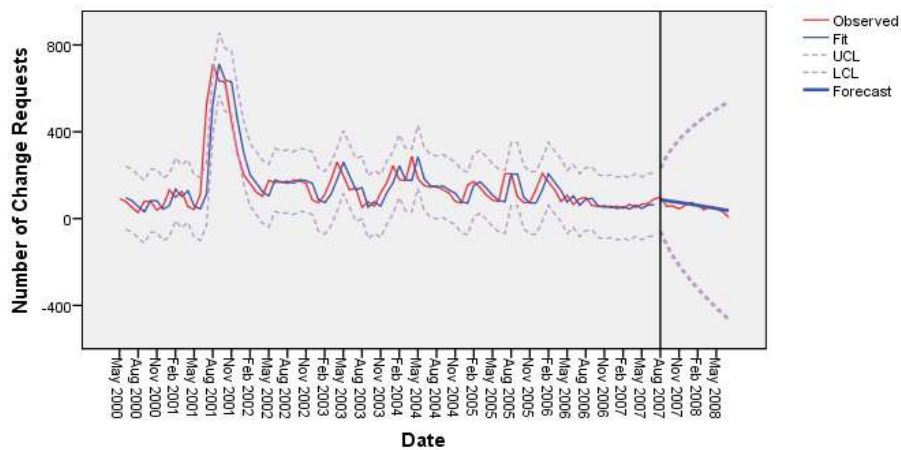


EXWM - Exponential Smoothing – Winter's Multiplicative

Nas figuras alusivas aos modelos apresentados podemos apurar que o seu comportamento é semelhante. Os valores observados acerca do número de pedidos de evolução de *software* encontram-se dentro dos limites do intervalo de confiança de 95%. Os valores de previsão encontram-se dentro dos limites credíveis e muito próximos dos valores observados. O comportamento dos diferentes valores exibidos ao longo do tempo está associado a uma tendência linear monótona, à exceção dos meses entre agosto de 2001 e fevereiro de 2002. Os modelos mostram ainda a existência de sazonalidade. Em resumo, quando comparados os dados observados (linha vermelha) com o modelo de ajuste (linha azul fina) deduz-se que o modelo se encontra ajustado o suficiente para efetuar previsões sobre períodos futuros. Comparando os valores previstos (linha azul espessa) com os valores dos dados observados, podemos concluir que o modelo realiza previsões dentro de um nível de confiança assumptível.



EXS - Exponential Smoothing - Simple



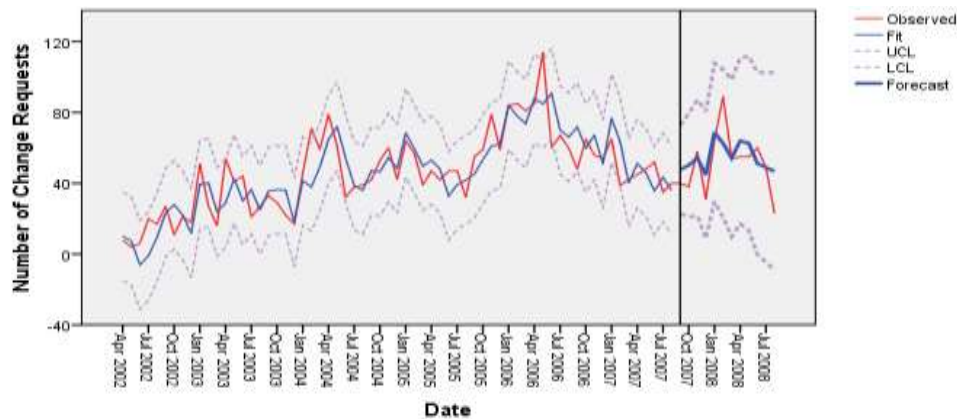
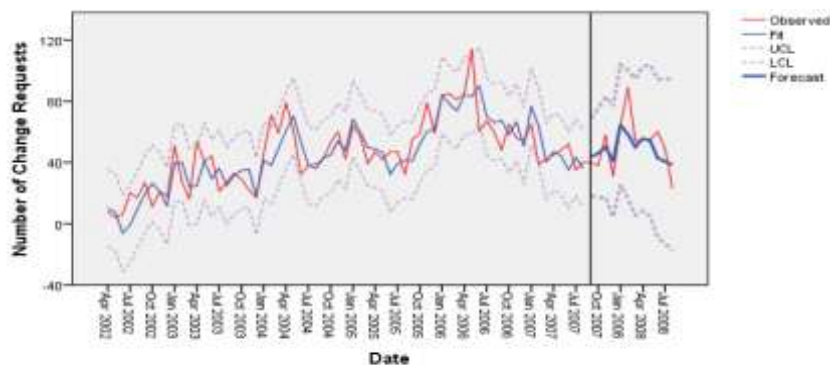
ANS010 – ARIMA NONSEASONAL (0,1,0)(0,0,0)

Nas figuras relativas aos modelos exibidos podemos verificar que o seu comportamento é semelhante. Ambos os modelos não apresentam nem sazonalidade nem tendência. O modelo ANS010 não exibe sazonalidade e indica que houve uma diferenciação da série. Os valores observados acerca do número de pedidos de evolução de *software* encontram-se dentro dos limites do intervalo de confiança de 95%. Os valores de previsão encontram-se dentro dos limites credíveis e muito próximos dos valores observados. O comportamento dos diferentes valores exibidos ao longo do tempo está associado a um comportamento monótono, à exceção dos meses entre agosto de 2001 e fevereiro de 2002. Em resumo, quando comparados os dados observados (linha vermelha) com o modelo de ajuste (linha azul fina) infere-se que o modelo se encontra ajustado razoavelmente para efetivar previsões sobre períodos futuros. Comparando os valores previstos (linha azul espessa) com os valores dos dados observados, podemos concluir que o modelo concretiza previsões dentro de um nível de confiança aceitável.

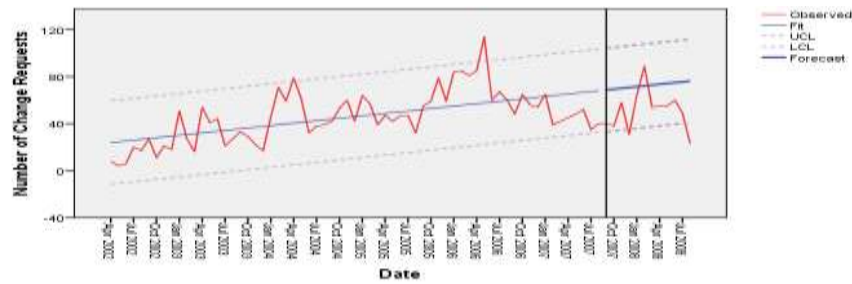
Apêndice VII - Figuras: Software *Gnome GStreamer* – Modelos de Previsão

Período de Treino: 04/2002 – 08/2007

Período de Previsão: 08/2007 – 08/2008

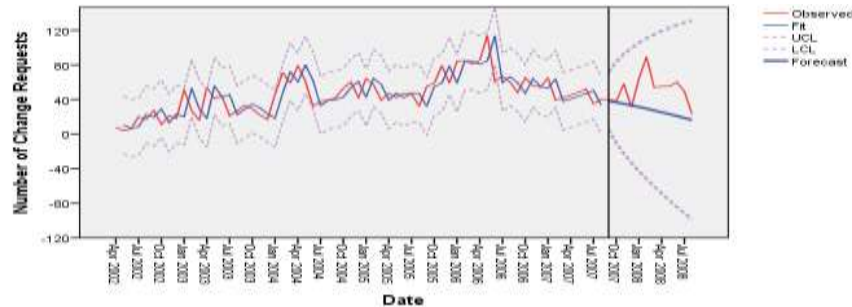
**EXWA - Exponential Smoothing - Winters' Additive****EXWM - Exponential Smoothing – Winter's Multiplicative**

Nas figuras referentes aos modelos exibidos podemos verificar que o seu comportamento é análogo. Os dois modelos apresentam tendência crescente e sazonalidade. Os valores observados acerca do número de pedidos de evolução de *software* encontram-se dentro dos limites do intervalo de confiança de 95%. Os valores de previsão encontram-se dentro dos limites credíveis e muito próximos dos valores observados. O comportamento dos diferentes valores exibidos ao longo do tempo está associado a uma tendência crescente de abril de 2002 até junho de 2006, e a uma tendência decrescente a partir dessa data. Em síntese, quando comparados os dados observados (linha vermelha) com o modelo de ajuste (linha azul fina) percebe-se que o modelo se encontra ajustado o suficiente para fazer previsões sobre períodos futuros. Comparando os valores previstos (linha azul espessa) com os valores dos dados observados, podemos concluir que o modelo realiza previsões dentro de um nível de confiança aceitável, apesar de se verificar uma ligeira subestimação do número de pedidos de evolução de *software* na previsão, no período de previsão.



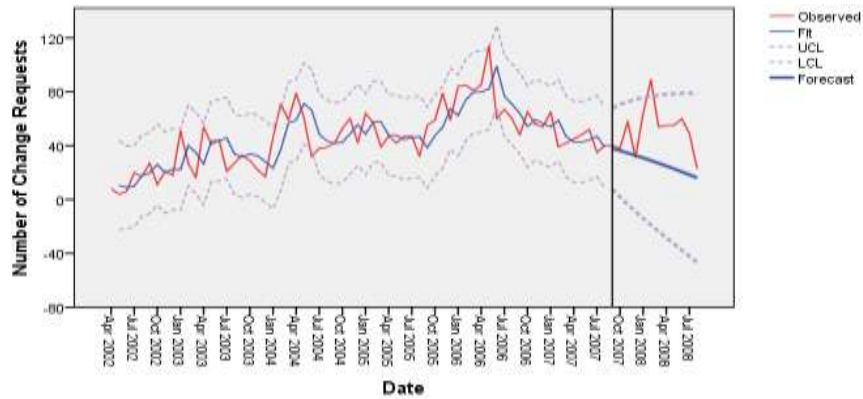
ANS000 - ARIMA NONSEASONAL (0,0,0)(0,0,0)

No gráfico relativo ao modelo apresentado podemos verificar que os valores observados acerca do número de pedidos de evolução de *software* encontram-se dentro dos limites do intervalo de confiança de 95%, à exceção dos meses de maio e de junho de 2006. Este modelo não exhibe sazonalidade. O comportamento dos diferentes valores exibidos ao longo do tempo está associado a uma tendência crescente de abril de 2002 até junho de 2006, e a uma tendência decrescente a partir dessa data. Em síntese, quando comparados os dados observados (linha vermelha) com o modelo de ajuste (linha azul fina) percebe-se que o modelo não se encontra ajustado o suficiente para fazer previsões sobre períodos futuros. Comparando os valores previstos (linha azul espessa) com os valores dos dados observados, podemos aferir que o modelo realiza previsões dentro de um nível de confiança aceitável, apesar de se verificar uma ligeira sobrestimação do número de pedidos de evolução de *software* na previsão, no período de previsão.



ANS010 - ARIMA NONSEASONAL (0,1,0)(0,0,0)

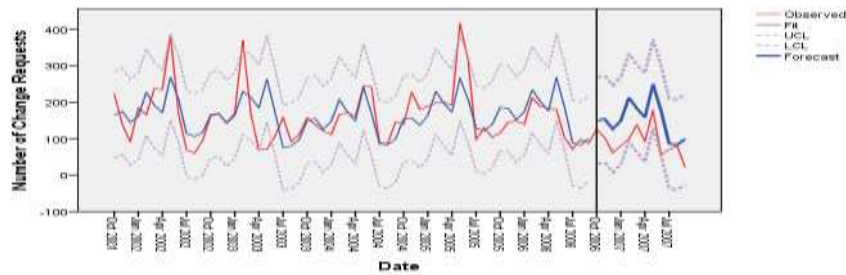
No gráfico respeitante ao modelo exposto podemos atestar que os valores observados acerca do número de pedidos de evolução de *software* encontram-se dentro dos limites do intervalo de confiança de 95%. Este modelo indica que não existe sazonalidade e que as séries foram diferenciadas uma vez. O comportamento dos diferentes valores exibidos ao longo do tempo está associado a uma tendência crescente de abril de 2002 até junho de 2006, e a uma tendência decrescente a partir dessa data. Em resumo, quando comparados os dados observados (linha vermelha) com o modelo de ajuste (linha azul fina) percebe-se que o modelo se encontra ajustado o suficiente para fazer previsões sobre períodos futuros. Comparando os valores previstos (linha azul espessa) com os valores dos dados observados, podemos aferir que o modelo realiza previsões dentro de um nível de confiança aceitável, apesar de se verificar uma suave subestimação do número de pedidos de evolução de *software* na previsão, no período de previsão.



ANS011 - ARIMA NONSEASONAL (0,1,1)(0,0,0)

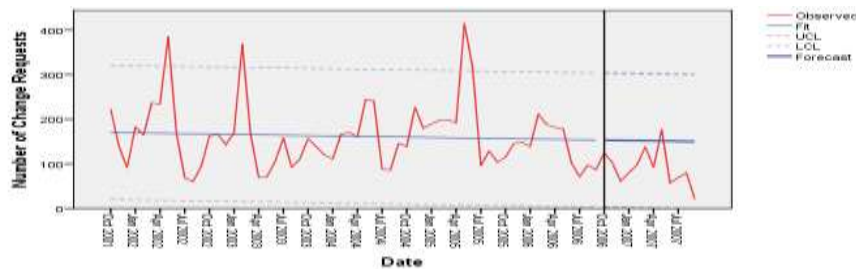
No gráfico pertencente ao modelo apresentado podemos afirmar que os valores observados acerca do número de pedidos de evolução de *software* encontram-se dentro dos limites do intervalo de confiança de 95%. O modelo ANS011 está associado a um parâmetro de média móvel e a uma diferenciação das séries. O comportamento dos diferentes valores exibidos ao longo do tempo está associado a uma tendência crescente de abril de 2002 até junho de 2006, e a uma tendência decrescente a partir dessa data. Em síntese, quando comparados os dados observados (linha vermelha) com o modelo de ajuste (linha azul fina) percebe-se que o modelo se encontra ajustado o suficiente para fazer previsões sobre períodos futuros. Comparando os valores previstos (linha azul espessa) com os valores dos dados observados, podemos aferir que o modelo realiza previsões dentro de um nível de confiança admissível, apesar de se verificar uma suave subestimação do número de pedidos de evolução de *software* na previsão, no período de previsão.

Apêndice VIII - Figuras: Software Mozilla Core – Modelos de Previsão
 Período de Treino: 10/2001 – 09/2006 Período de Previsão: 09/2006 – 09/2007



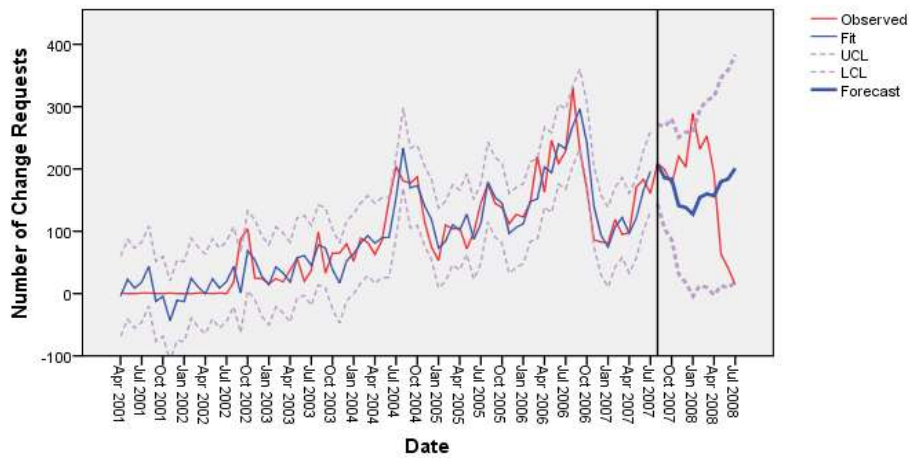
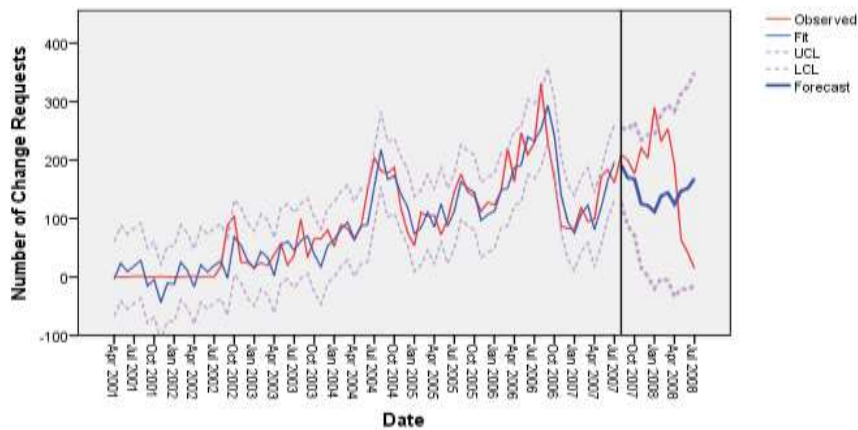
EXSS - Exponential Smoothing – Simple Seasonal

No gráfico pertencente ao modelo acima referido podemos confirmar que os valores observados acerca do número de pedidos de evolução de *software* encontram-se dentro dos limites do intervalo de confiança de 95%. Este modelo comprova a sazonalidade existente na série. O comportamento dos diferentes valores exibidos ao longo do tempo está associado a uma tendência monótona (diminuindo e aumentando de forma gradual). Em sumário, quando comparados os dados observados (linha vermelha) com o modelo de ajuste (linha azul fina) compreende-se que o modelo se encontra ajustado o suficiente para fazer previsões sobre períodos futuros. Comparando os valores previstos (linha azul espessa) com os valores dos dados observados, podemos avaliar que o modelo realiza previsões dentro de um nível de confiança aceitável, apesar de se verificar uma suave sobrestimação do número de pedidos de evolução de *software* na previsão, no período de previsão.

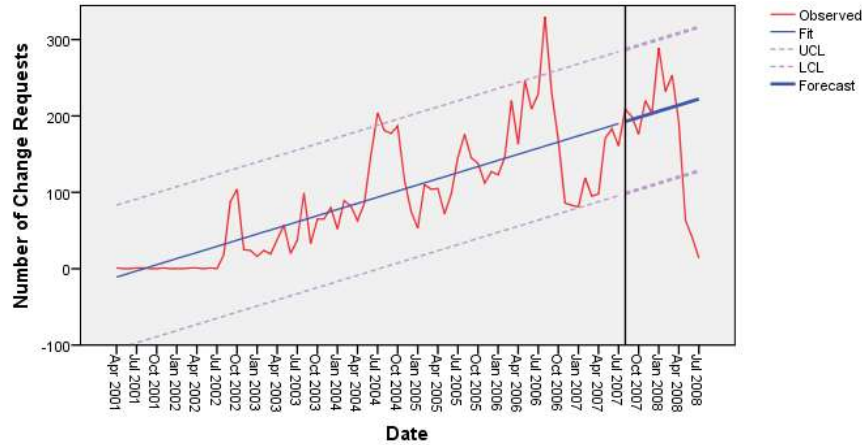


ANS000 - ARIMA NONSEASONAL (0,0,0)(0,0,0)

No gráfico relativo ao modelo apresentado podemos verificar que os valores observados acerca do número de pedidos de evolução de *software* encontram-se dentro dos limites do intervalo de confiança de 95%, à exceção dos meses de junho de 2002, março de 2003 e junho de 2005. Este modelo não exhibe sazonalidade. Em suma, quando comparados os dados observados (linha vermelha) com o modelo de ajuste (linha azul fina) percebe-se que o modelo não se encontra ajustado o suficiente para fazer previsões sobre períodos futuros. Comparando os valores previstos (linha azul espessa) com os valores dos dados observados, podemos aferir que o modelo realiza previsões dentro de um nível de confiança aceitável, apesar de se verificar uma ligeira sobrestimação do número de pedidos de evolução de *software* na previsão, no período de previsão.

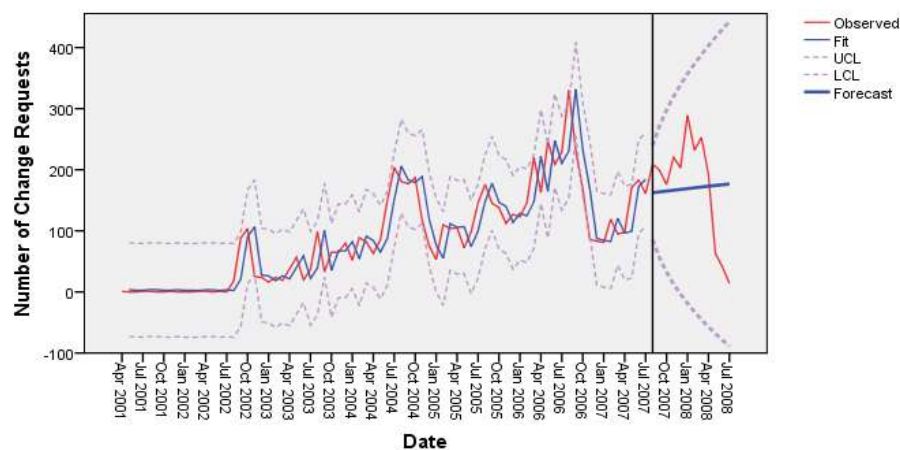
Apêndice IX - Figuras: *Software Mozilla Firefox* – Modelos de PrevisãoPeríodo de Treino: **04/2001 – 07/2007**Período de Previsão: **07/2007 – 07/2008****EXWA - Exponential Smoothing - Winters' Additive****EXSS - Exponential Smoothing – Simple Seasonal**

Nas figuras concernentes aos modelos expostos podemos assinalar que o comportamento dos diferentes valores apresentados ao longo do tempo está associado geralmente a uma tendência crescente a partir de julho de 2002 até agosto de 2006. Estes modelos exibem uma tendência, geralmente crescente, e sazonalidade na série. Os valores observados acerca do número de pedidos de evolução de *software* encontram-se dentro dos limites do intervalo de confiança. Em síntese, quando comparados os dados observados (linha vermelha) com o modelo de ajuste (linha azul fina) percebe-se que o modelo se encontra ajustado o suficiente para efetuar previsões sobre períodos futuros. Observando os valores previstos (linha azul espessa) e comparando-os com os valores dos dados observados, podemos concluir que o modelo faz previsões dentro de um nível de confiança aceitável, apesar de se verificar uma subestimação do número de pedidos de evolução de *software* na previsão, no período de previsão.



ANS000 - ARIMA NONSEASONAL (0,0,0)(0,0,0)

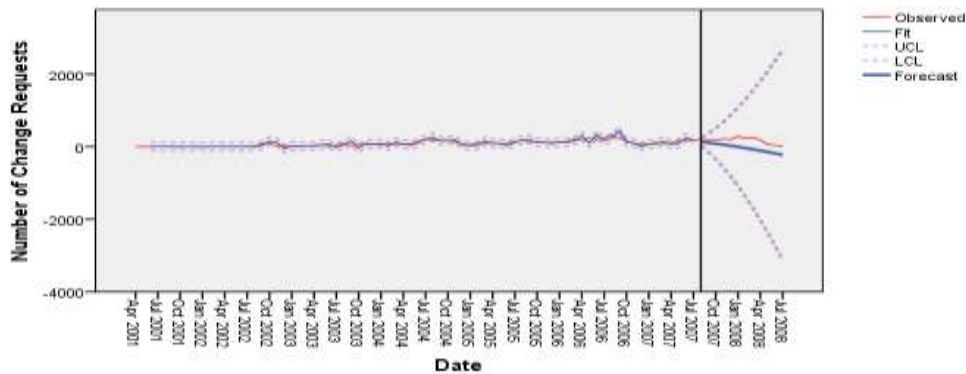
No gráfico relativo ao modelo apresentado podemos verificar que os valores observados acerca do número de pedidos de evolução de *software* encontram-se dentro dos limites do intervalo de confiança, à exceção dos meses de agosto de 2004, setembro de 2006 e de junho e julho de 2008. Este modelo não exhibe sazonalidade. O comportamento dos diferentes valores exibidos ao longo do tempo está geralmente associado a uma tendência crescente. Em síntese, quando comparados os dados observados (linha vermelha) com o modelo de ajuste (linha azul fina) percebe-se que o modelo não se encontra ajustado o suficiente para fazer previsões sobre períodos futuros. Comparando os valores previstos (linha azul espessa) com os valores dos dados observados, podemos aferir que o modelo realiza previsões dentro de um nível de confiança aceitável.



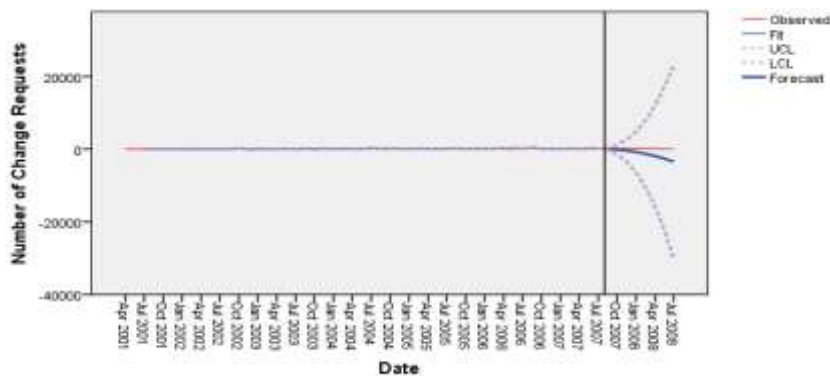
ANS010 - ARIMA NONSEASONAL (0,1,0)(0,0,0)

Nas figuras concernentes aos modelos expostos podemos assinalar que o comportamento dos diferentes valores apresentados ao longo do tempo está associado geralmente a uma tendência crescente a partir de julho de 2002 até agosto de 2006. O modelo ANS010 não exhibe sazonalidade e apresenta uma diferenciação da série. Os valores observados acerca do número de pedidos de

evolução de *software* encontram-se dentro dos limites do intervalo de confiança. Em suma, quando comparados os dados observados (linha vermelha) com o modelo de ajuste (linha azul fina) compreende-se que o modelo se encontra ajustado o suficiente para efetuar previsões sobre períodos futuros. Observando os valores previstos (linha azul espessa) e comparando-os com os valores dos dados observados, podemos concluir que o modelo faz previsões dentro de um nível de confiança.



ANS020 - ARIMA NONSEASONAL (0,0,0)(0,0,0)



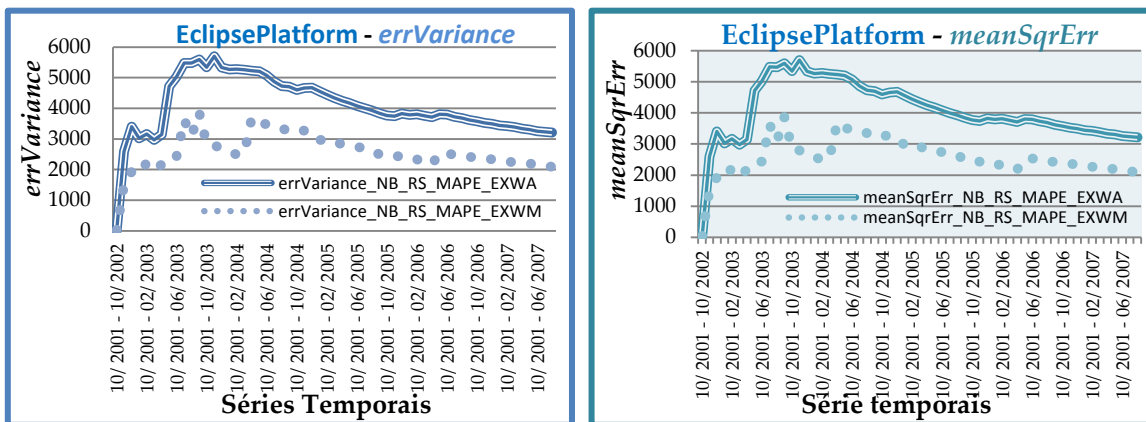
ANS030 - ARIMA NONSEASONAL (0,3,0)(0,0,0)

Nas figuras respeitantes aos modelos exibidos podemos indicar que o comportamento dos diferentes valores apresentados ao longo do tempo está associado geralmente a uma tendência linear estacionária. Os valores observados acerca do número de pedidos de evolução de *software* encontram-se dentro dos limites do intervalo de confiança. Em síntese, quando comparados os dados observados (linha vermelha) com o modelo de ajuste (linha azul fina) percebe-se que o modelo se encontra ajustado o suficiente para efetuar previsões sobre períodos futuros. Observando os valores previstos (linha azul espessa) e comparando-os com os valores dos dados observados, podemos concluir que o modelo faz previsões dentro de um nível de confiança aceitável, apesar de se verificar uma ligeira subestimação do número de pedidos de evolução de *software* na previsão, no período de previsão.

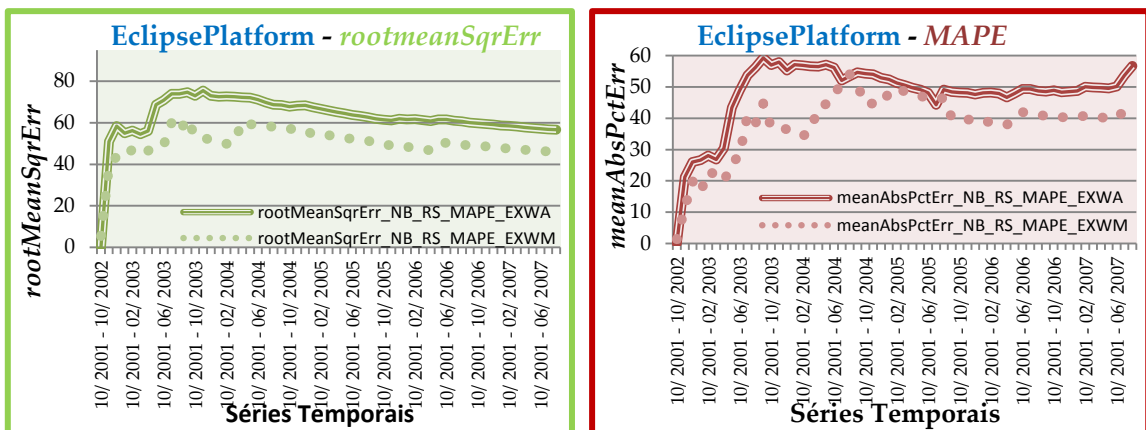
Apêndice X - Figuras: Software *Eclipse Platform* - Métricas de Erro — segundo os critérios *NormalizedBIC*, *RSQUARE* e *MAPE*

EXWA - Exponential Smoothing - Winters' Additive

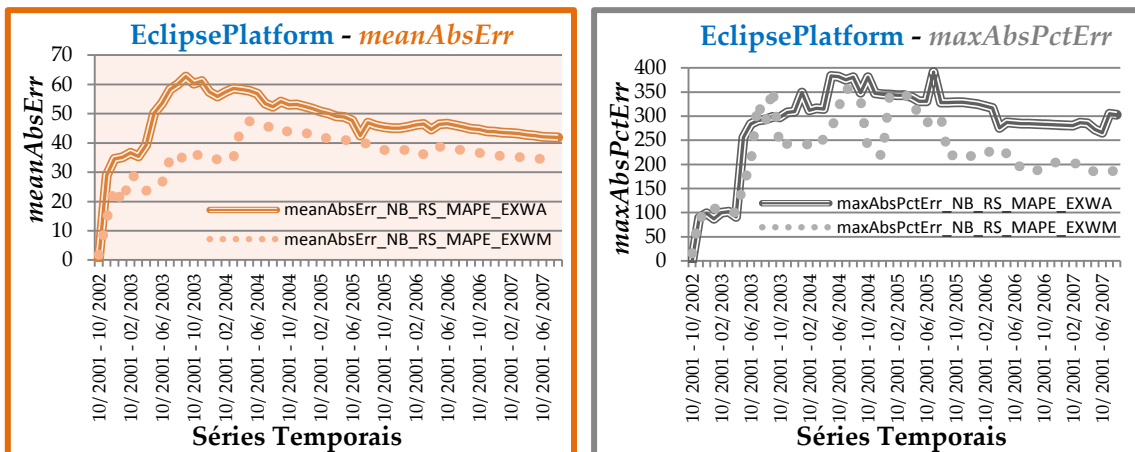
EXWM - Exponential Smoothing - Winters' Multiplicative



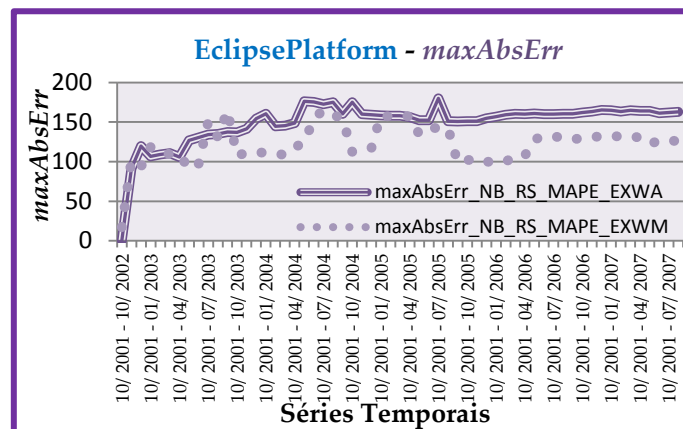
Pela análise das métricas de variância e de erro quadrático médio em cada um dos modelos selecionados pode observar-se que os modelos de previsão EXWA e EXWM apresentam um comportamento muito semelhante, destacando-se positivamente o modelo EXWM. Pela consulta das respetivas tabelas de *Excel* podemos confirmar que o modelo **EXWM** exibe uma média de valores (2578,626) inferior ao modelo EXWA (4073,670), o que torna **mais satisfatório**. De notar que, a partir da série treinada com dados de 10/2001 a 4/2004 os modelos em causa apresentam uma tendência linear decrescente.



Como se pode observar no gráfico a análise das métricas de raiz quadrada do erro quadrático médio e do erro médio percentual absoluto (MAPE - meanAbsPctErr) em cada um dos modelos selecionados mostra que os dois modelos de previsão exibem novamente um comportamento semelhante. Pela consulta das respetivas tabelas de *Excel* podemos confirmar que o modelo **EXWM** apresenta uma média de valores (50,217) inferior ao modelo EXWA (62,989) na métrica raiz quadrada do erro quadrático médio, e obtém uma média de valores (38,651) também inferior ao modelo EXWA (47,957) na métrica do erro médio percentual absoluto, o que torna **mais satisfatório**. A partir da série treinada com dados de 10/2001 a 4/2004 os modelos em causa apresentam uma tendência monótona (diminuindo e aumentando gradualmente).

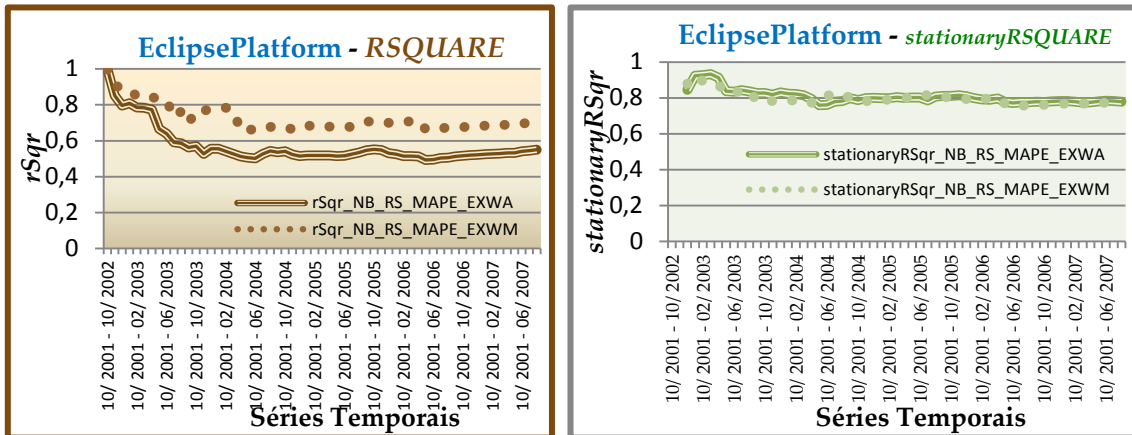


Como se pode verificar no gráfico a análise das métricas do erro médio absoluto e do erro médio percentual absoluto em cada um dos modelos seleccionados comprova que o modelo EXWM apresenta mais vezes valores mais favoráveis. Pela consulta das respetivas tabelas de *Excel* podemos confirmar que o modelo EXWM apresenta uma média de valores (35,893) inferior ao modelo EXWA (47,316) na 1ª métrica, e consegue uma média de valores (227,103) igualmente inferior ao modelo EXWA (288,076) na 2ª métrica, o que torna **mais razoável**. A partir da série treinada com dados de 10/2001 a 4/2004 os modelos em causa apresentam uma tendência monótona (diminuindo e aumentando gradualmente).



Como se pode verificar no gráfico a análise da métrica de erro máximo absoluto em cada um dos modelos seleccionados demonstra que os dois modelos de previsão apresentam um comportamento pouco regular, comprovando-se que o modelo EXWM apresenta mais vezes valores mais favoráveis. Pela consulta das respetivas tabelas de *Excel* podemos afirmar que o modelo EXWM apresenta uma média de valores (123,172) inferior ao modelo EXWA (149,212), o que torna **mais admissível**. A partir da série treinada com dados de 10/2001 a 4/2004 os modelos apresentam uma tendência monótona.

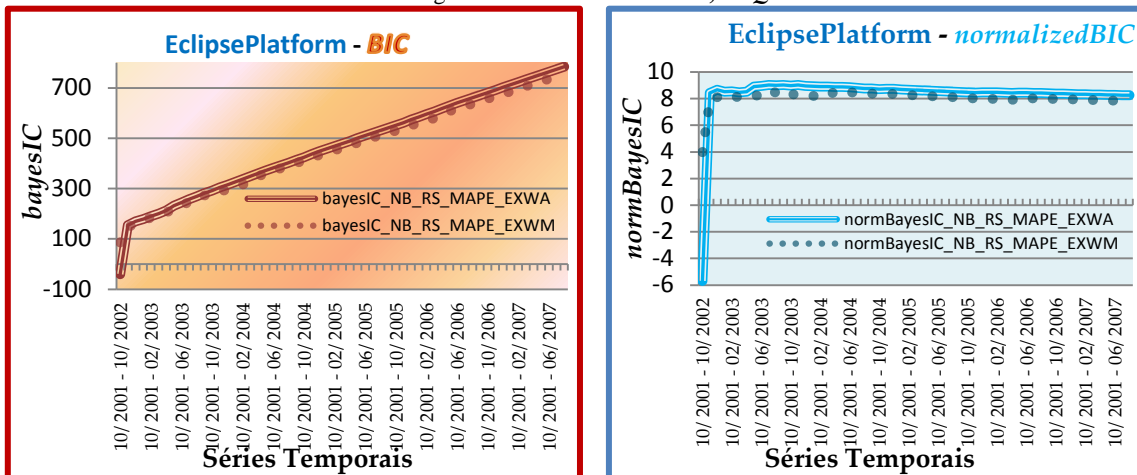
Apêndice XI - Figuras: Software *Eclipse Platform* - Métricas de Ajuste — segundo os critérios *NormBIC*, *RSQUARE* e *MAPE*



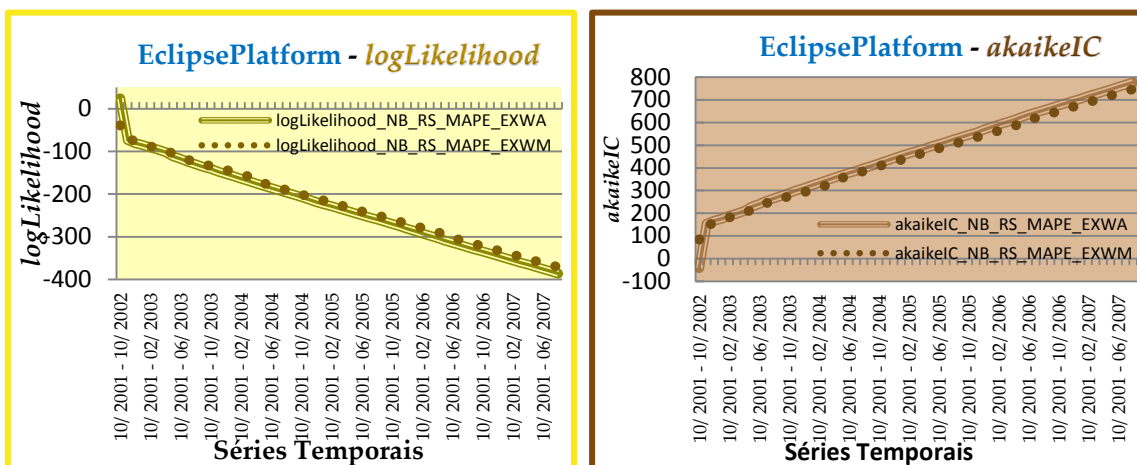
Como se pode observar no gráfico a análise da métrica de ajuste raiz quadrada (RSQUARE - rSqr) em cada um dos modelos selecionados mostra que os dois modelos de previsão exibem novamente um comportamento idêntico, comprovando-se que o modelo EXWM apresenta novamente valores mais favoráveis. Pela consulta das respetivas tabelas de *Excel* podemos confirmar que o modelo EXWM apresenta uma média de valores (0,724) inferior ao modelo EXWA (0,569), o que torna **mais satisfatório**.

Como se pode analisar no gráfico a métrica de ajuste raiz quadrada estacionária (stationaryRSQUARE) em cada um dos modelos selecionados indica que os dois modelos de previsão exibem novamente um comportamento semelhante, com valores muito próximos. Pela consulta das respetivas tabelas de *Excel* podemos verificar que o modelo EXWM apresenta uma média de valores (0,799) **muito próxima** da média associada ao modelo EXWA (0,807).

Apêndice XII - Figuras: Software Eclipse Platform - Métricas de Critérios de Informação –
segundo os critérios *NormBIC*, *RSQUARE* e *MAPE*



Como se pode observar no gráfico os critérios de informação Bayesianos (comum e o normalizado) em cada um dos modelos selecionados indica que os dois modelos de previsão apresentam novamente um comportamento idêntico, com valores muito próximos. Pela consulta respetivas tabelas de *Excel* podemos afirmar que o modelo **EXWM** apresenta uma média de valores (455,194) inferior ao modelo EXWA (471,446) no 1º critério, e exibe uma média de valores (8,080) inferior também ao modelo EXWA (8,365), o que torna **mais admissível**.



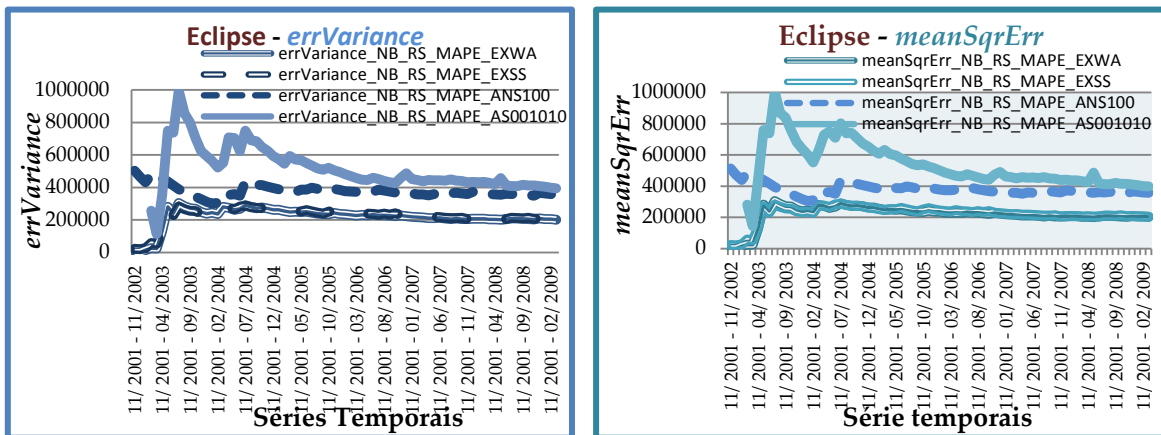
Analisando a métrica *log Likelihood* em cada um dos modelos percebe-se que apresentam um comportamento idêntico, com valores muito próximos. Pela consulta das tabelas de *Excel* podemos confirmar que o modelo **EXWM** exibe uma média de valores (-222,138) superior ao modelo EXWA (-230,264), o que torna **mais favorável, apesar dos valores serem muito próximos**.

O critério de informação *akaike* em cada um dos modelos explica que os dois modelos de previsão exibem novamente um comportamento conforme, com valores muito próximos. Nas tabelas de *Excel* verifica-se que o modelo **EXWM** dispõe de uma média de valores (450,277) inferior ao modelo EXWA (466,529), o que torna **mais satisfatório, apesar da proximidade**.

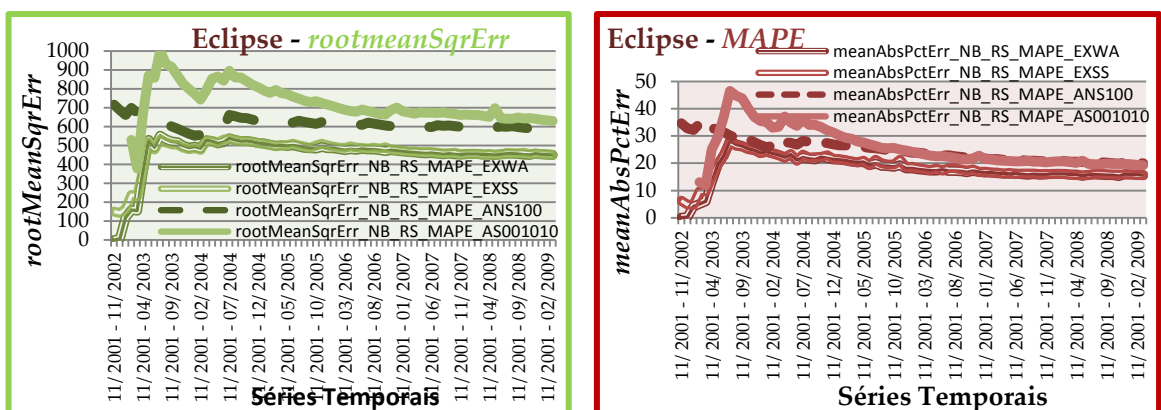
Apêndice XIII - Figuras: Software *Eclipse* - Métricas de Erro — segundo os critérios *NormBIC*, *RSQUARE* e *MAPE*

EXWA - Exponential Smoothing - Winters' Additive **EXSS** - Exponential Smoothing – Simple Seasonal

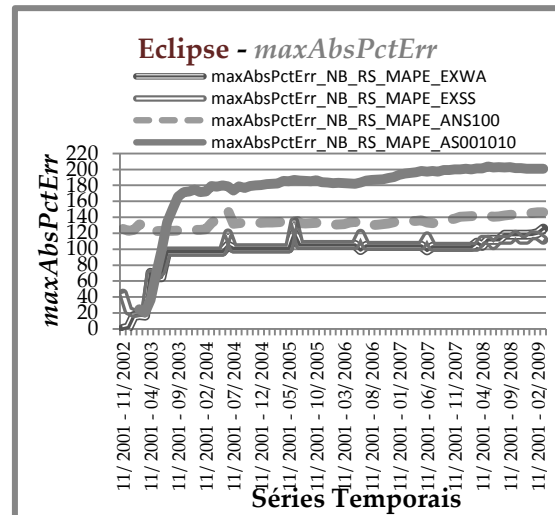
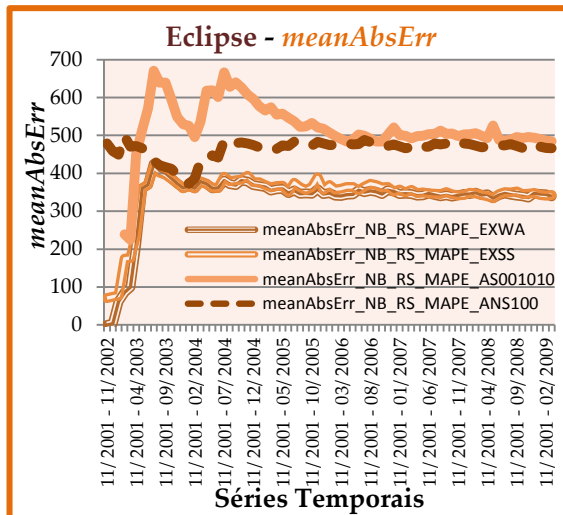
ANS100 – ARIMA (1,0,0)(0,0,0) **AS001010** – ARIMA (0,0,1)(0,1,0)



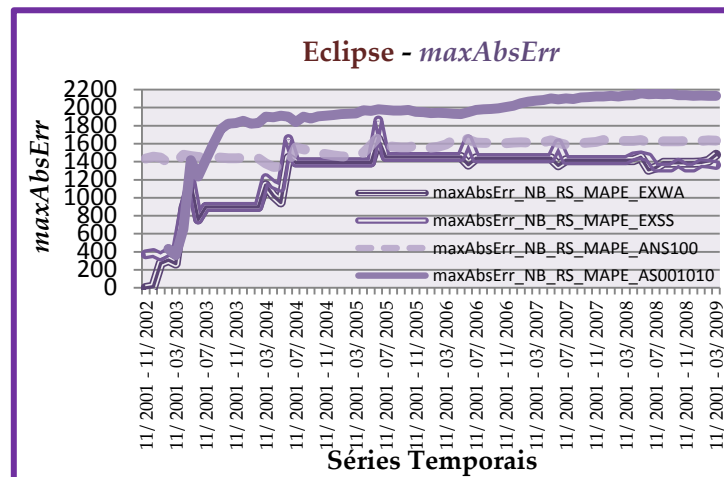
Pela análise das métricas de variância e de erro quadrático médio em cada um dos modelos selecionados pode observar-se que os modelos de previsão EXWA e EXSS apresentam um comportamento muito semelhante e mais favorável que os modelos ANS100 e AS001010. Pela consulta das respetivas tabelas de *Excel* podemos confirmar que o modelo **EXWA** exibe uma média de valores (219060,909) inferior ao modelo EXSS (222460,262), o que torna **menos mau**. A partir da série treinada com dados de 11/2001 a 7/2004 os modelos em causa apresentam uma tendência estacionária.



A análise das métricas de raiz quadrada do erro quadrático médio e do erro médio percentual absoluto (MAPE - meanAbsPctErr) em cada um dos modelos mostra que os mesmos exibem novamente um comportamento semelhante. Pela consulta das respetivas tabelas de *Excel* podemos confirmar que o modelo **EXWA** exibe uma média de valores (456,560) inferior ao modelo EXSS (464,964) na métrica raiz quadrada do erro quadrático médio, e obtém uma média de valores (17,780) inferior ao modelo EXSS (18,300) na métrica do erro médio percentual absoluto, o que torna **mais satisfatório, apesar da proximidade**. Os dois modelos ARIMA são penalizados pelo facto de apresentarem valores superiores. A partir da série treinada com dados de 11/2001 a 7/2004 os modelos exibem uma tendência estacionária.

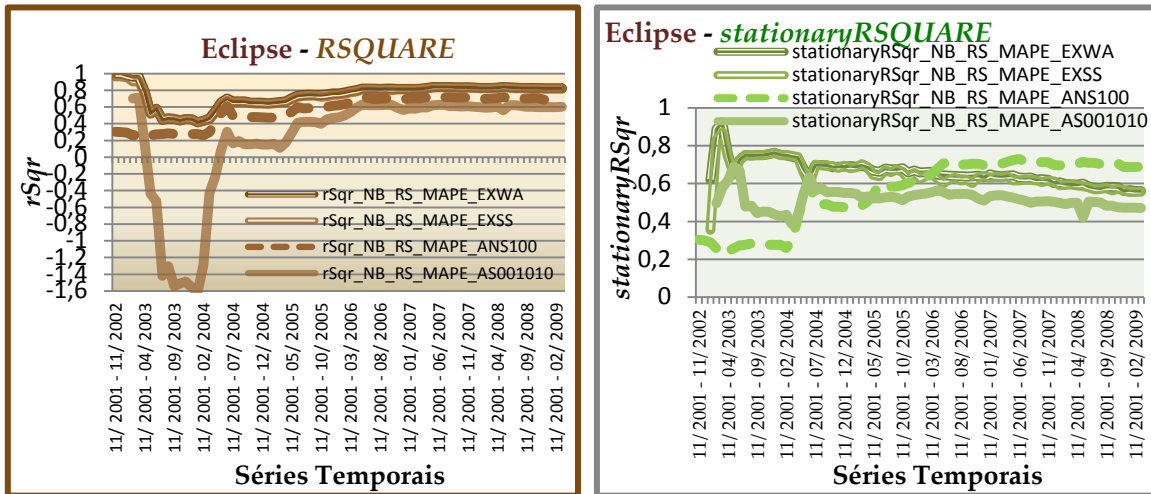


Como se pode verificar no gráfico a análise das métricas do erro médio absoluto e do erro médio percentual absoluto em cada um dos modelos selecionados comprova que o modelo EXWA apresenta mais vezes valores mais favoráveis. Pela consulta das respetivas tabelas de *Excel* podemos confirmar que o modelo **EXWA** apresenta uma média de valores (333,536) inferior ao modelo EXSS (344,027) na 1ª métrica, e consegue uma média de valores (97,094) igualmente inferior ao modelo EXSS (98,424) na 2ª métrica, o que torna **mais razoável, apesar dos valores serem muito aproximados**. A partir da 5ª série representada os modelos em causa apresentam uma tendência monótona (diminuindo e aumentando gradualmente).



Como se pode verificar no gráfico a análise da métrica de erro máximo absoluto em cada um dos modelos selecionados demonstra que o modelo EXWA apresenta mais vezes valores mais favoráveis. Pela consulta das respetivas tabelas de *Excel* podemos afirmar que o modelo **EXWA** apresenta uma média de valores (1251,662) inferior ao modelo EXSS (1280,569), o que torna **mais satisfatório, apesar da sua proximidade**. Os dois modelos ARIMA são penalizados pelo fato de exibirem valores superiores. A partir da série treinada com dados de 11/2001 a 7/2004 os modelos em causa apresentam uma tendência monótona (diminuindo e aumentando gradualmente).

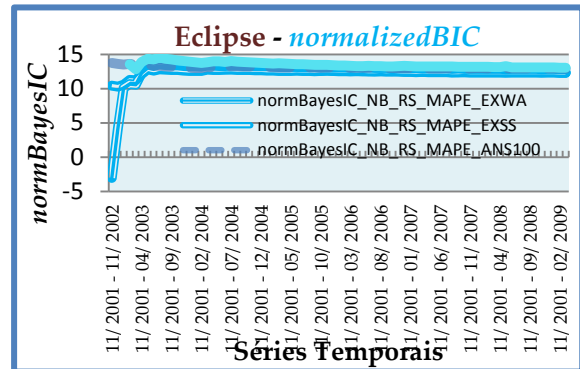
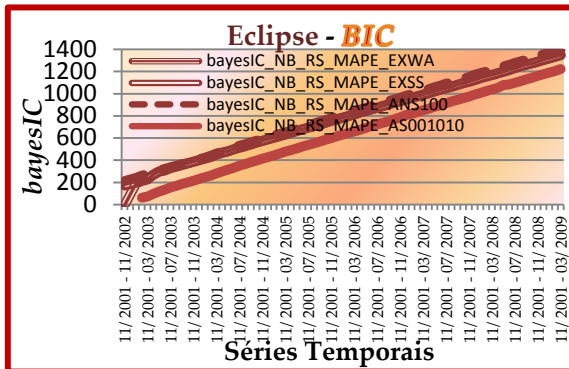
Apêndice XIV - Figuras: Software *Eclipse* - Métricas de Ajuste — segundo os critérios *NormBIC*, *RSQUARE* e *MAPE*



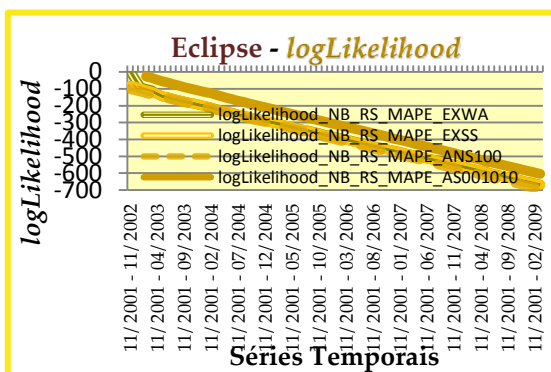
Como se pode observar no gráfico a análise da métrica de ajuste raiz quadrada (RSQUARE - rSqr) em cada um dos modelos seleccionados mostra que os modelos de previsão EXWA e EXSS exibem novamente um comportamento idêntico. Pela consulta das respetivas tabelas de *Excel* podemos confirmar que o modelo **EXWA** apresenta uma média de valores (0,745) superior ao modelo EXSS (0,735), o que torna **mais satisfatório, mas de uma forma pouco significativa**.

A métrica de ajuste raiz quadrada estacionária em cada um dos modelos indica que os modelos de previsão exibem novamente um comportamento semelhante, com valores muito próximos. Pela consulta das respetivas tabelas de *Excel* podemos verificar que o modelo **EXWA** exhibe a média de valores (0,663) **maior**, muito próxima da média associada ao modelo EXSS (0,649). O modelo ARIMA (1,0,0)(0,0,0) alcançou uma média de 0,559 e o modelo ARIMA (0,0,1)(0,1,0) atingiu um valor médio de 0,518.

Apêndice XV - Figuras: Software Eclipse - Métricas de Critérios de Informação — segundo os critérios *NormBIC*, *RSQUARE* e *MAPE*



Como se pode observar nas figuras os critérios de informação Bayesianos (comum e o normalizado) em cada um dos modelos elegidos indica que os modelos de previsão apresentam novamente um comportamento idêntico, com valores muito próximos. Pela consulta respetivas tabelas de *Excel* podemos afirmar que o modelo AS001010 apresenta uma média de valores (652,054) inferior aos modelos EXWA (775,579), EXSS (777,500) e ANS100 (807,407) no 1º critério, e o modelo EXWA volta a exibir uma média de valores (12,21) inferior aos restantes modelos EXSS (12,414), ANS100 (13,112) e AS001010 (13,508) no 2º critério, o que torna **um pouco melhor**.

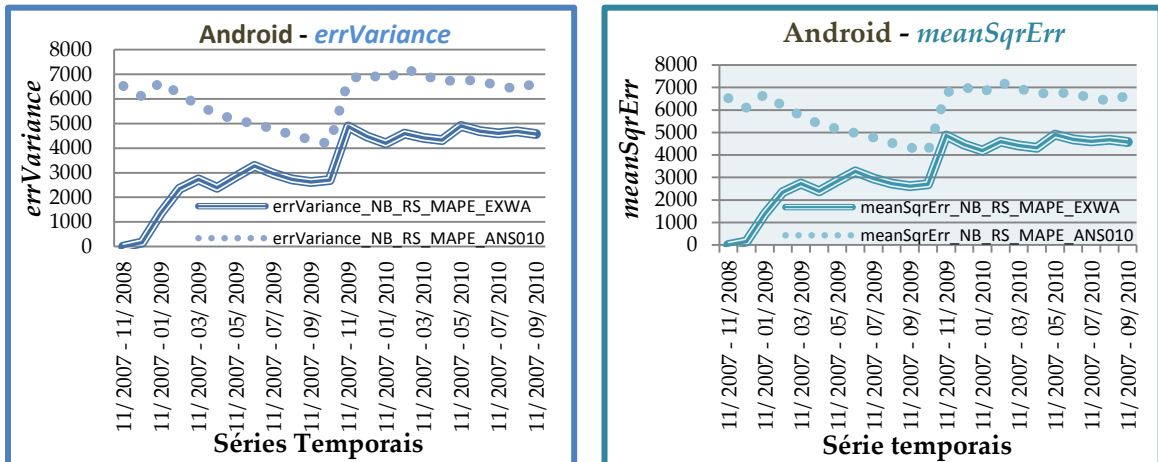


Analisando a métrica *log Likelihood* em cada um dos modelos percebe-se que apresentam um comportamento idêntico, com valores muito próximos. Pela consulta das tabelas de *Excel* podemos confirmar que o modelo AS001010 exibe uma média de valores (-320,781) superior aos modelos EXWA (-382,069), EXSS (-384,936) e ANS100 (-397,983), o que torna **um pouco mais favorável**.

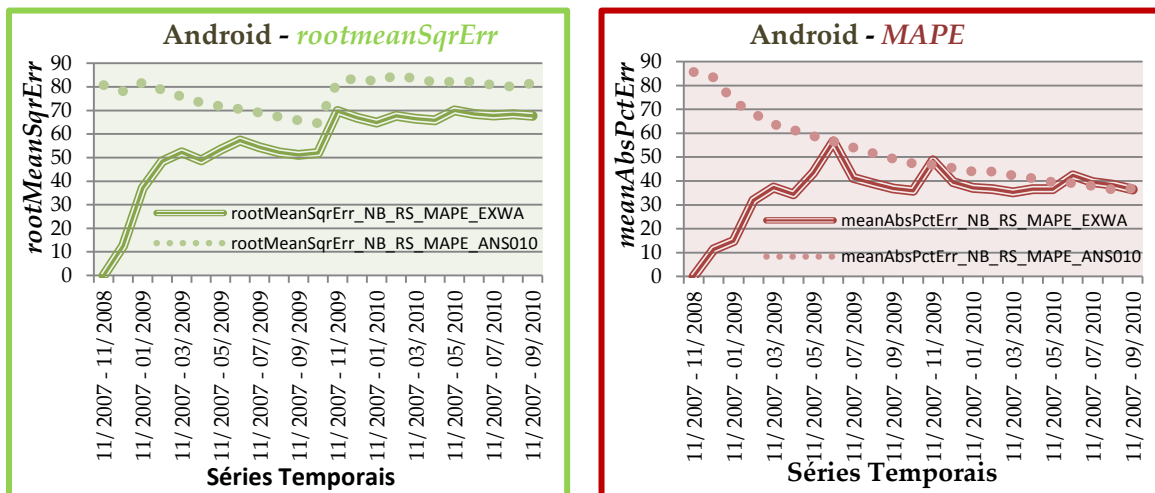
O critério de informação *akaike* em cada um dos modelos explica que os modelos de previsão exibem novamente um comportamento conforme, com valores muito próximos. Nas tabelas de *Excel* verifica-se que o modelo AS001010 dispõe de uma média de valores (801,966), inferior aos modelos EXWA (647,563), EXSS (773,873) e ANS100 (770,138), o que torna **um pouco mais satisfatório**.

Apêndice XVI – Figuras: Software Android - Métricas de Erro – segundo os critérios NormBIC, RSQUARE e MAPE

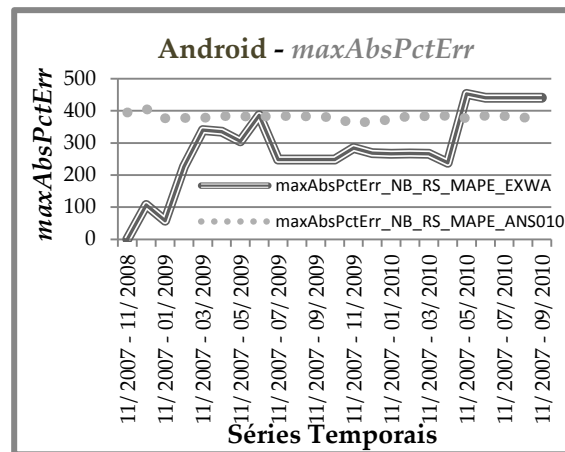
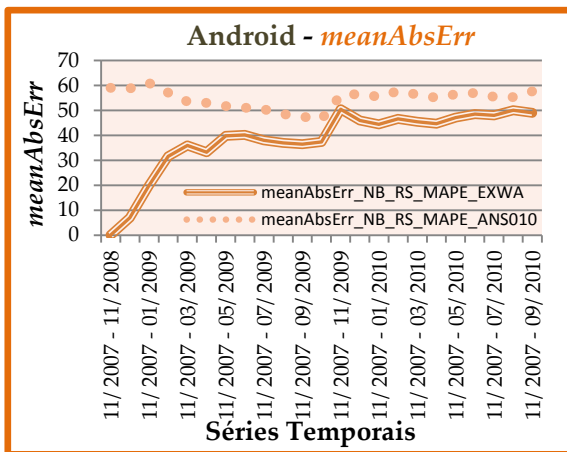
EXWA - Exponential Smoothing - Winters' Additive ANS100 – ARIMA (0,1,0)(0,0,0)



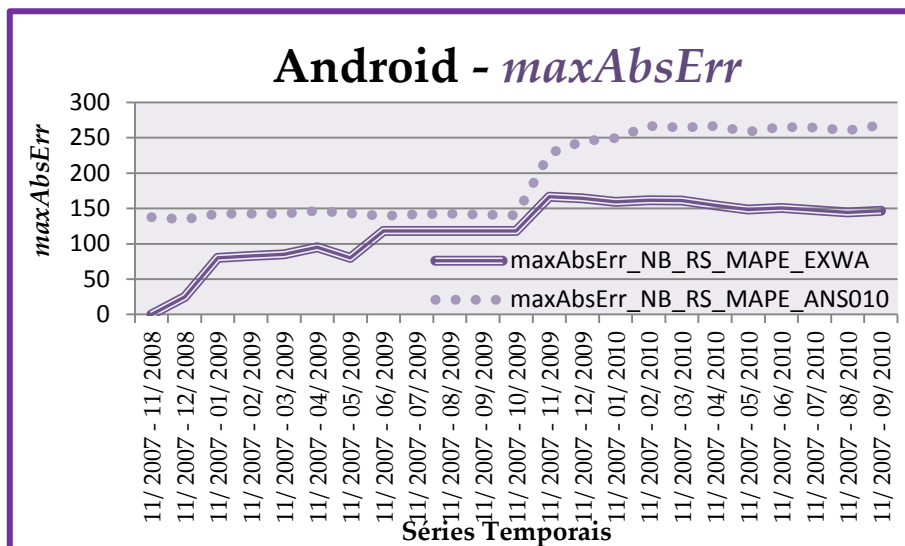
Pela análise das métricas de variância e de erro quadrático médio em cada um dos modelos selecionados pode observar-se que os modelos de previsão EXWA e EXWM apresentam um comportamento muito semelhante, nas duas figuras. Pela consulta das respetivas tabelas de *Excel* podemos confirmar que o modelo **EXWA** exibe uma média de valores (3324,553) inferior ao modelo ANS010 (6051,637), o que torna **mais favorável**. A partir da série treinada com dados de 11/2007 a 11/2009 os modelos em causa apresentam uma tendência linear monótona.



Como se pode observar nas figuras a análise das métricas de raiz quadrada do erro quadrático médio e do erro médio percentual absoluto (MAPE - meanAbsPctErr) em cada um dos modelos mostra que os dois modelos de previsão exibem novamente um comportamento semelhante, a partir da série treinada com dados de 11/2007 a 11/2009. Pela consulta das respetivas tabelas de *Excel* podemos confirmar que o modelo **EXWA** apresenta uma média de valores (54,935) inferior ao modelo ANS010 (77,545) na métrica raiz quadrada do erro quadrático médio, e obtém uma média de valores (35,212) também inferior ao modelo ANS010 (52,478) na métrica do erro médio percentual absoluto, o que torna um pouco **mais satisfatório**.

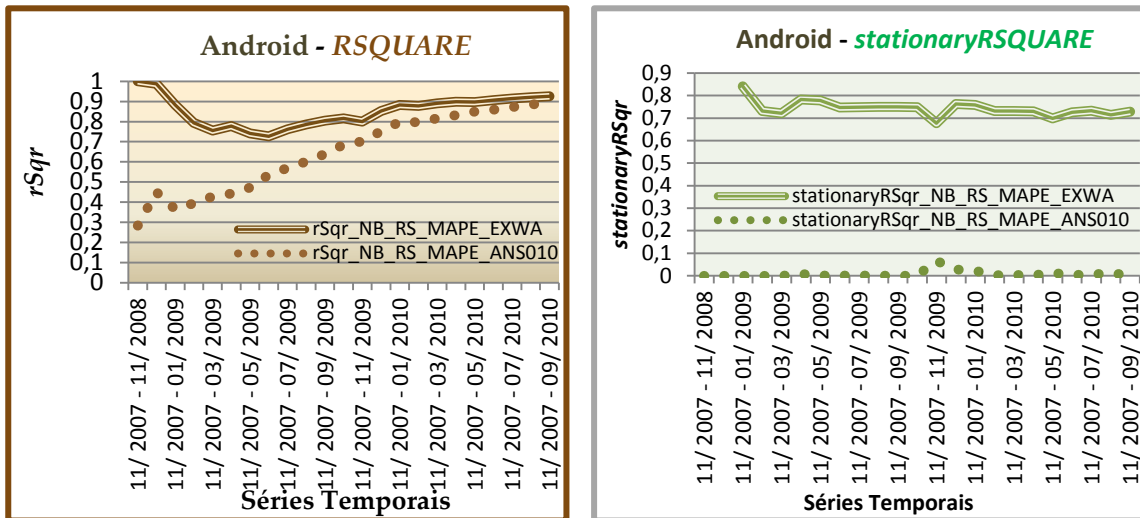


Como se pode verificar no gráfico a análise das métricas do erro médio absoluto e do erro médio percentual absoluto em cada um dos modelos comprova que o modelo EXWA apresenta mais uma vez valores mais favoráveis. Pela consulta das respetivas tabelas de *Excel* podemos confirmar que o modelo **EXWA** apresenta uma média de valores (38,147) inferior ao modelo ANS010 (54,694) na 1ª métrica, e consegue uma média de valores (285,100) igualmente inferior ao modelo ANS010 (381,646) na 2ª métrica, o que torna **mais razoável**. O modelo ANS010 assume uma tendência quase estacionária e o modelo EXWA apresenta uma tendência monótona (diminuindo e aumentando gradualmente).



Como se pode verificar no gráfico a análise da métrica de erro máximo absoluto em cada um dos modelos demonstra que o modelo EXWA apresenta mais uma vez valores mais favoráveis. Pela consulta das respetivas tabelas de *Excel* podemos afirmar que o modelo **EXWA** apresenta uma média de valores (119,246) inferior ao modelo ANS010 (196,992), o que torna um pouco **mais admissível**. A partir da série treinada com dados de 11/2007 a 11/2009 os modelos em causa apresentam uma tendência monótona (diminuindo e aumentando gradualmente).

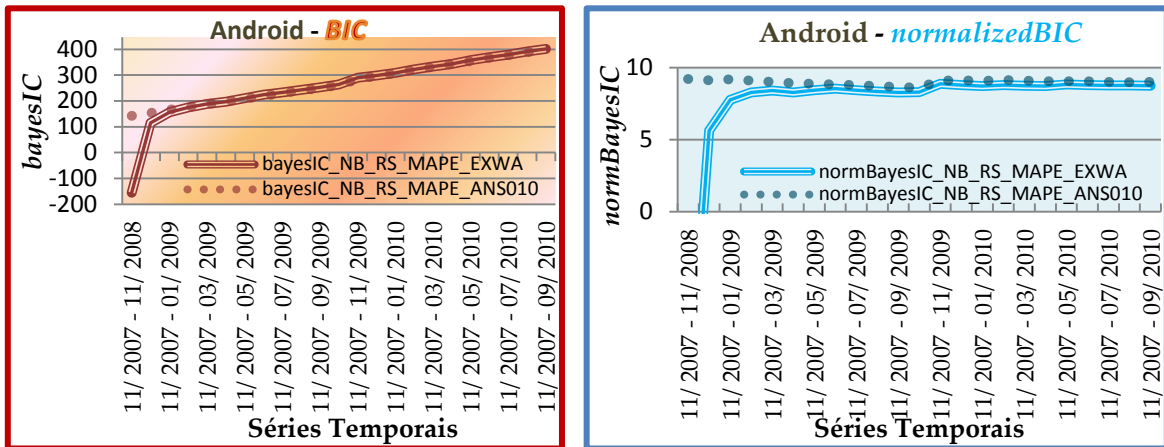
Apêndice XVII - Figuras: Software Android - Métricas de Ajuste — segundo os critérios *NormBIC*, *RSQUARE* e *MAPE*



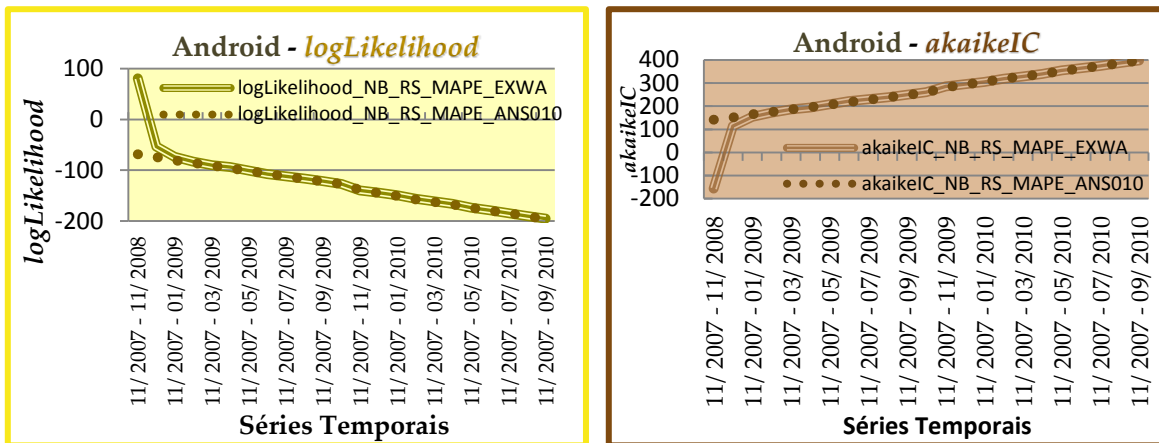
Como se pode observar no gráfico a evolução da métrica de ajuste raiz quadrada (RSQUARE - rSqr) em cada um dos modelos selecionados comprova que o modelo **EXWA** apresenta novamente valores mais favoráveis. Pela consulta das respetivas tabelas de *Excel* podemos confirmar que o modelo **EXWA** apresenta uma média de valores (0,852) superior ao modelo ANS010 (0,649), o que torna **mais francamente satisfatório**.

Como se pode analisar no gráfico a métrica de ajuste raiz quadrada estacionária em cada um dos modelos indica que os dois modelos de previsão exibem um comportamento muito distinto. Pela consulta das respetivas tabelas de *Excel* podemos verificar que o modelo **EXWA** apresenta uma média de valores (0,742) **muito distante** da média associada ao modelo **ANS010** (0,007).

Apêndice XVIII - Figuras: Software Android - Métricas de Critérios de Informação — segundo os critérios *NormBIC*, *RSQUARE* e *MAPE*



Como se pode observar no gráfico os critérios de informação Bayesianos (comum e o normalizado) em cada um dos modelos selecionados indica que ambos apresentam novamente um comportamento idêntico, com valores muito próximos. Pela consulta respetivas tabelas de *Excel* podemos afirmar que o modelo **EXWA** apresenta uma média de valores (256,423) inferior ao modelo ANS010 (269,725) no 1º critério, e exibe uma média de valores (7,419) inferior também ao modelo ANS010 (8,981), o que torna ligeiramente **mais admissível**.



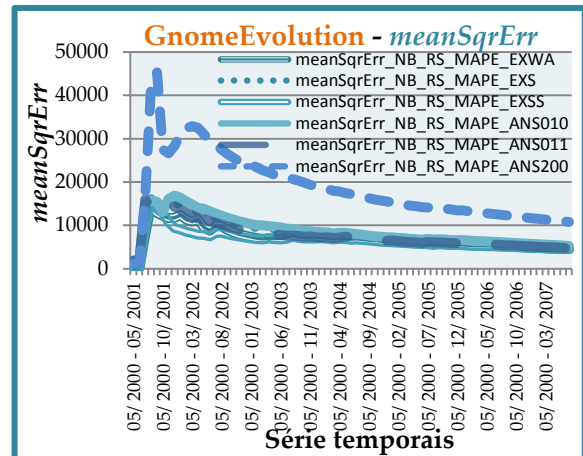
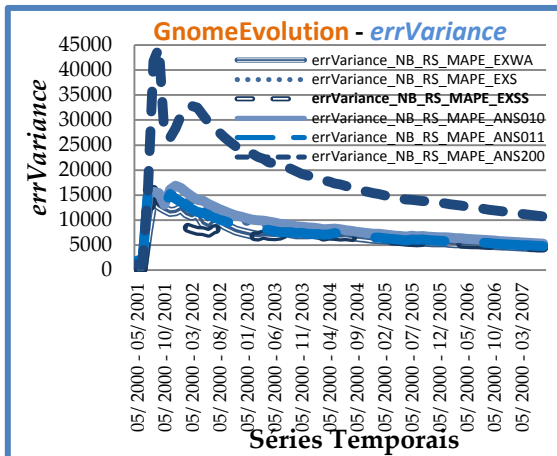
Analisando a métrica *log Likelihood* em cada um dos modelos percebe-se que apresentam um comportamento idêntico, com valores muito próximos. Pela consulta das tabelas de *Excel* podemos confirmar que o modelo **EXWA** exibe uma média de valores (-123,506) superior ao modelo ANS010 (-131,772), o que torna um pouco **mais favorável**.

O critério de informação *akaike* em cada um dos modelos explica que os dois modelos de previsão exibem novamente um comportamento conforme, com valores muito próximos. Nas tabelas de *Excel* verifica-se que o modelo **EXWA** dispõe de uma média de valores (253,012) inferior ao modelo ANS010 (267,544), o que torna um pouco **mais satisfatório**.

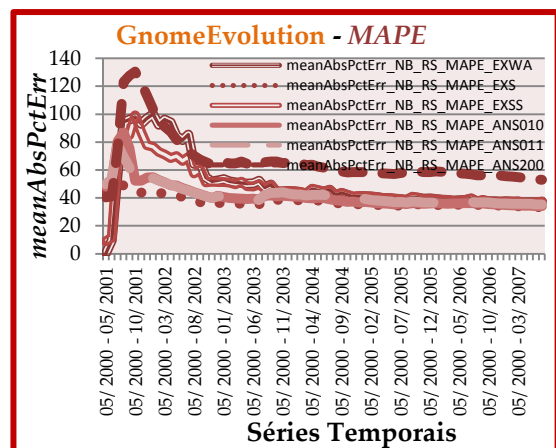
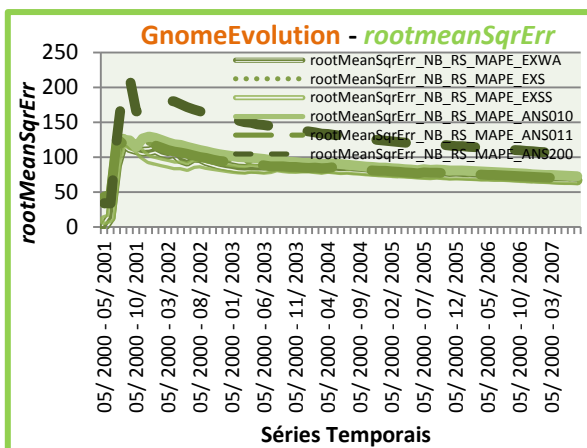
Apêndice XIX – Figuras: Software *Gnome Evolution* - Métricas de Erro –
segundo os critérios *NormBIC*, *RSQUARE* e *MAPE*

EXWA - Exponential Smoothing - Winters' Additive
EXSS - Exponential Smoothing – Simple Seasonal
ANS010 – ARIMA (0,1,0)(0,0,0)
ANS011 – ARIMA (0,1,1)(0,0,0)

EXS - Exponential Smoothing – Simple
ANS200 – ARIMA (2,0,0)(0,0,0)

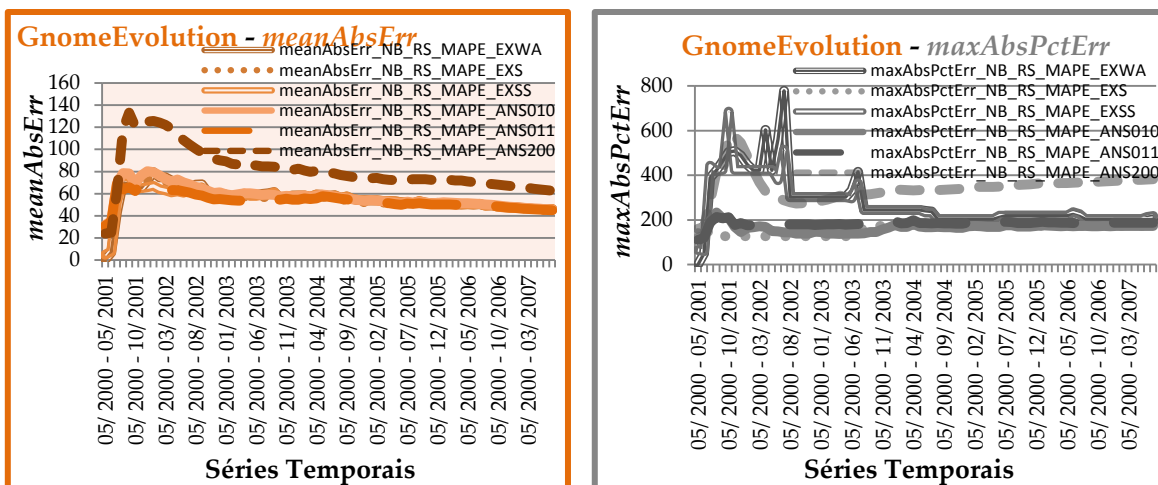


Pela análise das métricas de variância e de erro quadrático médio em cada um dos modelos selecionados pode observar-se que todos à exceção do modelo ANS200 apresentam um comportamento muito semelhante. Pela consulta das respetivas tabelas de *Excel* podemos confirmar que o modelo **EXSS** exibe uma média de valores (6586,139) inferior ao modelo mais próximo, EXWA (6932,665), o que torna um pouco **melhor**. Os modelos ARIMA são penalizados pelo facto de apresentarem valores superiores. A partir da série treinada com dados de 5/2000 a 1/2003 os modelos em causa apresentam uma tendência monótona, quase estacionária, à exceção do modelo ANS200, que assume uma tendência ligeiramente decrescente.

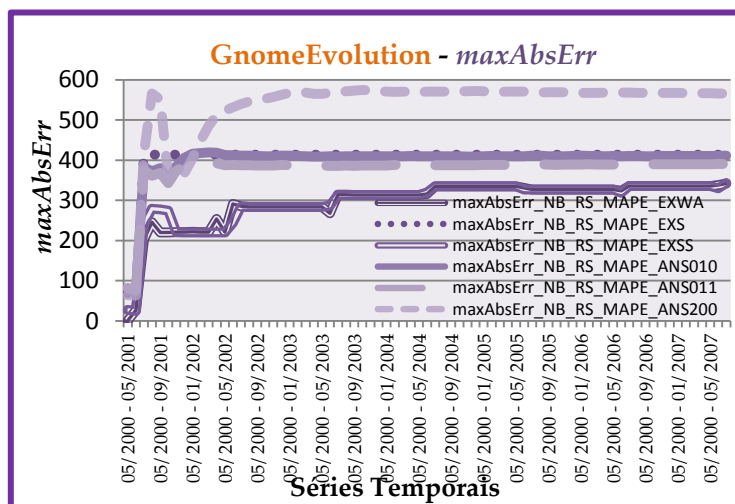


A análise das métricas de raiz quadrada do erro quadrático médio e do erro médio percentual absoluto (MAPE - meanAbsPctErr) em cada um dos modelos mostra que os mesmos exibem novamente um comportamento semelhante. Pela consulta das respetivas tabelas de *Excel* podemos confirmar que o modelo **EXSS** exibe uma média de valores (79,609) inferior ao modelo mais imediato, EXWA (81,407), na métrica raiz quadrada do erro quadrático médio, mas obtém uma

média de valores (47,870) superior, embora aproximada, ao modelo EXS (37,480) na métrica do erro médio percentual absoluto, o que um pouco **melhor**. A partir da série treinada com dados de 5/2000 a 1/2003 os modelos exibem uma tendência quase estacionária.

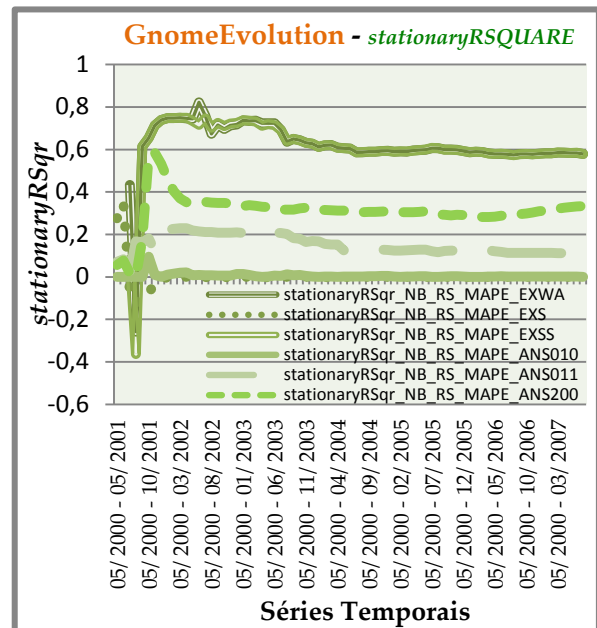
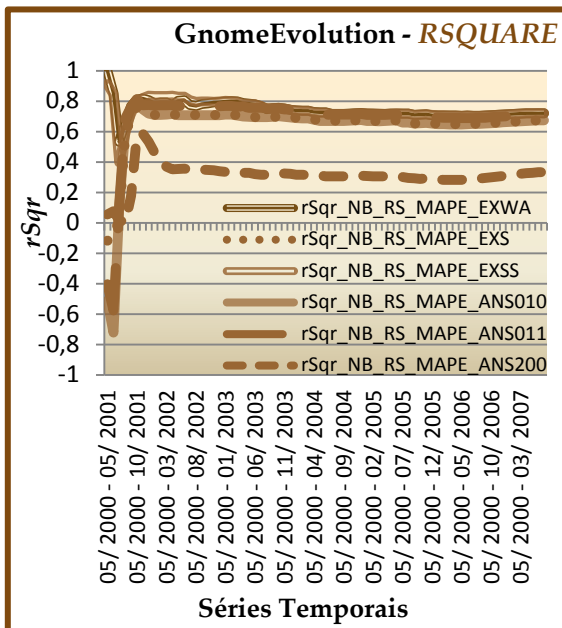


Como se pode verificar nos gráficos a análise das métricas do erro médio absoluto e do erro médio percentual absoluto em cada um dos modelos selecionados comprova que os modelos em causa apresentam geralmente uma tendência monótona (diminuindo e aumentando gradualmente). Pela consulta das respetivas tabelas de *Excel* podemos confirmar que o modelo **ANS011** apresenta na 1ª métrica uma média de valores (53,831) inferior aos restantes modelos, mas próximo do modelo EXSS (54,440), e na 2ª métrica o modelo **EXS** conseguiu uma média de valores (157,509) inferior aos restantes modelos.



Como se pode verificar no gráfico a análise da métrica de erro máximo absoluto em cada um dos modelos selecionados demonstra que o modelo EXWA apresenta mais vezes valores mais favoráveis. Pela consulta das respetivas tabelas de *Excel* podemos afirmar que o modelo **EXWA** apresenta uma média de valores (296,769) inferior aos restantes modelos, mas próximo do modelo EXSS (298,546), o que torna **apenas um pouco melhor**. A partir da série treinada com dados de 5/2000 a 3/2002 os modelos em causa apresentam uma tendência monótona, quase estacionária.

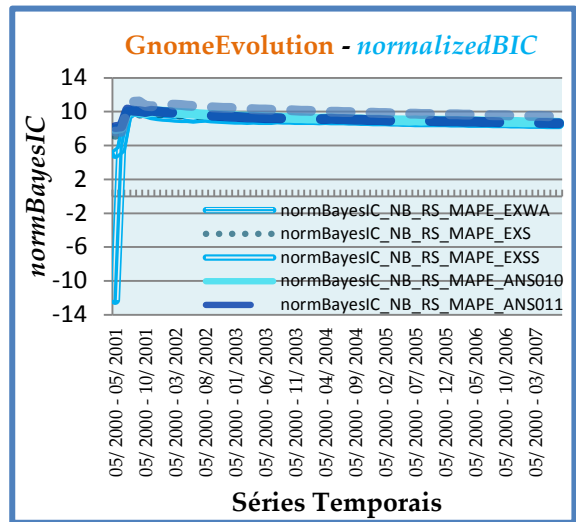
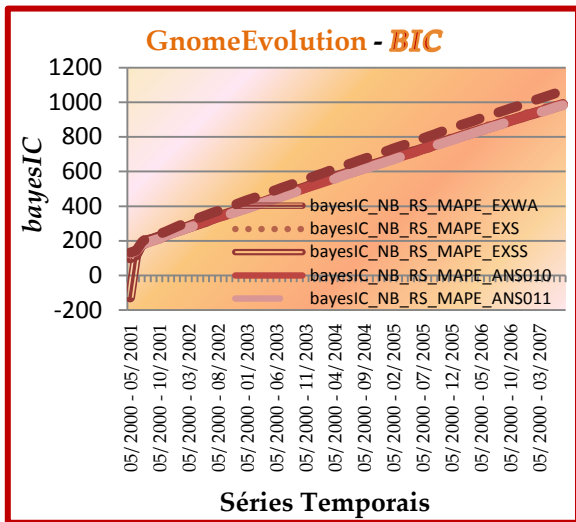
Apêndice XX - Figuras: *Software Gnome Evolution* - Métricas de Ajuste —
segundo os critérios *NormBIC*, *RSQUARE* e *MAPE*



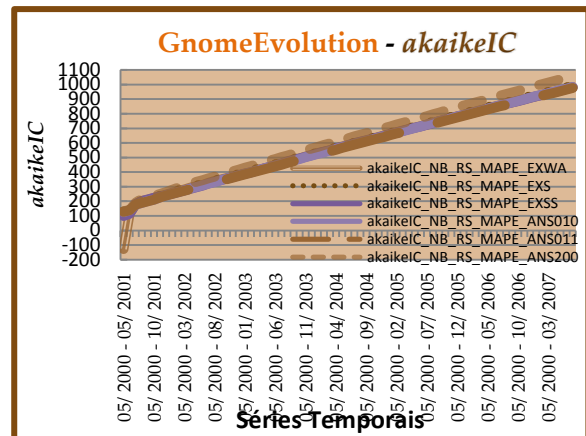
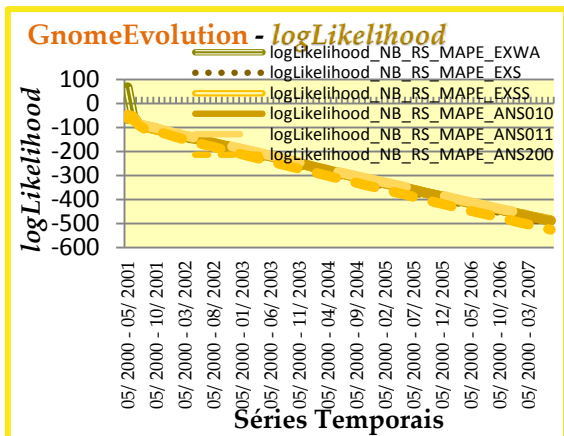
Como se pode observar no gráfico a análise da métrica de ajuste raiz quadrada (RSQUARE - $rSqr$) em cada um dos modelos selecionados mostra que todos os modelos de previsão, à exceção do modelo ANS200 exibem novamente um comportamento idêntico. Pela consulta das respetivas tabelas de *Excel* podemos confirmar que o modelo **EXSS** apresenta uma média de valores (0,744) superior aos restantes modelos, mas muito próximo do modelo EXWA (0,743), o que torna **mais satisfatório**.

A métrica de ajuste raiz quadrada estacionária em cada um dos modelos indica que os modelos de previsão exibem novamente um comportamento semelhante, com valores muito próximos. Pela consulta das respetivas tabelas de *Excel* podemos verificar que o modelo **EXWA** exhibe a média de valores (0,622) **maior**, muito próxima da média associada ao modelo EXSS (0,615).

Apêndice XXI - Figuras: Software Gnome Evolution –
Métricas de Critérios de Informação – segundo os critérios *NormBIC*, *RSQUARE* e *MAPE*

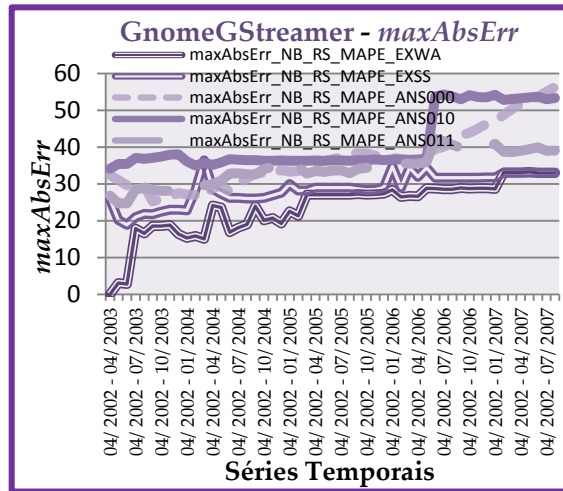


Como se pode observar nas figuras os critérios de informação Bayesianos (comum e o normalizado) em cada um dos modelos elegidos indica que os modelos de previsão apresentam novamente um comportamento idêntico, com valores muito próximos. Pela consulta respetivas tabelas de *Excel* podemos afirmar que o modelo **ANS011** apresenta uma média de valores (577,194) inferior aos outros modelos, mas próximo do modelo EXSS (580,420) no 1º critério, e o modelo **EXWA** exibe uma média de valores (8,763) inferior aos restantes modelos, próximo do modelo EXSS (8,861).

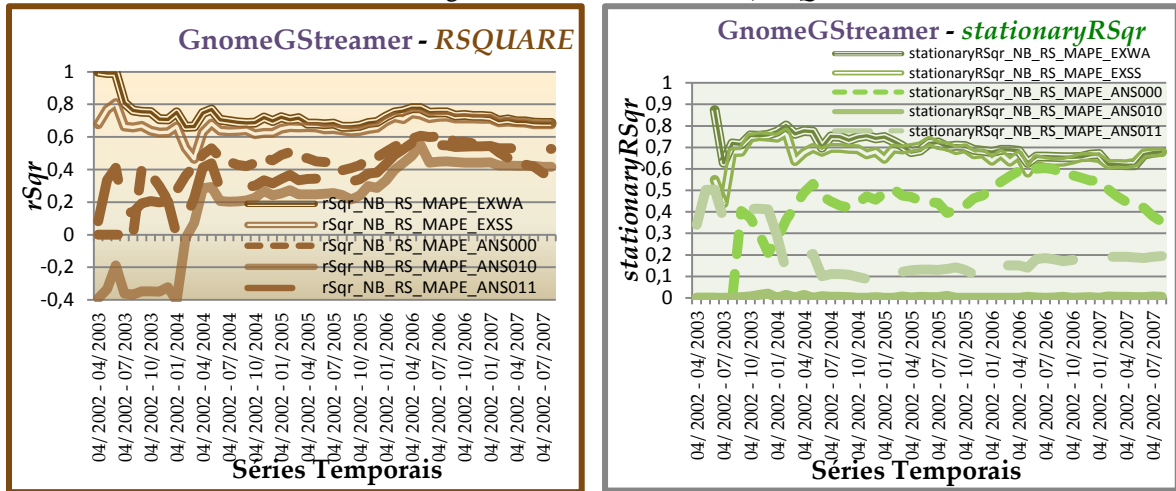


Analisando a métrica *log Likelihood* em cada um dos modelos percebe-se que apresentam um comportamento idêntico, com valores muito próximos. Pela consulta das tabelas de *Excel* podemos confirmar que o modelo **ANS011** exibe uma média de valores (-282,942) superior aos restantes modelos, mas próximo do modelo **EXWA** (-285,134).

O critério de informação *akaike* em cada um dos modelos explica que os modelos de previsão exibem novamente um comportamento conforme, com valores muito próximos. Nas tabelas de *Excel* verifica-se que o modelo **ANS011** dispõe de uma média de valores (571,885) inferior aos outros modelos, mas próximo do modelo **EXWA** (576,268).

Apêndice XXII – Figuras: Software *Gnome GStreamer* - Métricas de Erro –
segundo os critérios *NormBIC*, *RSQUARE* e *MAPE***EXWA** - Exponential Smoothing - Winters' Additive**ANS000** – ARIMA (0,0,0)(0,0,0)

Como se pode verificar no gráfico a análise da métrica de erro máximo absoluto em cada um dos modelos demonstra que o modelo EXWA apresenta mais vezes valores mais favoráveis. Pela consulta das respetivas tabelas de Excel podemos afirmar que o modelo EXWA apresenta uma média de valores (23,757) inferior ao modelo EXSS (28,739), o que torna mais admissível. Em geral, os modelos apresentam uma tendência monótona (diminuindo e aumentando gradualmente), à exceção do modelo ANS010 que no período de 5/2006 a 6/2006 sofre um visível aumento.

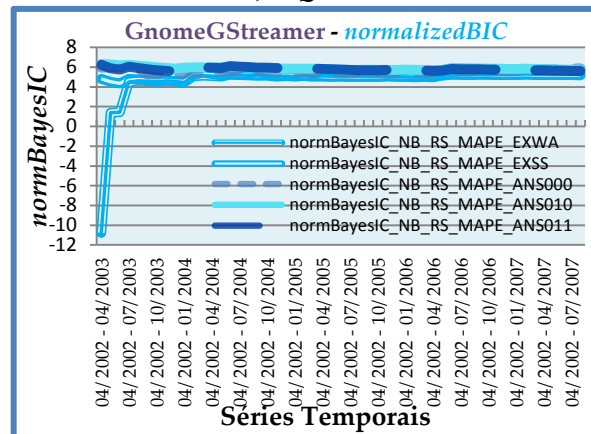
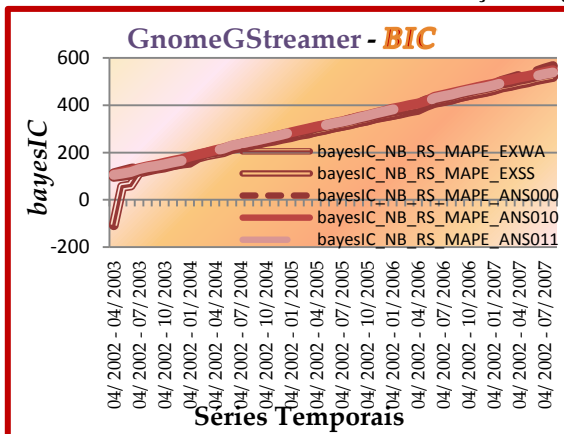
Apêndice XXIII - Figuras: Software *Gnome GStreamer* - Métricas de Ajuste —segundo os critérios *NormBIC*, *RSQUARE* e *MAPE*

Como se pode observar no gráfico a análise da métrica de ajuste raiz quadrada (*RSQUARE* - *rSqr*) em cada um dos modelos seleccionados mostra que os modelos de previsão EXWA e EXSS exibem novamente um comportamento idêntico. Pela consulta das respetivas tabelas de *Excel* podemos confirmar que o modelo EXWA apresenta uma média de valores (0,731) superior ao modelo EXSS (0,669), o que torna mais satisfatório. Os modelos ARIMA são penalizados pelo facto de apresentarem valores inferiores.

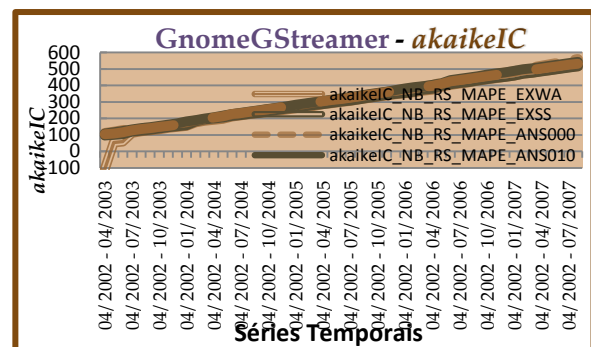
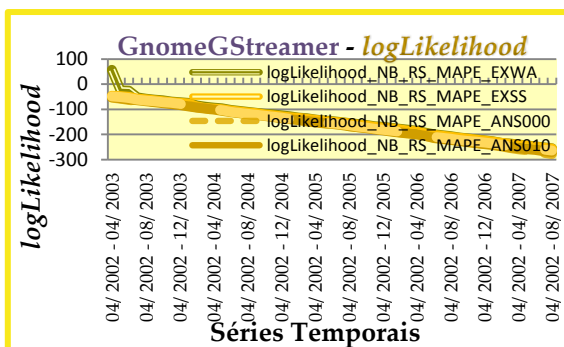
A métrica de ajuste raiz quadrada estacionária em cada um dos modelos indica que os modelos de previsão EXWA e EXSS exibem novamente um comportamento semelhante, com valores muito próximos. Pela consulta das respetivas tabelas de *Excel* podemos verificar que o modelo EXWA exhibe a média de valores (0,703) maior, muito próxima da média associada ao modelo EXSS (0,668). Os modelos ARIMA são penalizados pelo facto de apresentarem valores inferiores.

Apêndice XXIV - Figuras: Software *Gnome GStreamer* -

Métricas de Critérios de Informação – segundo os critérios *NormBIC*, *RSQUARE* e *MAPE*



Como se pode observar nos figuras os critérios de informação Bayesianos (comum e o normalizado) em cada um dos modelos elegidos indica que os modelos de previsão apresentam novamente um comportamento idêntico, com valores muito próximos. Pela consulta respetivas tabelas de *Excel* podemos afirmar que o modelo **EXWA** apresenta uma média de valores (301,204) inferior aos modelos EXSS (308,148), ANS011 (323,989), ANS000 (327,497) e ANS010 (327,706) no **1º critério**, e o modelo **EXWA** volta a exibir uma média de valores (4,624) inferior aos restantes modelos EXSS (5,063), ANS000 (5,601), ANS011 (5,801) e ANS010 (5,884) no **2º critério**, o que torna mais admissível, apesar dos valores serem muito próximos.

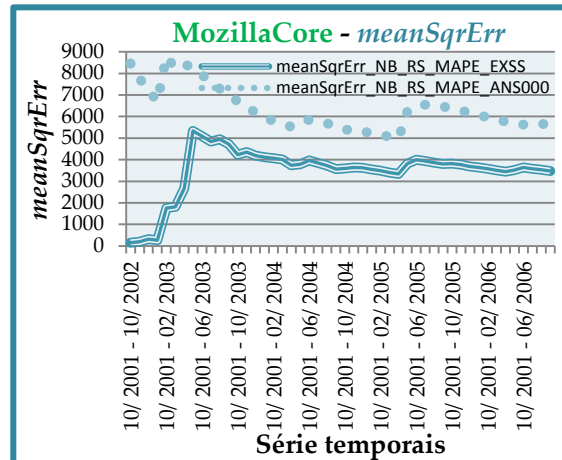
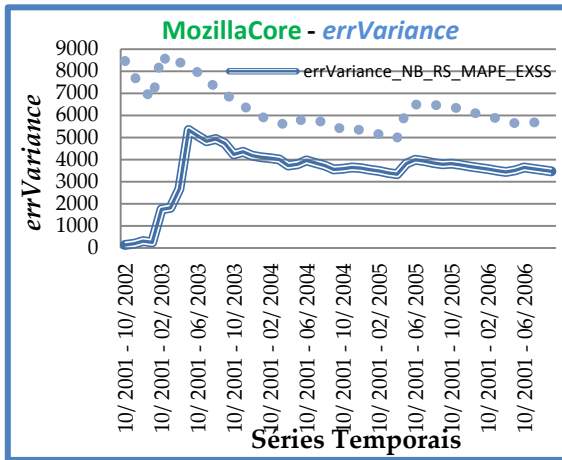


Analisando a métrica *log Likelihood* em cada um dos modelos percebe-se que apresentam um comportamento idêntico, com valores muito próximos. Pela consulta das tabelas de *Excel* podemos confirmar que o modelo **EXWA** exibe uma média de valores (-145,243) superior aos modelos EXSS (-150,501), ANS000 (-160,176), ANS011 (-156,683) e ANS010 (-160,312). O critério de informação *akaike* em cada um dos modelos explica que os modelos de previsão exibem novamente um comportamento conforme, com valores muito próximos. Nas tabelas de *Excel* verifica-se que o modelo **EXWA** exibe uma média de valores (296,486) inferior aos modelos EXSS (305,002), ANS011 (319,366), ANS000 (324,352) e ANS010 (324,624), o que torna mais satisfatório.

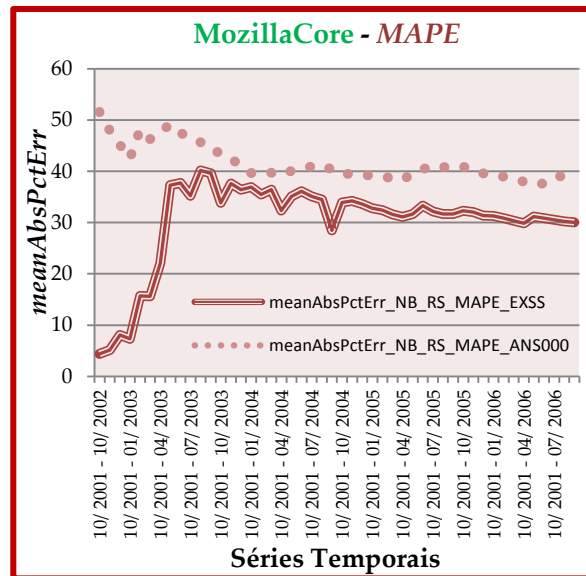
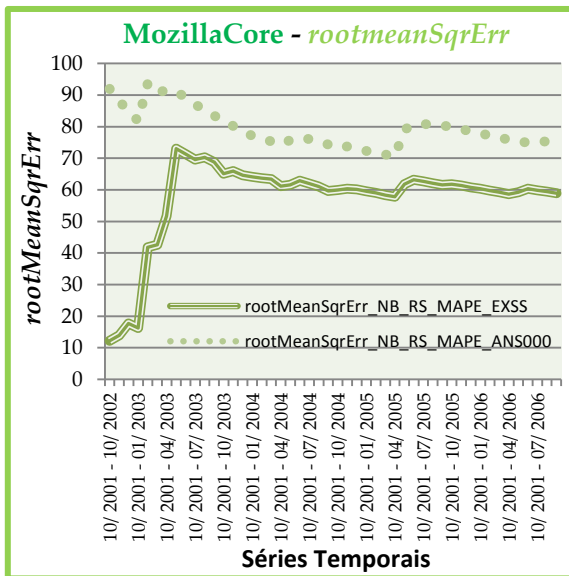
Apêndice XXV – Figuras: Software *Mozilla Core* - Métricas de Erro — segundo os critérios *NormBIC*, *RSQUARE* e *MAPE*

EXSS - Exponential Smoothing – Simple Seasonal

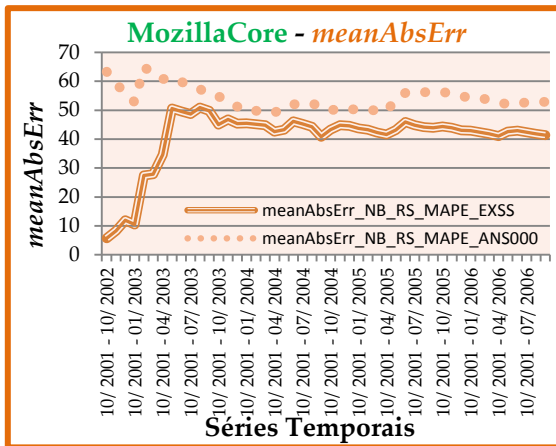
ANS000 – ARIMA (0,0,0)(0,0,0)



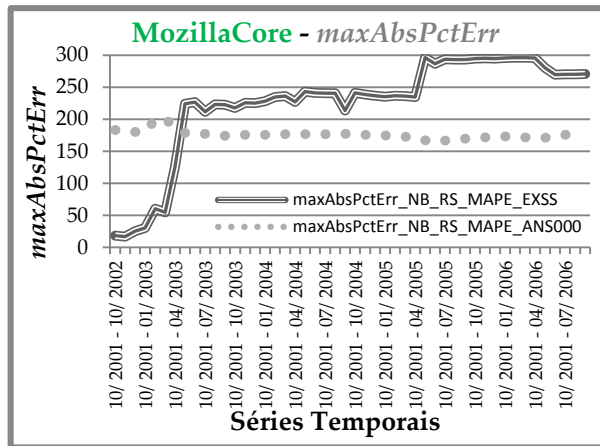
Pela análise das métricas de variância e de erro quadrático médio em cada um dos modelos selecionados pode observar-se que o modelo de previsão **EXSS** apresenta melhores resultados do que o modelo ANS000. Pela consulta das respetivas tabelas de *Excel* podemos confirmar que o modelo EXSS exibe uma média de valores (3866,671) inferior ao modelo ANS000 (6116,271), o que torna mais favorável. No período de 10/2001 até 06/2003 o modelo EXSS exibe uma tendência linear crescente e o modelo ANS000 apresenta um comportamento irregular. No período de 06/2003 até 04/2005 os dois modelos mostram uma tendência decrescente, que a partir dessa data se mantém monótona.



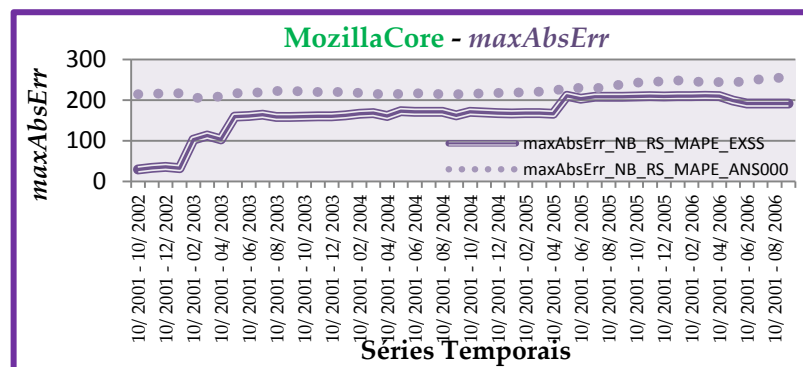
Como se pode observar nas figuras a análise das métricas de raiz quadrada do erro quadrático médio e do erro médio percentual absoluto (MAPE - meanAbsPctErr) em cada um dos modelos mostra que o modelo de previsão **EXSS** apresenta melhores resultados do que o modelo ANS000. Pela consulta das respetivas tabelas de *Excel* podemos confirmar que o modelo **EXSS** apresenta uma média de valores (62,057) inferior ao modelo ANS000 (78,047) na métrica raiz quadrada do erro quadrático médio, e obtém uma média de valores (33,123) também inferior ao modelo ANS000 (40,784) na métrica do erro médio percentual absoluto, o que torna **mais favorável**.



Como se pode verificar no gráfico a análise da métrica do erro médio absoluto em cada um dos modelos comprova que o modelo EXSS apresenta mais uma vez valores mais favoráveis. Pela consulta das respetivas tabelas podemos confirmar que o modelo **EXSS** apresenta uma média de valores (44,188) inferior ao modelo ANS000 (53,425), o que torna **mais razoável**. O modelo ANS000 assume uma tendência irregular (com valores entre 49 e 61) e o modelo EXWA apresenta uma tendência crescente no período de 10/2001 a 6/2003 e a partir dessa data exibe uma tendência monótona (diminuindo e aumentando gradualmente).

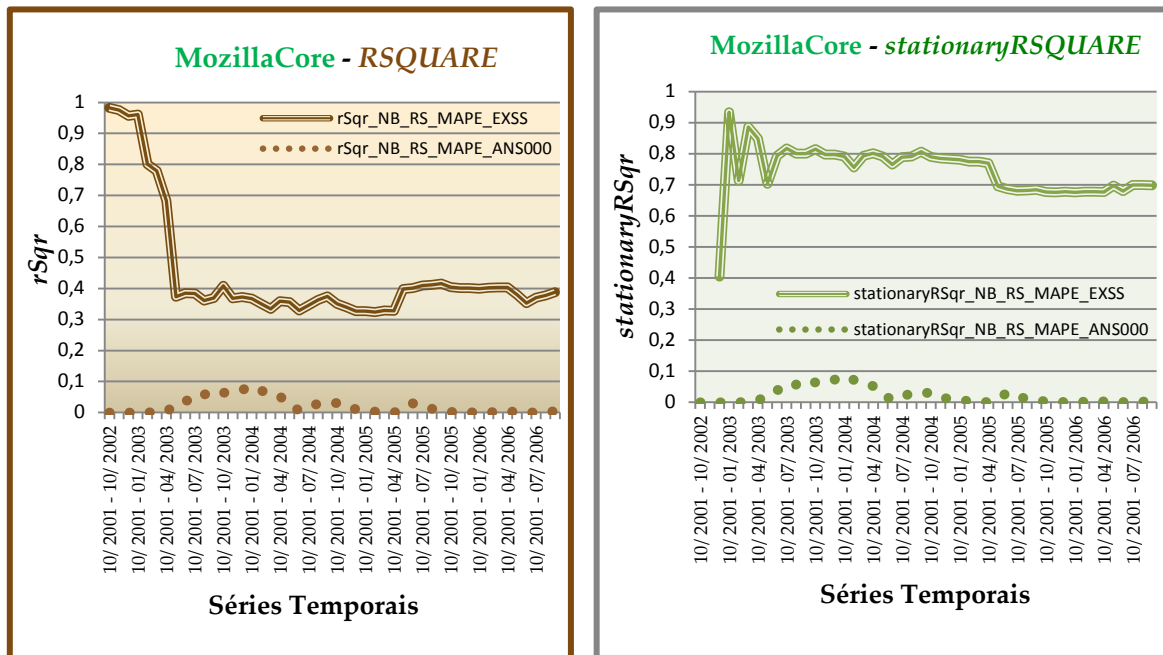


Com a análise da métrica do erro médio percentual absoluto em cada um dos modelos pode atestar-se que o modelo ANS000 apresenta valores mais favoráveis, a partir de 5/2003. Pela consulta das respetivas tabelas podemos confirmar que o modelo ANS000 apresenta uma média de valores (174,431) consideravelmente inferior ao modelo EXSS (251,528). O modelo ANS000 assume uma tendência quase estacionária e o modelo EXSS apresenta uma tendência crescente no período de 10/2001 a 6/2003, que a partir dessa data se converte em monótona até 5/2005, e posteriormente sofre um aumento gradual até 6/2005, tornando-se novamente monótona.



Como se pode averiguar no gráfico a análise da métrica de erro máximo absoluto em cada um dos modelos demonstra que o modelo EXSS apresenta mais vez valores mais favoráveis. Pela consulta das respetivas tabelas de *Excel* podemos afirmar que o modelo **EXSS** apresenta uma média de valores (179,587) inferior ao modelo ANS000 (228,133), o que torna **mais admissível**. O modelo ANS000 apresenta um comportamento quase estacionário (tendência monótona com valores entre 200 e 250, aproximadamente). O modelo EXSS exibe uma tendência crescente no período de 10/2001 a 5/2003 e monótona, a partir dessa data (com valores entre 150 e 210, sensivelmente).

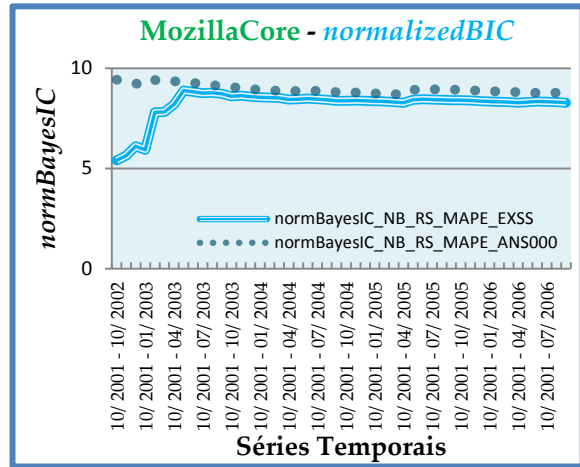
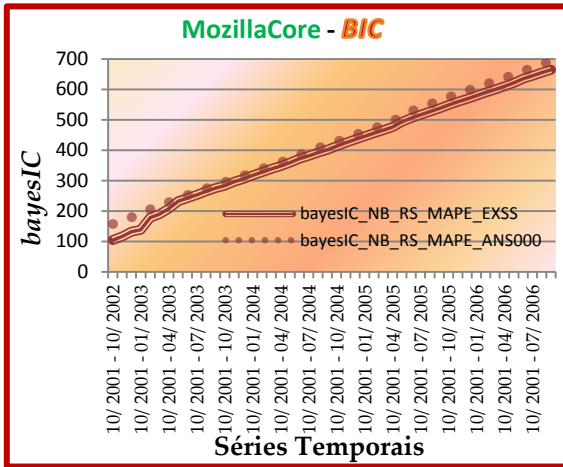
Apêndice XXVI - Figuras: Software *Mozilla Core* - Métricas de Ajuste — segundo os critérios *NormBIC*, *RSQUARE* e *MAPE*



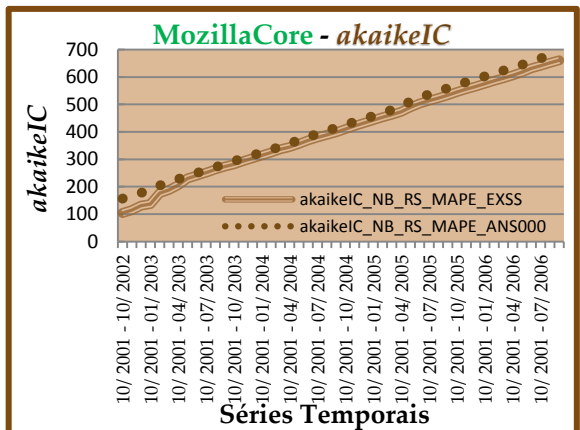
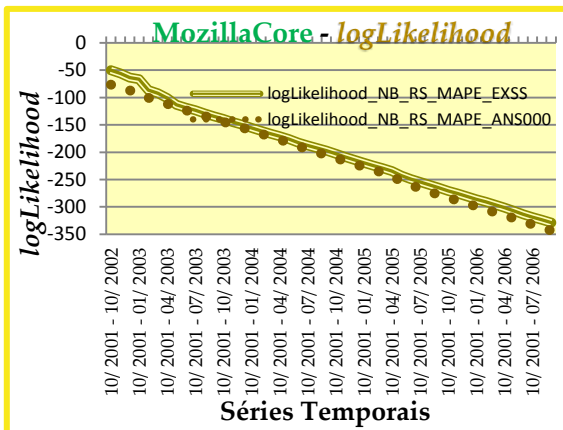
Como se pode observar no gráfico a evolução da métrica de ajuste raiz quadrada (RSQUARE - rSqr) em cada um dos modelos selecionados comprova que o modelo EXSS apresenta novamente valores mais favoráveis. Pela consulta das respetivas tabelas de *Excel* podemos confirmar que o modelo EXSS apresenta uma média de valores (0,379) superior ao modelo ANS000 (0,023), o que torna **mais favorável**. O modelo EXSS, no período de 10/2001 a 3/2003, atinge valores muito satisfatórios, mas a partir dessa data apresenta uma tendência decrescente até 5/2003, mantendo-se depois monótona. O modelo ANS000 exibe uma tendência monótona com valores muito baixos (entre 0 e 0,084).

Como se pode analisar no gráfico a métrica de ajuste raiz quadrada estacionária em cada um dos modelos indica que os dois modelos de previsão exibem um comportamento muito distinto. Pela consulta das respetivas tabelas de *Excel* podemos verificar que o modelo EXSS apresenta uma média de valores (0,746) **muito distante** da média associada ao modelo ANS010 (0,023). O modelo EXSS, no período de 12/2002 a 1/2003, apresenta uma tendência crescente e a partir dessa data atinge valores muito satisfatórios (entre 0,7 e 0,8, aproximadamente). O modelo ANS000 exibe uma tendência monótona com valores muito baixos (entre 0 e 0,084).

Apêndice XXVII - Figuras: Software *Mozilla Core* - Métricas de Critérios de Informação – segundo os critérios *NormBIC*, *RSQUARE* e *MAPE*



Como se pode observar no gráfico os critérios de informação Bayesianos (comum e o normalizado) em cada um dos modelos selecionados indica que ambos apresentam geralmente um comportamento idêntico, com valores muito próximos. Pela consulta respetivas tabelas de *Excel* podemos afirmar que o modelo EXSS apresenta uma média de valores (442,433) inferior ao modelo ANS000 (460,608) no 1º critério, e exibe uma média de valores (8,450) inferior também ao modelo ANS000 (8,909), o que torna mais admissível, apesar dos valores serem muito próximos.

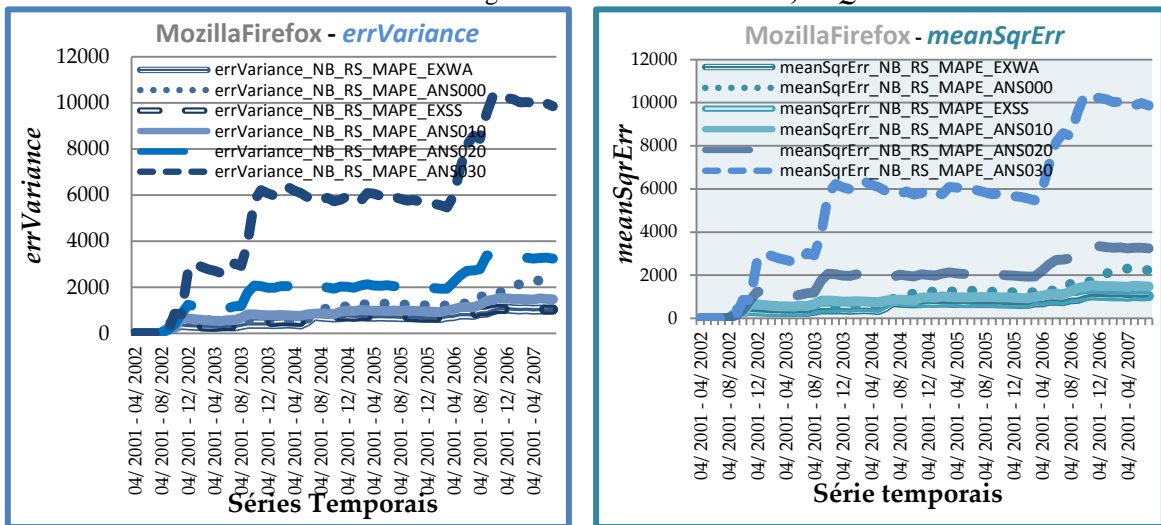


Analisando a métrica *log Likelihood* em cada um dos modelos percebe-se que apresentam um comportamento idêntico (tendência linear decrescente), com valores muito próximos. Pela consulta das tabelas de *Excel* podemos confirmar que o modelo **EXSS** exibe uma média de valores (-217,592) superior ao modelo ANS000 (-226,679), o que torna **mais**

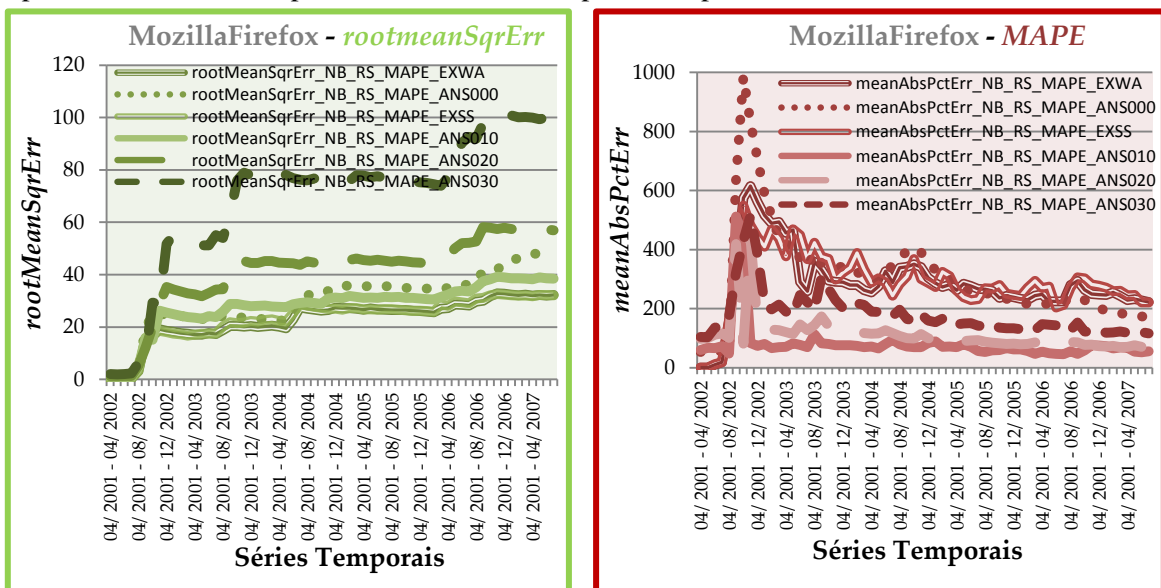
O critério de informação *akaike* em cada um dos modelos explica que os dois modelos de previsão exibem novamente um comportamento conforme (tendência linear crescente), com valores muito próximos. Nas tabelas verifica-se que o modelo **EXSS** dispõe de uma média de valores (439,184) inferior ao modelo ANS000 (457,359), o que torna **mais favorável**.

favorável.

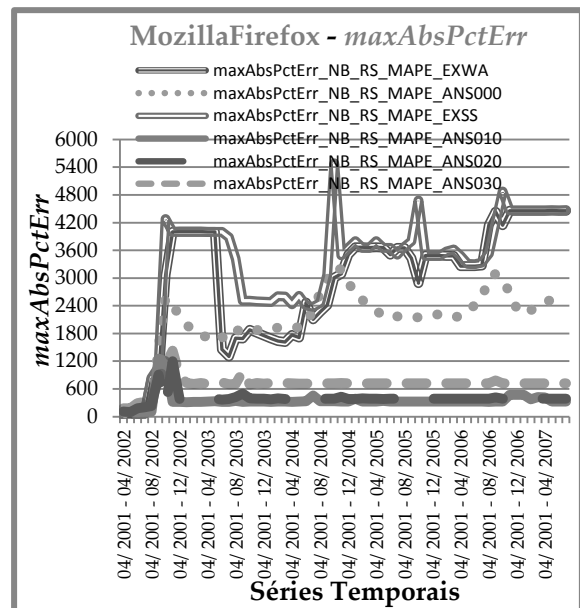
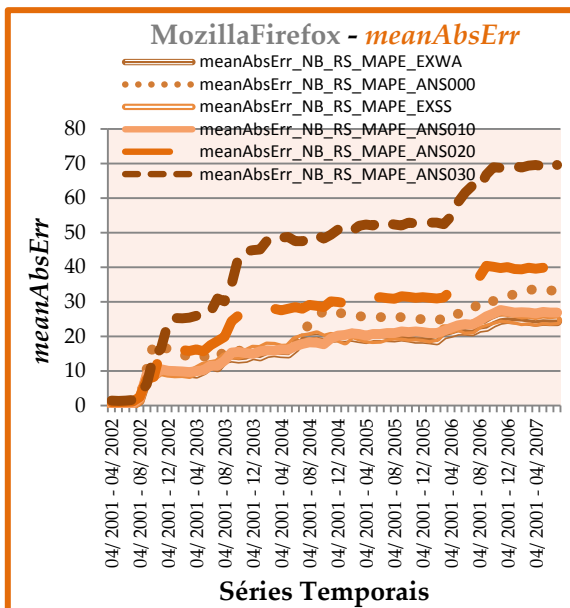
Apêndice XXVIII – Figuras: Software Mozilla Firefox - Métricas de Erro – segundo os critérios *NormBIC*, *RSQUARE* e *MAPE*



Pela análise das métricas de variância e de erro quadrático médio em cada um dos modelos selecionados pode observar-se que todos à exceção do modelo ANS030 apresentam um comportamento muito semelhante. Pela consulta das respetivas tabelas podemos confirmar que o modelo **EXWA** exibe uma média de valores (596,312) inferior ao modelo mais próximo, EXSS (621,495), o que torna mais favorável. Os modelos ARIMA são penalizados pelo facto de apresentarem valores superiores. Os modelos de previsão apresentam uma tendência.

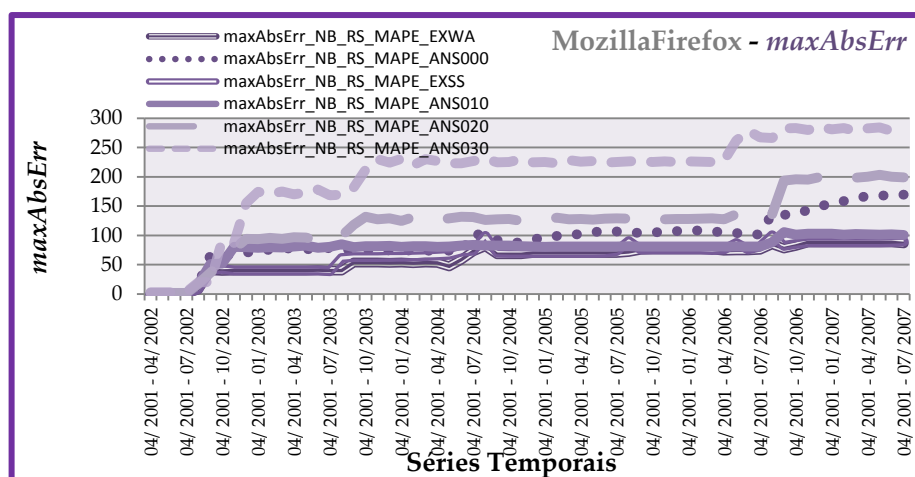


A análise das métricas de raiz quadrada do erro quadrático médio e do erro médio percentual absoluto (MAPE - meanAbsPctErr) em cada um dos modelos mostra que os mesmos exibem novamente um comportamento semelhante, isto é, na primeira métrica apresentam uma tendência crescente e na segunda métrica exibem uma tendência crescente no período de 04/2001 a 11/2002, que depois se converte numa tendência monótona. Pela consulta das respetivas tabelas de *Excel* podemos confirmar que o modelo **EXWA** exibe uma média de valores (23,012) inferior ao modelo mais imediato, EXSS (23,541), na métrica raiz quadrada do erro quadrático médio, mas obtém uma média de valores (285,252) inferior, embora aproximada, ao modelo EXSS (290,747) na métrica do erro médio percentual absoluto, o que torna um pouco melhor.



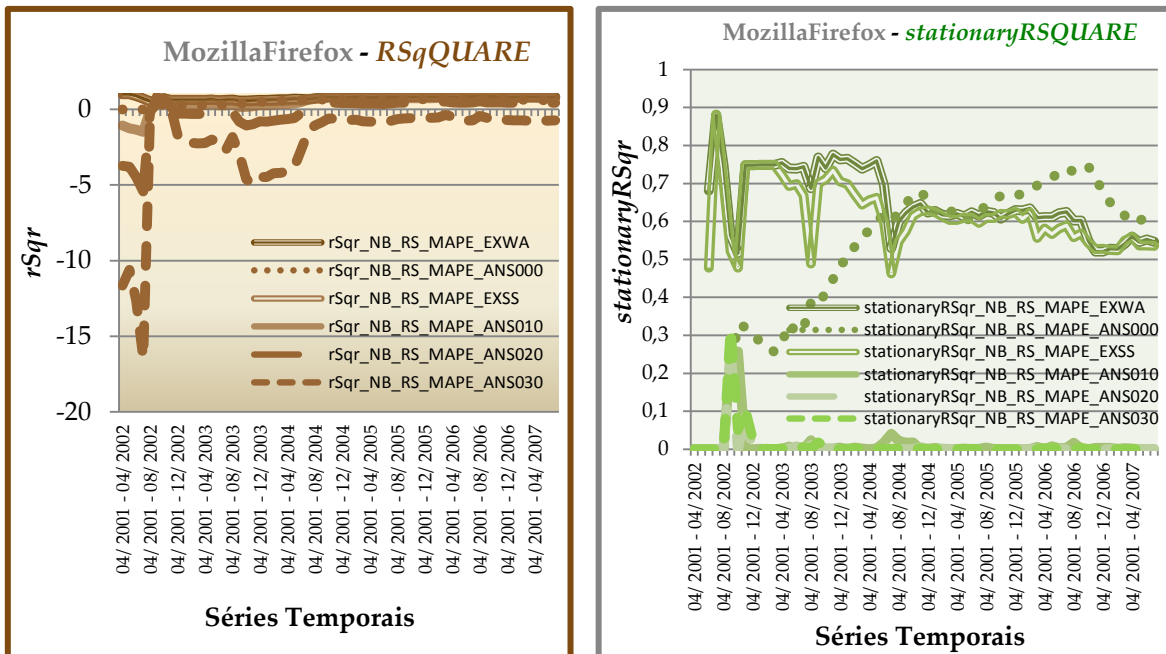
Como se pode verificar no gráfico a análise da métrica do erro médio absoluto em cada um dos modelos selecionados comprova que os modelos em causa apresentam geralmente uma tendência crescente. Pela consulta das respetivas tabelas de *Excel* podemos confirmar que o modelo **EXWA** apresenta uma média de valores (16,520) inferior aos restantes modelos, mas próximo do modelo EXSS (17,029).

Como se pode confirmar no gráfico a análise da métrica do erro médio percentual absoluto em cada um dos modelos selecionados atesta que os modelos ANS010, ANS020 e ANS030 apresentam um comportamento quase estacionário (com valores próximos), ao contrário dos restantes que exibem um comportamento muito irregular. Pela consulta das respetivas tabelas de *Excel* podemos confirmar que o modelo **ANS010** conseguiu uma média de valores (349,785) inferior aos restantes modelos.



Como se pode validar no gráfico a análise da métrica de erro máximo absoluto em cada um dos modelos selecionados demonstra que o modelo EXWA apresenta mais vez valores mais favoráveis. Pela consulta das respetivas tabelas de *Excel* podemos comprovar que o modelo **EXWA** apresenta uma média de valores (60,744) inferior aos restantes modelos, mas próximo do modelo EXSS (64,412), o que torna um pouco **mais satisfatório**. Todos os modelos em causa apresentam uma tendência crescente.

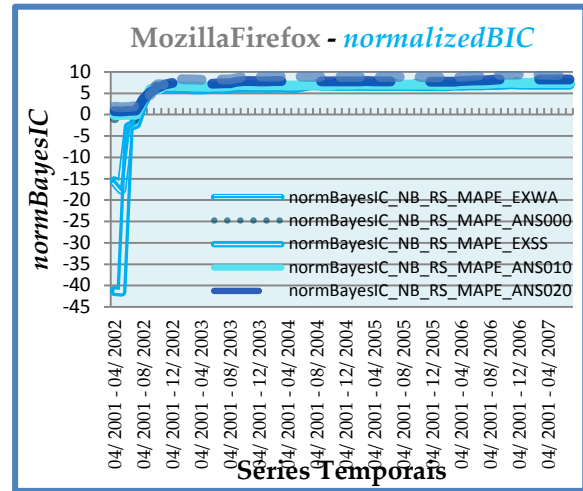
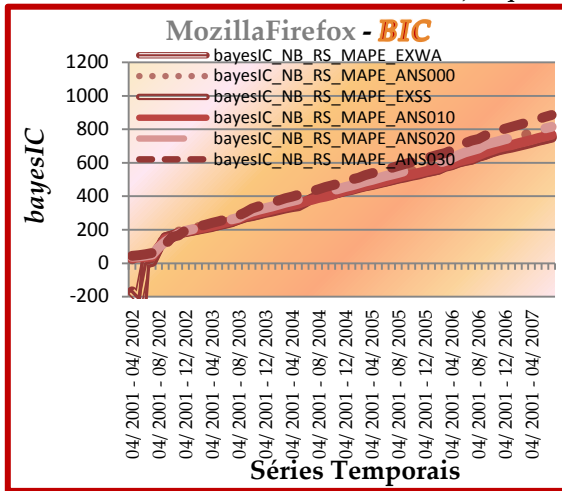
Apêndice XXIX - Figuras: *Software Mozilla Firefox* - Métricas de Ajuste — segundo os critérios *NormBIC*, *RSQUARE* e *MAPE*



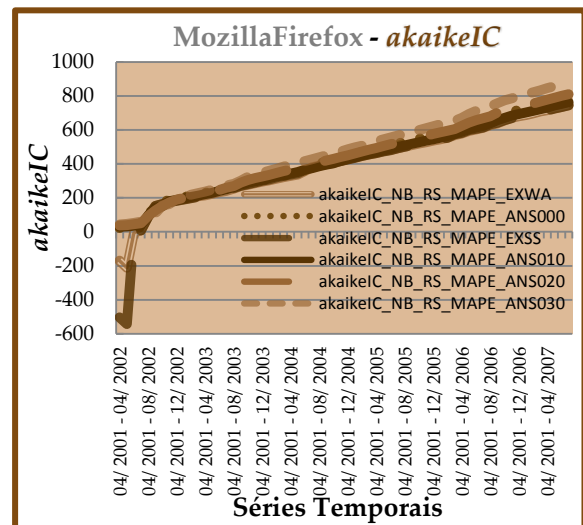
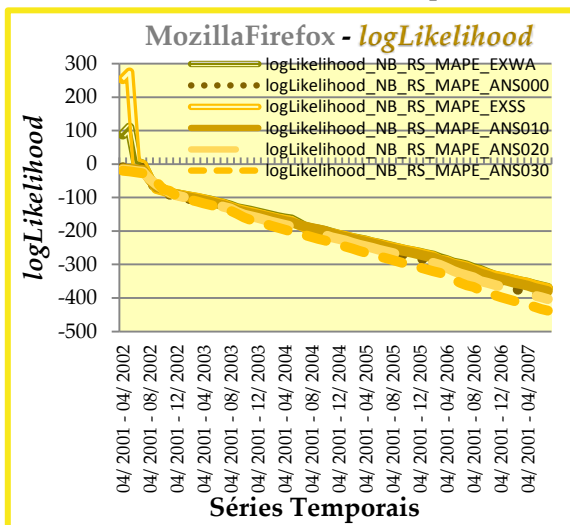
Como se pode observar no gráfico a análise da métrica de ajuste raiz quadrada ($RSQUARE - rSqr$) em cada um dos modelos seleccionados mostra que todos os modelos de previsão, à exceção do modelo ANS030 exibem novamente um comportamento idêntico. Pela consulta das respetivas tabelas de *Excel* podemos confirmar que o modelo **EXWA** apresenta uma média de valores (0,751) superior aos restantes modelos, mas muito próximo do modelo EXSS (0,726), o que torna **mais favorável**.

A métrica de ajuste raiz quadrada estacionária em cada um dos modelos indica que os modelos de previsão exibem um comportamento irregular. Pela consulta das respetivas tabelas de *Excel* podemos verificar que o modelo EXWA exhibe a média de valores (0,651) maior, muito próxima da média associada ao modelo EXSS (0,617).

Apêndice XXX - Figuras: Software Mozilla Firefox - Métricas de Critérios de Informação segundo os critérios *Normbic*, *RSquare* e *MAPE*



Como se pode observar nos figuras os critérios de informação Bayesianos (comum e o normalizado) em cada um dos modelos elegidos indica que os modelos de previsão apresentam novamente um comportamento idêntico (tendência crescente na 1ª métrica e na 2ª métrica uma tendência crescente de 4/2001 a 11/2012 e quase estacionária a partir dessa data), com valores muito próximos. Pela consulta respetivas tabelas de *Excel* podemos afirmar que o modelo **EXSS** apresenta uma média de valores (398,117) inferior aos outros modelos, mas próxima do modelo EXWA (409,192) no 1º critério. O modelo **EXSS** volta a exibir uma média de valores (4,768) inferior aos restantes modelos, mas próximo do modelo EXWA (5,585).



Analisando a métrica *log Likelihood* em cada um dos modelos percebe-se que apresentam um comportamento idêntico (tendência decrescente), com valores muito próximos. Pela consulta das tabelas de *Excel* podemos confirmar que o modelo **EXSS** exibe uma média de valores (-195,367) superior aos restantes modelos, mas próximo do modelo **EXWA** (-199,059), o que torna ligeiramente **mais favorável**.

O critério de informação akaike em cada um dos modelos explica que os modelos de previsão exibem novamente um comportamento conforme (tendência crescente), com valores muito próximos. Nas tabelas de *Excel* verifica-se que o modelo **EXSS** dispõe de uma média de valores (394,735) inferior aos outros modelos, mas próximo do modelo EXWA (404,118), o que torna um pouco **mais satisfatório**.