FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Gestão da Capacidade nas Aplicações da Direcção de Sistemas de Informação da Sonae

Paulo Alexandre Rodrigues Martins

Mestrado Integrado em Engenharia Informática e Computação

Orientador: Prof. Dr. Jorge Manuel Pinho de Sousa

26 de Julho de 2010

Gestão da Capacidade nas Aplicações da Direcção de Sistemas de Informação da Sonae

Paulo Alexandre Rodrigues Martins

Mestrado Integrado em Engenharia Informática e Computação

Aprovado em provas públicas pelo Júri:

Presidente: Luís Paulo Gonçalves dos Reis (Professor Auxiliar)

Vogal Externo: Feliz Alberto Ribeiro Gouveia (Professor Associado)

Orientador: Jorge Manuel Pinho de Sousa (Professor Associado)

---

26 de Julho de 2010

Resumo

A Gestão da Capacidade é cada vez mais uma preocupação das empresas e organizações que dependem fortemente de tecnologias de informação, e em que estas suportam o seu modelo de negócio. A Sonae, líder nacional do mercado do retalho alimentar, depende cada vez mais de infra-estruturas de tecnologias de informação que lhe possibilitem optimizar constantemente os seus processos e operações. Porém, à medida que esta dependência aumenta, também aumenta a probabilidade de ocorrência de eventos relacionados com esta infra-estrutura que possam causar graves prejuízos financeiros ou de qualquer outra natureza. Neste sentido, a Sonae delineou um plano de acção com o intuito de minorar a probabilidade de ocorrência destas situações.

Como parte deste plano, a Sonae pretende dotar-se de uma aplicação que lhe permita obter, de uma forma simples e eficiente, dados relacionados com os níveis de utilização dos vários componentes da sua infra-estrutura de tecnologias de informação. Pretende-se que esta aplicação analise a evolução passada da utilização dos recursos – variáveis de capacidade –, traduzindo estes valores para uma expressão em função das variáveis de negócio, mais particularmente do número de lojas existentes. Esta aplicação terá dois objectivos principais: a realização de estimativas dos valores futuros da utilização das várias variáveis de capacidade, e a avaliação da escalabilidade da infra-estrutura.

Ao possibilitar a realização de estimativas dos valores futuros da utilização das várias variáveis de capacidade, a aplicação permitirá antecipar a detecção de eventuais carências da infra-estrutura. Esta detecção antecipada facilita o processo de aquisição de novos componentes, permitindo a realização do processo negocial sem a pressão da urgência, o que se traduz, na generalidade, na obtenção de melhores preços e consequente redução da despesa. A avaliação da escalabilidade é também muito importante quando se pretende optimizar os processos e reduzir custos. Ao avaliar os mecanismos responsáveis pela evolução das variáveis de capacidade, torna-se possível identificar antecipadamente os limites impostos pela infra-estrutura tecnológica, o que permite uma melhor avaliação e uma tomada de decisão mais ponderada e informada.

A aplicação desenvolvida, que se apresenta como o produto final do projecto em que se insere a presente dissertação, permite responder a estas questões. Esta ferramenta possibilita a visualização gráfica da evolução das variáveis de capacidade, assim como da previsão dos seus valores futuros, possuindo ainda funcionalidades secundárias como a possibilidade de parametrizar as variáveis de input (o que aumenta a versatilidade da aplicação), ou a definição dos períodos de análise.

Este documento descreve os processos utilizados com vista ao desenvolvimento da referida aplicação, sendo também apresentado um estudo de algumas metodologias utilizadas frequentemente em projectos e aplicações que lidam com esta problemática, identificando-se as suas vantagens e desvantagens, sempre que estas se afigurem relevantes para a situação em análise.

Abstract

Capacity Management is a growing concern for companies and organizations that rely heavily on information technology, especially when those are used to support their business model. Sonae, food retail market leader in Portugal, is becoming increasingly dependent of their IT infrastructure, which enables the company to constantly improve and optimize its operational processes. But just as this dependency increases, so does the probability of occurrence of events related with it that may cause serious financial loss. With this in mind, Sonae has created a plan of action that aims to reduce the probability of the occurrence of these situations.

As part of this plan, Sonae wants to equip himself with a software application, developed with the purpose of forecasting the utilization of capacity resources and thus preventing the occurrence of these situations, through the use of techniques that aim to identify in advance events that could pose potential problems. This application should be capable of analyzing the utilization levels, observed in the past, of the resources being considered, and translate them into an expression as a function of business terms, more specifically as a function of the number of existing stores. The application has two main goals: forecasting future utilization levels of each capacity resource, and evaluating the scalability of the IT infrastructure.

By allowing the forecasting of future utilization levels for each of the capacity resources, the software application will enable an anticipated detection of possible weakness in the infrastructure. This anticipated detection will also ease the process of acquisition of new IT components when they are required by removing the urgency factor from the negotiations, which translates, in general, to better/lower prices and consequent expenditure reduction. Scalability evaluation is also very important when process optimization and cost reduction are goals to achieve. Evaluating the mechanisms that drive the evolution of the capacity variables enables the identification of the technological limits of the infrastructure, which allows for a better and more informed decision making process.

The developed software application, which could be presented as the final product of the project that originated this dissertation, makes possible to answer these problems. This tool enables the graphical visualization of the evolution of the capacity resources, as well as of their forecasted values. Secondary features of the software application include the possibility of parameterization of the input variables (which improves the application versatility) and the definition of the time period in which to conduct the analysis.

This document describes the processes that were used in order to implement the referred application. A study of several methodologies commonly used in projects that deal with this problem is also presented here. Additionally, advantages and disadvantages of these methods are identified, where they appear to be relevant.

Agradecimentos

Ao meu orientador, Professor Jorge Pinho de Sousa, pelo seu apoio, orientação e conselhos prestados, que contribuíram de forma benéfica para a elaboração desta dissertação.

Ao Pedro Souto, investigador no INESC Porto, e ao Professor António Miguel Gomes, pela disponibilidade que demonstraram e pelos conselhos que disponibilizaram, muitos deles fundamentais para o resultado final deste projecto.

À Sonae, na figura do Jorge Preza Reis, pela oportunidade e pelo desafio oferecidos.

Ao meu orientador na Sonae, André Espírito Santo, pelo incentivo constante e pela disponibilidade que sempre demonstrou, e que contribuiu decisivamente para a elaboração desta dissertação.

A todos os colaboradores da Sonae com quem contactei diariamente, particularmente o Filipe Martins, o Nuno Ribeiro e o Flávio Oliveira, pelo companheirismo e amizade com que me acolheram durante a minha estadia na organização.

Aos meus amigos, pelo apoio e amizade sincera sempre demonstrados.

Aos meus pais, por todos os sacrifícios que fizeram e que permitiram a concretização deste sonho.

A todos o meu muito sincero obrigado.

Conteúdo

Lista de Figuras

Capítulo 1

Introdução

O planeamento da capacidade tornou-se numa das áreas necessárias a qualquer organização que dependa intensamente de tecnologias da informação (TI) para o funcionamento eficiente do seu modelo de negócio. Manter uma elevada disponibilidade das aplicações e serviços é um requisito crítico de qualquer empresa ou organização, mas devido à crescente complexidade das infra-estruturas de TI existentes, bem como do modelo de negócio que suportam, este requisito torna-se cada vez mais difícil de alcançar. Os administradores de redes, por exemplo, enfrentam desafios cada vez maiores para garantir o bom funcionamento dos sistemas, quando confrontados com factores como a falta de experiência dos colaboradores, as ferramentas e recursos insuficientes, as quebras de serviço não calendarizadas, a complexidade das tecnologias, as consolidações do negócio, ou a competitividade dos mercados. Devido ao clima financeiro e comercial actual, a necessidade de reduzir custos através de consolidações e previsões acertadas dos requisitos futuros de capacidade estão na primeira linha das medidas tomadas por várias organizações. Para grandes projectos de TI, não é incomum verificar que os custos associados a melhorias de performance, gestão dos recursos e actividades de minoração de danos consomem uma grande fatia do orçamento disponível.

Os problemas de desempenho das aplicações têm um impacto determinante e imediato na satisfação dos clientes. Uma pequena falha súbita pode afectar um grande número de indivíduos, pode causar atrasos em projectos e, em última instância, causar graves prejuízos financeiros para a organização. Não são raras as situações em que uma nova peça de hardware é adicionada à infra-estrutura para corrigir alguns problemas de performance, sem que os responsáveis compreendam claramente onde reside o verdadeiro problema. É para responder a desafios como este que se têm vindo a desenvolver e a aplicar técnicas e metodologias adaptadas a esta realidade, e que, em última instância, deram origem ao aparecimento da área processual da Gestão da Capacidade.

1.1 Contexto/Enquadramento

A Sonae é uma empresa de retalho, com uma forte posição no mercado nacional. A sua actividade assenta em torno de dois negócios *core*: o retalho alimentar e o retalho especializado.

A Sonae MC é responsável pela área de retalho alimentar da Sonae e é hoje uma referência no mercado, após ter iniciado uma verdadeira revolução nos hábitos de consumo e no panorama comercial português, com a implementação do primeiro hipermercado em Portugal, em 1985. A Sonae MC é líder de mercado nacional, no retalho alimentar, com um conjunto de formatos distintos que oferecem uma variada gama de produtos. O seu portfólio de insígnias engloba marcas como a Área Saúde (parafarmácias), Bom Bocado (restaurantes), Book.it (livraria/papelaria), Continente (hipermercados) e Modelo (supermercados).

A área de retalho especializado da Sonae está sob a alçada da Sonae SR. A Sonae SR detém um universo de insígnias com posições de referência nos respectivos segmentos de mercado. A oferta é bastante diferenciada: Loop (calçado), Modalfa (vestuário), Zippy (vestuário de bebé e criança), SportZone (equipamento e vestuário desportivo), Vobis (equipamento informático), Worten (electrodomésticos e electrónica de consumo) e Worten Mobile (telecomunicações móveis). Ao longo dos últimos anos, estas empresas têm vindo a dar corpo a uma estratégia de geração de valor assente na conjugação de sustentados ritmos de crescimento com um forte investimento na proposta de valor de cada uma das suas insígnias, suportando-se na motivação e qualidade dos seus colaboradores e no estabelecimento de parcerias sustentadas com os seus fornecedores.

Devido à dimensão e complexidade crescentes do seu modelo de negócio, a Sonae tem necessidade de uma infra-estrutura tecnológica adequada e que seja capaz de assegurar o funcionamento eficiente das suas actividades e transacções, nomeadamente garantindo uma boa comunicação inter e intra-departamental, com os clientes (B2C) e com os seus fornecedores (B2B).

Esta infra-estrutura é, como seria de esperar, um complexo sistema onde recursos como espaço de armazenamento em disco, poder de processamento ou largura de banda se conjugam para responder às necessidades de um largo conjunto de aplicações necessárias para o bom funcionamento da organização. Estes recursos podem ser suficientes no momento actual, mas revelarem-se escassos a curto/médio-prazo no caso de um eventual crescimento da empresa. Por outro lado, a existência de recursos em quantidades que ultrapassem largamente o necessário também se pretende evitar, principalmente aqueles que requerem investimentos mais avultados, como o poder de processamento.

Com este objectivo, a Sonae definiu a metodologia ITIL como uma das *frameworks* a adoptar para melhorar os seus processos e serviços. Neste sentido, a empresa pretende dotar o agente de decisão com uma ferramenta que permita auxiliar, agilizar e suportar o processo de Gestão da Capacidade, sob a forma de um sistema de apoio à decisão que estime, com base no plano de crescimento da empresa e na utilização passada dos recursos, qual a capacidade necessária para garantir o cumprimento dos acordos de nível de serviço estabelecidos, em vários horizontes temporais no futuro. Esta informação poderá ser vital, uma vez que possibilita a supressão atempada de eventuais carências detectadas. Adicionalmente, a detecção antecipada destes problemas permite também que a eventual aquisição de novos componentes possa ser efectuada com um custo inferior, uma vez que a pressão exercida pelo tempo é menor sobre o agente que conduz a negociação. Podemos, por isso, definir duas áreas que se pretendem melhorar com este projecto: a previsão dos valores futuros dos níveis de utilização dos vários recursos de capacidade, e a avaliação da escalabilidade do sistema actual.

1.2 Objectivos

Esta dissertação tem como objectivo principal o desenvolvimento de uma ferramenta que permita à Sonae, mais particularmente ao(s) colaborador(es) responsável(eis) pela área de infra-estruturas de tecnologias de informação, a estimação dos níveis de utilização de alguns recursos de capacidade num determinado período no futuro.

Esta ferramenta deverá ser capaz de suportar a análise de vários recursos de capacidade, e deverá permitir o estudo de várias das aplicações utilizadas pela Sonae, i.e. o estudo das necessidades em termos das variáveis de capacidade de cada aplicação. Adicionalmente, a ferramenta a desenvolver deverá possuir funcionalidades que lhe permitam, para além de visualizar graficamente a evolução das necessidades das aplicações, a introdução em tempo real das variáveis utilizadas como parâmetros do modelo para que melhor possam ser compreendidos os resultados obtidos, e para que novos cenários possam ser analisados (e.g. se em vez de abrirem 5 lojas Continente numa determinada data abrirem 10). A ferramenta deverá ainda apresentar uma interface simples mas agradável, e possuir características que tornem a sua utilização o mais intuitiva possível.

1.3 Estrutura da Dissertação

O resto do documento está organizado da seguinte forma: no capítulo 2 será apresentada a metodologia ITIL, bem como um breve apanhado de alguns trabalhos relacionados com o presente projecto, e uma análise a alguns dos vários métodos de previsão existentes; o capítulo 3 detalha os processos de modelação realizados para as diferentes variáveis de capacidade; no capítulo 4 é apresentada a aplicação desenvolvida, que complementa o trabalho apresentado neste documento; por último, no capítulo 5 são expostas as conclusões retiradas do trabalho realizado e sugeridos alguns melhoramentos futuros.

Capítulo 2

Revisão Bibliográfica

À medida que a complexidade dos sistemas de TI aumenta, a gestão do desempenho e o planeamento da capacidade tornam-se numa das áreas onde é mais complicado controlar as despesas. Novas metodologias e técnicas de modelação que expliquem os comportamentos de sistemas de grande dimensão e que ajudem a prever a sua performance futura são cada vez mais necessárias para enfrentar de forma eficiente os desafios que emergem a cada dia. Com o paradigma de arquitectura *multi-tier* a ganhar cada vez mais relevância e a tornar-se um standard da indústria para desenvolver aplicações cliente-servidor escaláveis [1][2], é cada vez mais importante projectar modelos de previsão de desempenho eficientes e precisos para este tipo de sistemas, quando estes são utilizados num ambiente de produção empresarial e sujeitos a várias e diversas cargas de trabalho.

Da mesma forma, é fundamental apostar em técnicas e metodologias proactivas, por oposição às reactivas, mais tradicionais, uma vez que a médio prazo estas se podem tornar economicamente mais vantajosas.

Para melhor compreendermos esta diferença, tomemos como exemplo um processo reactivo para gestão da capacidade e desempenho, a Gestão de Excepções. Este é uma processo utilizado para detectar, identificar e resolver problemas de capacidade e/ou desempenho. O seu modo de funcionamento consiste em receber notificações quando ocorrem violações dos limites de capacidade e desempenho pré-especificados, e imediatamente investigar e solucionar o problema [3]. O ponto-chave para o bom desempenho do processo de Gestão de Excepções reside na notificação o mais imediata possível dos problemas. Caso contrário, o problema pode desaparecer antes que seja possível aferir a sua origem, dificultando assim a sua resolução e a prevenção de recorrências futuras. O principal problema deste tipo de processos é a necessidade constante de um responsável para corrigir os problemas, e o facto de a organização estar menos preparada para enfrentar eventuais desafios.

Por outro lado, um processo como uma análise do tipo *what-if* [4] é um bom exemplo de uma metodologia proactiva. Este processo, especificamente, consiste na formulação de vários cenários alternativos (possíveis) e no estudo dos seus impactos ou, por outras palavras, na análise dos efeitos provocados por variações nos parâmetros e variáveis do sistema em estudo.

Isto permite que eventuais carências ou limitações do sistema possam ser detectadas *a priori*, facilitando o processo de resolução e/ou evitando problemas futuros.

Em suma, podemos afirmar que as metodologias e os processos reactivos são, na sua generalidade, mais simples e menos dispendiosas de implementar mas, por outro lado, a probabilidade de ocorrência de graves problemas como consequência de uma falha de capacidade é consideravelmente superior.

Este capítulo traça um retrato geral desta problemática, descrevendo a metodologia ITIL nas disciplinas que se afiguram como relevantes para o desenvolvimento do presente projecto, e fazendo uma pequena abordagem a alguns projectos semelhantes, desenvolvidos para enfrentar o problema da Gestão da Capacidade. É também apresentada neste capítulo uma análise de vários métodos de previsão, frequentemente utilizados como ferramentas pelos profissionais que lidam com esta problemática.

2.1 ITIL

A Information Technology Infrastructure Library (ITIL) [16] é uma série de documentos que compilam as melhores práticas na área da Gestão de Serviços, providenciando uma importante *framework* em que as organizações se podem basear para implementarem as mudanças necessárias ao seu modelo de negócio, com vista a tornarem-se mais eficientes. A sua aceitação tem vindo a crescer nos últimos anos, tendo já sido adoptada por centenas de organizações em todo o mundo. A primeira versão da ITIL foi criada no Reino Unido por uma agência estatal, a Central Computer and Telecomunications Agency (CCTA) – actual Office of Government Commerce -, durante o período compreendido entre 1986 e 1992. Esta versão inicial mais não era que uma série de folhetos que compilava as boas práticas documentadas em manuais da especialidade, como os IBM Technical Labs ISMA Manuals ou o Government Information Technology Infrastructure Management Method (GITIMM). A segunda versão da ITIL surgiu em 1996, e rapidamente se afirmou como uma referência na área da Gestão de Serviços. Ocorreu então um alinhamento entre a ITIL e a norma BS15000, que culminou na aparição da norma ISO/IEC20000 (Service Management Standard) e na terceira versão da ITIL, em 2007, denominada ITIL V3 – The Service Lifecycle [16].

Esta nova versão é composta por cinco volumes, quatro dos quais definem um igual número de estados-chave de um serviço (Estratégia, Desenho, Transição e Operações) e um último que se foca na melhoria contínua dos serviços [17].

O volume da Estratégia concentra-se em garantir que uma estratégia para o serviço é definida, mantida e implementada. Introduz novos conceitos como criação de valor, definição de mercados e espaço de solução. Foca-se ainda na tomada prática de decisões, baseada no conhecimento dos recursos, estruturas e aspectos financeiros dos serviços, com o objectivo de aumentar a viabilidade económica destes e prolongar o seu ciclo de vida.

O Desenho do Serviço tem como objectivo o estabelecimento de planos que convertam a estratégia em realidade. Aborda tópicos como Gestão da Disponibilidade, Gestão da Capacidade, Gestão da Continuidade e Gestão dos Níveis de Serviço, já presentes na segunda versão da ITIL. Este volume introduz uma nova área processual, a Gestão de Fornecedores, e conceitos como garantia e utilidade do serviço, considerados fundamentais pelos clientes.

O terceiro volume, Transição, preocupa-se com a eficiência da passagem dos projectos para a fase operacional. Atribui responsabilidades a processos como Gestão de Recursos e

Configurações, Validação e Teste. Neste volume podem ainda ser encontrados alguns exemplos de modelos organizacionais para suportar as transições, e linhas mestras sobre como reduzir variações da entrega.

Por último, o capítulo sobre Operações procura garantir que existem processos robustos que acompanham todo o ciclo de vida dos serviços, e que garantem que estes se mantêm activos e estáveis.

Como referido, o presente projecto insere-se no âmbito da área processual Gestão da Capacidade. A ITIL define três sub-processos para esta área processual: Gestão da Capacidade do Negócio, Gestão da Capacidade do Serviço e Gestão da Capacidade dos Recursos [18]. O objectivo destes três sub-processos é descrever o que precisa de ser feito nesta área, porém sem especificar a forma como estes devem ser implementados, nem quem os deve realizar. Em vez disso, a ITIL providencia uma *framework* para o processo, colocando o foco da sua atenção na necessidade de combinar as pessoas, os processos e os recursos existentes na organização para que a implementação de um serviço de Gestão da Capacidade possa ser efectuada com sucesso. Por esta razão, os sub-processos desta área processual devem ser traduzidos para procedimentos adaptados à realidade da organização, para que estes possam ser assimilados e adoptados pelos colaboradores que irão interagir com eles. Estes colaboradores, por sua vez, deverão receber formação para que possam efectuar de forma eficiente as suas novos tarefas. Por último, alguns produtos poderão/deverão ser instalados para fazer face aos novos desafios da organização, produtos estes que permitirão a automatização de algumas tarefas de trabalho intensivo [18]. É precisamente neste último ponto que reside o interesse da Sonae em munir-se de uma ferramenta que lhe possibilite automatizar algumas das tarefas relacionadas com a problemática da Gestão da Capacidade. Porém, para efectuar a implementação de uma ferramenta deste género é necessária, para além do conhecimento dos problemas a enfrentar, a compreensão de todos os processos com que esta irá interagir, bem como da realidade da organização.

O sub-processo Gestão da Capacidade do Negócio é responsável por garantir que os requisitos futuros do negócio, em termos de Tecnologias da Informação, são considerados, planeados e implementados atempadamente. É o mais proactivo dos três sub-processos da Gestão da Capacidade definidos pela ITIL, mas é também o mais incerto, uma vez que o futuro é altamente susceptível à mudança. Devido a esta última razão, este é o sub-processo que, geralmente, recebe menos atenção por parte das organizações e, por consequência, o menos maturado. A Gestão da Capacidade do Negócio compreende análises de tendências, previsões, modelação, prototipagem, escalabilidade e documentação dos requisitos futuros do negócio. A análise de tendências tem uma forte correlação com os eventos registados no passado; esta informação é particularmente útil quando ocorrem periodicamente mudanças na dimensão da organização, pois permite antecipar as necessidades que advirão desse evento. A função das previsões consiste em traduzir os novos requisitos do negócio para requisitos de TI, que poderão ser aumentos ou reduções. A modelação e a prototipagem aplicadas à Gestão da Capacidade do Negócio permitem a expressão em termos numéricos do ambiente organizacional, possibilitando a realização de análises de cenários e análises de sensibilidade para determinar os efeitos de potenciais mudanças. Todas estas funções despoletam a necessidade de criar um repositório para a informação numérica, denominado Base de Dados de Capacidade.

A Gestão da Capacidade do Serviço tem como objectivo a compreensão dos recursos dos serviços, padrões de trabalho e intervalos temporais de elevada e de reduzida utilização, como forma de garantir que os objectivos descritos nos Acordos de Nível de Serviço (*Service Level Agreements* - SLA) são cumpridos. Este sub-processo inclui tarefas como monitorização, análise, afinação e documentação do desempenho do serviço, definição de perfis de utilização dos serviços, e gestão da procura destes últimos. A monitorização é efectuada, normalmente, ao nível da aplicação, e inclui a disponibilidade da aplicação, o seu tempo de resposta e estatísticas decorrentes da sua utilização. A análise consiste numa visualização das TI ao nível da aplicação, de uma forma que permita compreender não só quais as transacções efectuadas pelas aplicações, mas também qual o caminho crítico destas, para que seja possível afinar o processo com vista a melhorar o seu desempenho. Periodicamente, deverá ser elaborado um relatório do desempenho do serviço, para que seja possível confirmar que os níveis de serviço acordados estão a ser cumpridos.

Por último, o objectivo da Gestão da Capacidade dos Recursos é identificar e compreender a utilização de cada um dos componentes da infra-estrutura de TI, de forma a garantir a optimização da utilização dos recursos de hardware e de software existentes. Este é o sub-processo mais familiar para as pessoas, uma vez que estas percebem claramente que cada componente tem uma capacidade finita, e que existe um potencial problema de desempenho à medida que a utilização dos recursos se aproxima do seu limite de capacidade. As tarefas da Gestão da Capacidade dos Recursos incluem a monitorização, análise, afinação e documentação da utilização dos componentes, e a definição de perfis de utilização destes. Estas tarefas são semelhantes às descritas pela Gestão da Capacidade do Serviço, mas devem ser realizadas ao nível dos recursos, e não ao nível dos serviços.

A aplicação destes sub-processos deverá levar à produção de um Plano de Capacidade para a organização. Este é o *deliverable* mais importante da área processual da Gestão da Capacidade, pois este descreve a infra-estrutura actual e o seu ambiente, e sumariza todas as mudanças que deverão ser efectuadas para responder às novas necessidades e desafios da organização. Este documento deverá conter as três vertentes da Gestão da Capacidade: uma vista ao nível do negócio, outra ao nível dos serviços, e outra ao nível dos recursos. Este plano deverá ser revisto e alterado periodicamente.

O projecto em que se insere esta dissertação foca-se no sub-processo da Gestão da Capacidade dos Recursos. Com este projecto procura-se não só analisar o estado actual da infra-estrutura tecnológica da Sonae naquilo que concerne a este sub-processo, mas também implementar um produto que permita agilizar e automatizar algumas das tarefas que este define, e que possibilite a realização de estimativas relativamente ao desempenho futuro do sistema.

2.2 Projectos relacionados

A avaliação da performance e o planeamento da capacidade de sistemas de hardware e de software é, tal como já foi referido, um ponto crítico no processo de desenho dos sistemas. Por isso, como seria expectável, existem inúmeras técnicas de planeamento da capacidade propostas e implementadas em várias aplicações.

De entre estas técnicas, a teoria das filas (*queueing theory*) é provavelmente a metodologia mais utilizada para modelar o comportamento de um sistema e responder a questões relacionadas com a capacidade [5][6]. A teoria das filas é uma teoria matemática que surge como um ramo da teoria das probabilidades aplicadas, sendo conhecida por vários outros nomes como teoria do tráfego, teoria da congestão, teoria do serviço em massa ou teoria dos sistemas estocásticos de serviços [7][8]. A modelização de sistemas simples, como servidores HTTP, já foi estudada exaustivamente [9][10].

A utilização de métodos estatísticos como ferramenta para o planeamento da capacidade foi proposto no início da década de 1980 [11][12], mas estas abordagens focavam-se numa simples máquina/cluster que era muito mais simples do que os actuais sistemas *multi-tier* de larga escala. Métodos estatísticos mais recentes estão ser cada vez mais utilizados para efectuar análises e previsões de desempenho de sistemas. Várias técnicas de regressão linear são utilizadas, por exemplo, para calcular os tempos médios de serviço de aplicações num servidor *single-threaded* [13], correlacionando depois estes tempos de serviço com os dados obtidos através do pacote open-source *Application Response Measurement* para que a performance futura do sistema pudesse ser estimada. Noutros projectos [14][15], os autores focaram-se em modelos de desempenho de transacções de negócio. Estes partiram da assumpção de que o tempo de resposta de uma transacção dependia fundamentalmente do tempo de serviço desta e não no seu tempo de espera, e utilizaram o tempo de resposta para fazer uma aproximação à procura do respectivo serviço. Os autores utilizaram então a regressão linear para identificar problemas de desempenho em cargas de trabalho passadas e detectar as suas causas.

2.3 Técnicas e Métodos de Previsão

Nesta secção são apresentadas algumas das técnicas e métodos de previsão utilizados frequentemente para lidar com a problemática da Gestão da Capacidade.

2.3.1 Moving Average

A utilização da *Moving Average* é muito frequente no estudo de séries temporais [21]. Isto deve-se, fundamentalmente, à sua extrema facilidade de utilização quando a série temporal em estudo apresenta um comportamento estacionário. Diz-se que uma série apresenta um comportamento estacionário quando as suas propriedades estatísticas (média, variância, auto-correlação, …) não dependem do tempo. Este tipo de séries é definido pelo modelo de regressão

$$y_t = \mu_t + \varepsilon_t \,, \tag{2.1}$$

onde y_t é o valor observado no instante t, μ_t é o termo de intersecção, e que indica o ponto onde a linha da regressão intercepta o eixo dos yy, e ε_t é uma perturbação aleatória, também conhecida como residual ou erro. Este modelo assume ainda que os residuais são independentes e normalmente distribuídos, com média 0 e variância σ_ε^2. Adicionalmente, assume-se que os residuais são homoescedásticos, ou seja, com uma variância constante em relação ao tempo [27].

O algoritmo *Moving Average* consiste em, dado um conjunto ordenado de m números, calcular a média do primeiro subconjunto de n elementos (para $n < m$). O processo repete-se para todos os subconjuntos de tamanho m, até todos os elementos terem sido considerados. Tipicamente, o cálculo das médias dos subconjuntos é efectuado da forma tradicional,

contribuindo cada elemento de forma idêntica para o resultado final, mas estas podem ser calculadas atribuindo pesos diferentes a cada um destes elementos. Há várias alternativas para esta formulação, sendo uma das mais populares a *Weighted Moving Average*, em que cada elemento é associado a um peso que decresce linearmente à medida que recuamos no tempo, dando assim uma maior importância aos elementos mais recentes. Após o cálculo destas médias, se traçarmos um gráfico com os valores obtidos, a linha que une os pontos é a linha que representa *Moving Average*. A *Moving Average* é, portanto, um conjunto de valores, por oposição a um valor único com que é frequentemente associada. Assim, podemos definir uma função para a série que representa a *Moving Average*, na sua formulação básica, da seguinte forma:

$$MA_n(t) = \frac{(y_t + y_{t-1} + \cdots + y_{t-n+1})}{n}, \tag{2.2}$$

onde n é o número de elementos do subconjunto, $MA_n(t)$ é o valor da *Moving Average* no instante t, e y_t é o valor no instante t da série a analisar.

A *Moving Average* é geralmente utilizada na análise de séries temporais quando se pretende suavizar as flutuações que ocorrem em períodos temporais curtos, dando um maior relevo tendências e ciclos demonstrados em intervalos de tempo maiores. Exemplos deste tipo de utilização são as análises de volumes transaccionados de um determinado produto [22], ou das oscilações nos preços das acções cotadas em bolsa [23].

Quando se tenta prever o comportamento futuro de uma série, os resultados obtidos são bastante satisfatórios se esta for estacionária, pelas razões óbvias, porém estes mostram-se insuficientes quando se pretende estender a série não é estacionária e apresenta uma tendência significativa. Para tentar combater esta limitação, existe um procedimento denominado *Double Moving Average*, que calcula uma segunda *Moving Average* utilizando a primeira como input, em vez da série original, ou seja

$$MA2_n(t) = \frac{\left(MA_n(t) + MA_n(t-1) + \cdots + MA_n(t-n-1)\right)}{n}, \tag{2.3}$$

onde $M2_n(t)$ é a segunda *Moving Average*. Calcula-se, em seguida, a diferença entre as duas séries obtidas

$$a_t = 2 \cdot MA_n(t) - MA2_n(t) \tag{2.4}$$

e um factor de ajustamento para a tendência

$$b_t = \frac{2}{n-1}\left[MA_n(t) - MA2_n(t)\right], \tag{2.5}$$

obtendo-se assim a função que procura estimar o valor da série no futuro,

$$\hat{Y}_{t+h} = a_t + b_t h, \tag{2.6}$$

onde $\hat{Y}_{t+h}$ é o valor estimado para a série no instante $t + h$.

2.3.2 Exponential Smoothing

A *Exponential Smoothing,* também conhecida como *Exponential Moving Average,* baseia-se na técnica descrita na subsecção anterior, tal como o nome sugere. Esta difere no facto de os pesos de cada elemento decrescerem de forma exponencial, e não linearmente como na *Weighted Moving Average*, o que faz com que as observações mais recentes sejam

substancialmente mais importantes para o cálculo do que as mais antigas. Esta técnica está vocacionada para séries que tomem uma forma idêntica à apresentada em (2.1), tal como a *Moving Average*. A equação para o cálculo da *Exponential Smoothing* é

$$S_t = \alpha y_{t-1} + (1-\alpha) S_{t-1}\,, \tag{2.7}$$

onde S_t é o valor da *Exponential Smoothing* no instante t e α é uma constante de alisamento sujeita às restrições $0 < \alpha < 1$. Esta equação só é válida quando $t \geq 1$, por isso assume-se que $S_0 = y_0$. Para melhor compreender a natureza exponencial deste algoritmo de previsão, tomemos (2.7), substituindo S_{t-1}, da qual resulta

$$\begin{aligned} S_t &= \alpha y_{t-1} + (1-\alpha)\left[\alpha y_{t-2} + (1-\alpha) S_{t-2}\right] \\ &= \alpha y_{t-1} + \alpha(1-\alpha) y_{t-2} + (1-\alpha)^2 S_{t-2}\,. \end{aligned} \tag{2.8}$$

Se continuarmos a substituir para S_{t-2}, S_{t-3}, e assim sucessivamente, obtemos a expressão:

$$S_t = \left[\alpha \sum_{i=1}^{t-1} (1-\alpha)^{i-1} y_{t-i}\right] + (1-\alpha)^{t-1} y_0 \tag{2.9}$$

O valor de α determina as diferenças entre os pesos das várias observações ou, por outras palavras, o ritmo ao qual as observações perdem importância para o cálculo do valor final. Quanto mais elevado for o valor de α, maiores serão as diferenças entre os pesos, o que faz com que as observações mais recentes sejam muito mais importantes do que as antigas. Se o valor de α for muito baixo, as diferenças verificadas entre os pesos das observações tornam-se menores. A constante α pode ser arbitrada, mas deve haver o cuidado de seleccionar um valor que se adeqúe ao problema em mãos. Frequentemente, o valor óptimo de α é aquele que minimiza a raiz quadrada do erro quadrático médio (RMSE[1]), calculado de acordo com a fórmula

$$RMSE = \sqrt{\frac{\sum_{i=1}^{m} \left(S_i - y_i\right)^2}{m}}\,, \tag{2.10}$$

onde m é o número de observações. A raiz quadrada do erro quadrático médio é uma das mais populares medidas de erro, e tem a vantagem de dar mais atenção aos erros maiores do que àqueles que são menores [24].

Uma vez que o algoritmo *Exponential Smoothing* depende, para cada valor de t, do valor observado no instante $t-1$, surge um problema quando se pretende efectuar previsões utilizando este método, pois não existem dados sobre o futuro. Para ultrapassar esta limitação é necessário modificar (2.7), de tal forma que

$$S_{t+1} = \alpha y_{origem} + (1-\alpha) S_t\,, \tag{2.11}$$

onde y_{origem} é um valor que permanece constante, e que deve tomar o valor da última observação conhecida antes do instante que se pretende estimar. Este processo é conhecido como *bootstrapping*. Porém, tal como quando se utiliza a técnica da *Moving Average*, os

[1] Root Mean Squared Error

resultados obtidos por este algoritmo não são satisfatórios quando a série que se está a analisar apresenta uma tendência de crescimento ou decrescimento [43].

Isto mostra que este algoritmo não é suficientemente eficaz para lidar com estas situações. Esta razão levou ao aparecimento de um outro algoritmo, desenvolvido independentemente por Charles C. Holt e por Robert Brown, denominado *Double Exponential Smoothing*.

2.3.3 Double Exponential Smoothing

Em 1957, Charles C. Holt apresentou um novo modelo que tem em consideração a possível existência de tendências nas séries temporais que se pretendem analisar [25]. Paralelamente, Robert Brown encontrava-se também a estudar esta problemática e publica em 1959 um livro onde descreve um outro algoritmo para lidar com este problema [26], e que se viria a revelar uma particularização do modelo desenvolvido por Holt. Estes algoritmos destinam-se ao estudo de modelos que tomem a forma

$$y_t = \mu_t + \delta_1 t + \varepsilon_t \,, \tag{2.12}$$

em que μ_t é o termo de intersecção, $\delta_1 t$ é o termo de declive da linha de regressão, e ε_t é o residual. As assumpções tomadas em relação a (2.1) mantêm-se válidas para este modelo.

O método desenvolvido por Brown pode ser comparado ao *Double Moving Average*, apresentado na subsecção anterior. De facto, este utiliza uma segunda *Exponential Smoothing*, obtida a partir da primeira, de acordo com o par de equações seguinte:

$$S_t = \alpha y_{t-1} + (1-\alpha) S_{t-1} \,, \tag{2.13}$$

$$S_t' = \alpha S_t + (1-\alpha) S_{t-1}' \,, \tag{2.14}$$

onde S_t e S_t' são a primeira e segunda *Exponential Smoothing*, respectivamente. De forma semelhante, para efectuar previsões com este método recorre-se novamente a (2.6), em que o termo de intercepção é dado por

$$a_t = 2S_t - S_t' \tag{2.15}$$

e o termo de declive é

$$b_t = \frac{\alpha}{1-\alpha}\left[S_t - S_t'\right] . \tag{2.16}$$

Note-se que os termos de intercepção e de declive são ambos função da mesma constante de alisamento, o que pode causar dificuldades quando se pretende determinar o valor óptimo desta.

A abordagem de Holt difere neste aspecto do método adoptado por Brown, na medida em que utiliza dois parâmetros de alisamento. Tal como Brown, Holt define duas equações no seu método:

$$S_t = \alpha y_t + (1-\alpha)(S_{t-1} + B_{t-1}) \,, \tag{2.17}$$

$$B_t = \beta(S_t - S_{t-1}) + (1-\beta)B_{t-1}\,, \tag{2.18}$$

onde S_t é uma variação da equação tradicional do *Exponential Smoothing* – (2.7) –, B_t é uma função que tenta captar a tendência da série que se pretende analisar e α e β são os factores de alisamento para cada equação. A primeira equação ajusta os valores registados utilizando a tendência e o valor calculados no instante anterior, o que permite reduzir o tempo que o algoritmo demora a adaptar-se às mudanças de tendência e minimizar o erro, quando comparado com métodos como a *Moving Average*. A segunda equação apresenta um mecanismo semelhante à equação tradicional do Exponential Smoothing, mas aplicado aos valores da tendência da série, aqui representados como a diferença entre os últimos dois valores registados. Tal como no algoritmo do *Exponential Smoothing*, também este necessita de valores iniciais para S_t e B_t. S_0 toma normalmente o valor registado em y_0, enquanto B_0 é geralmente inicializado com uma das seguintes alternativas: $B_0 = y_1 - y_0$, $B_0 = [(y_1 - y_0) + (y_2 - y_1) + (y_3 - y_2)]/3$ ou $B_0 = (y_n - y_1)/(n-1)$. Os valores dos parâmetros α e β podem ser obtidos através de técnicas de optimização não-linear, tais como o algoritmo de Levenberg-Marquardt [28], que minimizam a o erro quadrático médio ou o erro percentual absoluto médio (MAPE[2]). Quando o objectivo é estimar valores no futuro utilizando o algoritmo de Holt, recorre-se à equação

$$\hat{Y}_{t+h} = S_t + B_t h\,. \tag{2.19}$$

2.3.4 Método de Holt-Winters

Frequentemente, as séries temporais que se pretende estudar apresentam, para além de tendência, flutuações periódicas que formam determinados padrões sazonais. Nenhum dos métodos descritos anteriormente consegue lidar de forma eficaz com estas situações, pois estes não consideram a existência deste factor adicional. Este factor de sazonalidade pode ser introduzido no modelo de forma aditiva ou de forma multiplicativa. Em 1965, Winters generaliza o algoritmo desenvolvido por Holt alguns anos antes para incluir um factor de sazonalidade, e apresenta um novo método. Este método recebeu o nome de método de Holt-Winters.

O método de Holt-Winters considera séries que apresentam este tipo de comportamento, e que podem ser expressas pelo modelo

$$y_t = \mu_t + \delta_1 t + \rho_t + \varepsilon_t\,, \tag{2.20}$$

no caso aditivo, ou

$$y_t = (\mu_t + \delta_1 t) \cdot \rho_t + \varepsilon_t\,, \tag{2.21}$$

no caso multiplicativo, onde μ_t é o termo de intersecção, $\delta_1 t$ é o termo de declive da linha de regressão, ρ_t o termo de sazonalidade e ε_t o residual.

Se o modelo for aditivo, as suas componentes podem ser calculadas com as expressões

$$S_t = \alpha(y_t - S_{t-L}) + (1-\alpha)(S_{t-1} + B_{t-1})\,, \tag{2.22}$$

$$B_t = \beta(S_t - S_{t-1}) + (1-\beta)B_{t-1}\,, \tag{2.23}$$

[2] Mean Absolute Percentual Error

$$I_t = \gamma(y_t - S_t) + (1-\gamma)I_{t-L}\,, \tag{2.24}$$

$$\hat{Y}_{t+h} = S_t + B_t h + I_{t-L+m}\,. \tag{2.25}$$

Por outro lado, se o modelo for multiplicativo as seguintes equações são utilizadas para o cálculo dos seus componentes:

$$S_t = \alpha\frac{y_t}{I_{t-L}} + (1-\alpha)(S_{t-1} + B_{t-1})\,, \tag{2.26}$$

$$B_t = \beta(S_t - S_{t-1}) + (1-\beta)B_{t-1}\,, \tag{2.27}$$

$$I_t = \gamma\frac{y_t}{S_t} + (1-\gamma)I_{t-L}\,, \tag{2.28}$$

$$\hat{Y}_{t+h} = (S_t + B_t h)I_{t-L+m}\,, \tag{2.29}$$

onde y_t são as observações, S_t é uma variação da *Exponential Smoothing*, B_t é um factor de tendência, I_t é a componente de sazonalidade do modelo de regressão, $\hat{Y}_t$ é a fórmula de previsão, e L é o número de períodos da série. O número de períodos da série refere-se ao número de ciclos sazonais presentes nas observações (e.g. se existir um ano completo de dados, e a sazonalidade for anual, existe 1 período; se a sazonalidade for mensal existem 12 períodos). α, β e γ são constantes que deverão ser estimadas com o objectivo de minimizar o erro quadrático médio. Uma vez que se lida com ciclos temporais que se pretendem identificar e que são, não raramente, muito complexos, deve-se dispor de uma quantidade de observações iniciais que possibilite a inicialização do método e que deverá conter pelo menos $2L$ períodos (mais ainda se o modelo for muito complexo).

É ainda necessário associar valores iniciais para S_0, B_0 e I_0. S_t assume na maioria dos casos o valor de y_t para $t < L$. Os primeiros L elementos de B_t são geralmente inicializados de acordo com a expressão

$$B_t = \frac{1}{L}\left(\frac{y_{L+1}-y_1}{L} + \frac{y_{L+2}-y_2}{L} + \cdots + \frac{y_{L+L}-y_L}{L}\right). \tag{2.30}$$

No caso de I_t, os seus primeiros L elementos são frequentemente inicializados seguindo a regra

$$I_t = y_t - S_t, \quad i = 1, \ldots, L\,, \tag{2.31}$$

ou utilizando técnicas mais complexas, tais como a descrita em [29].

2.3.5 Regressão Linear

A regressão linear é a mais popular técnica de modelação utilizada actualmente, principalmente pela sua versatilidade e facilidade de implementação [30]. Na sua forma mais simples, um modelo de regressão linear é definido pela expressão

$$y_i = \beta_0 + \beta_1 x_i\,, \tag{2.32}$$

onde β_0 é o termo de intercepção (o valor de y_i quando $i = 0$), β_1 é o declive da linha de regressão e x_i é uma variável independente. É fácil verificar que variando os parâmetros β_0 e β_1 é possível obter qualquer linha recta. Em algumas situações é fácil determinar os valores destes,

mas de uma forma geral é necessário recorrer à utilização de métodos numéricos para os estimar com base em observações realizadas.

Um desses métodos é o método dos mínimos quadrados, que determina os valores dos parâmetros que minimizam a soma dos erros quadráticos ou, por outras palavras, a soma dos residuais ao quadrado. Considerando $\hat{y}_i$ a função que estima os valores de y_i, obtida após a utilização do método dos mínimos quadrados, torna-se também importante estimar a variância de $\hat{y}_i$, $\hat{\sigma}^2$, para que um intervalo de confiança possa ser estabelecido [33]. Partindo da definição de $\hat{y}_i$,

$$\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i \,, \tag{2.33}$$

define-se a função que estima os residuais obtidos

$$\hat{\varepsilon}_i = y_i - \hat{y}_i = y_i - \left(\hat{\beta}_0 + \hat{\beta}_1 x_i\right) . \tag{2.34}$$

Assumindo que os valores de $\hat{\varepsilon}_x$ são variáveis aleatórias não correlacionadas, com média zero e variância constante, então o valor de $\hat{\sigma}^2$ pode ser estimado dividindo a soma dos residuais ao quadrado pelos seus graus de liberdade (número de observações menos o número de parâmetros na função). Neste caso $\hat{\sigma}^2$, também conhecido como média quadrática residual, é definido por

$$\hat{\sigma}^2 = \frac{\sum_{i=1}^{n} \hat{\varepsilon}_i}{n-2} . \tag{2.35}$$

Assumindo que se dispõe de n observações, e generalizando o modelo básico analisado, conclui-se que qualquer modelo de regressão linear pode ser representado pela seguinte expressão:

$$y = X\beta \,, \tag{2.36}$$

onde y é um vector $n \times 1$ que contêm os valores da resposta escalar registada em cada observação, β é um vector $p \times 1$ de parâmetros desconhecidos e X é uma matriz $n \times p$ de regressores. Regra geral, o termo constante (termo de intercepção) está sempre incluído no conjunto dos regressores, e.g. se definirmos $X_{i1} = 1$, para $i = 1, \ldots, n$, então β_1 é o termo de intercepção. O modelo diz-se linear pois os parâmetros β são linearmente independentes, e não devido ao significado geométrico da expressão, que provoca frequentemente a confusão. Por esta razão, e porque o desenho da função não é necessariamente uma linha recta, é relativamente comum designar estes modelos como "estatisticamente lineares" ou "lineares nos parâmetros" como forma de reduzir a ambiguidade.

Uma vez que os modelos lineares não se cingem a simples linhas ou planos, estes podem portanto assumir inúmeras formas. A expressão (2.36) possibilita a definição, entre muitas outras, das seguintes funções:

$$f(x) = \beta_0 + \beta_1 x + \beta_2 x^2 \,;$$

$$f(x) = \beta_0 + \beta_1 \ln x \,;$$

$$f(x, z) = \beta_0 + \beta_1 \sin 2x + \beta_2 \cos^2 x + \beta_2 \cos z \,.$$

No entanto funções como

$$f(x) = \beta_0 + \beta_1\beta_2 x$$

não são suportadas pelo modelo, pois os seus parâmetros não são lineares. Para resolver um problema deste género é necessário utilizar um método apropriado para lidar com modelos não lineares. De notar que, apesar de muitos modelos serem melhor expressos de uma forma não-linear, há muitos destes que podem ser representados suficientemente bem por um modelo linear. Isto deve-se ao facto de muitos deles serem inerentemente lineares e/ou porque, para pequenos intervalos, qualquer modelo pode ser aproximado por um modelo linear [31][32].

2.3.6 Loess

Loess - ou *Lowess*, nome pelo qual esta técnica também é conhecida – provêm da expressão ***lo****cally* ***we****ighted* ***s****catterplot* ***s****moothing*. *Loess* é uma técnica desenvolvida William Cleveland em 1979 [40], e que procura associar uma curva (função) a um conjunto de pontos, baseada no algoritmo de regressão local [41], uma variante da regressão linear, mas em que é associado um peso a cada observação. Simplificadamente, a técnica *Loess* resume-se à execução de múltiplas regressões locais sobre partes de um conjunto de dados, em janelas que se sobrepõem, até que todas as observações sejam consideradas, e à computação dos resultados das regressões numa única curva.

Com este objectivo, o algoritmo *Loess* procede, numa primeira etapa, à execução do algoritmo de regressão local para cada par (x_t, y_t) do conjunto das observações. O algoritmo *Loess* associa a cada um destes pares uma janela do conjunto de dados, composta pelos n pares/observações mais próximos do par original. Estas janelas são também designadas como "vizinhanças", e possuem um número fixo de "vizinhos" (observações), o que faz com que estas não tenham sempre a mesma dimensão – quanto mais próximos estiverem os vizinhos, menor é a vizinhança. O algoritmo *Loess* associa ainda um peso a cada vizinho, em função da distância à observação que dá origem à janela – quanto mais afastado estiver o vizinho, menor é o seu peso -, e executa o algoritmo de regressão local com estes pesos como parâmetros. Depois disto, o algoritmo *Loess* recalcula os pesos das observações em função dos residuais obtidos com a regressão local – quanto maior for o valor do residual, menor é o peso da observação associada – e volta a executar este algoritmo, agora com os novos pesos. O algoritmo *Loess* repete este último passo até ser atingida a convergência, diminuindo assim a influência de eventuais *outliers*. Uma descrição mais detalhada do algoritmo *Loess* pode ser encontrada em [42].

A principal vantagem do algoritmo *Loess*, quando comparado com outros métodos, é o facto de dispensar a necessidade de especificar uma função para modelar as observações existentes. Em vez disso, é apenas requerido um valor d, o grau da função polinomial que se pretende utilizar, e um valor para o parâmetro de alisamento q, que deverá tomar um valor entre $(d+1)/m$ e 1, em que d é o grau da função polinomial e m é o número de observações. O número de elementos contido em cada janela de observações é mq, arredondado para cima. Uma outra vantagem deste algoritmo é a sua versatilidade, o que lhe permite modelar processos complexos mesmo quando não existe um modelo teórico. Estas duas vantagens, aliadas à facilidade de implementação, tornam este algoritmo num dos mais populares no estudo de modelos com uma estrutura determinística muito complexa.

Por outro lado, a necessidade de grandes volumes de dados para produzir modelos de boa qualidade é apontada como um dos principais problemas deste algoritmo. Adicionalmente, o facto de este algoritmo produzir uma função de regressão que não é facilmente representável

por uma fórmula matemática também afasta alguns dos seus potenciais utilizadores, principalmente quando se pretendem transmitir os resultados da análise a outras pessoas. Um outro aspecto negativo deste algoritmo é a sua baixa performance computacional, devido ao grande volume de cálculos intensivos que realiza, o que pode tornar-se um problema quando os conjuntos de dados que se pretendem analisar são bastante grandes. De facto, os mecanismos que utiliza são tão complexos que seria praticamente impossível utilizar este algoritmo quando as técnicas de regressão linear em que se baseia foram desenvolvidas.

2.3.7 Modelos ARIMA (Box-Jenkins)

Em 1970, George Box e Gwilym Jenkins apresentam uma nova abordagem para este problema [34]. O método que propõem difere das técnicas convencionais, pois recorre à análise do comportamento passado de uma variável para escolher o melhor modelo de previsão entre uma vasta colecção de modelos possíveis. Box e Jenkins assumem que qualquer padrão numa qualquer série temporal pode ser representado por uma de três categorias de modelos possíveis: a) Auto-regressivo (*Autoregressive*), b) *Moving Average*, ou c) *Autoregressive Moving Average*.

Um modelo auto-regressivo estima o futuro com base numa função linear dos seus valores passados, de acordo com a expressão

$$x_t = \varphi_0 + \varphi_1 x_{t-1} + \varphi_2 x_{t-2} + \cdots + \varphi_p x_{t-p} + \varepsilon_t \,, \tag{2.37}$$

onde x_t é a série que se pretende analisar, φ_i são parâmetros que devem ser estimados, p é a ordem do processo auto-regressivo, e ε_t é o residual. Os parâmetros são estimados recorrendo a um método de regressão múltipla *standard* ou a técnicas optimizadas para este processo, como as equações de Yule Walker [35].

O modelo *Moving Average* encontra-se descrito na subsecção 2.3.1 deste documento e, nos modelos ARIMA, procura estimar o futuro baseado numa combinação linear dos erros passados, tomando a forma

$$x_t = \mu + \varepsilon_t - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \cdots - \theta_q \varepsilon_{t-q} \,, \tag{2.38}$$

onde x_t é a série que se pretende analisar, θ_i são parâmetros a estimar, q é a ordem do processo de *Moving Average*, μ é a média dos valores de x, e $\varepsilon_t, \ldots, \varepsilon_{t-q}$ são os residuais. A estimativa dos parâmetros de um processo *Moving Average* como este é mais complexa do que a dos parâmetros de um modelo auto-regressivo, uma vez que os valores dos residuais não são observáveis. Isto implica a utilização de procedimentos iterativos não-lineares, por oposição aos mais simples métodos de regressão linear, sendo um dos mais utilizados o método de Durbin [36]. Ainda devido a esta razão, torna-se também menos óbvia a interpretação de um modelo deste género, quando comparado com um modelo auto-regressivo.

A conjugação destes dois modelos dá origem a um modelo designado *Autoregressive-Moving Average*, também conhecido simplesmente pelo seu acrónimo, ARMA. Este modelo expressa-se através da equação

$$x_t = \varphi_0 + \varphi_1 x_{t-1} + \varphi_2 x_{t-2} + \cdots + \varphi_p x_{t-p} + \varepsilon_t - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \cdots - \theta_q \varepsilon_{t-q} \,. \tag{2.39}$$

Muitas vezes a série a analisar não apresenta um comportamento estacionário. Nestes casos é necessário proceder a uma operação designada diferenciação, que deve ser aplicada d vezes à

série até que esta atinja a estacionariedade. Quando se torna necessário diferenciar uma série, o modelo deverá adoptar a sua designação mais geral, ARIMA – *Autoregressive Integrated Moving Average*. Uma vez que os mecanismos que suportam o modelo são claros, este pode ser expresso de uma forma mais conveniente de acordo com a notação $ARIMA(p, d, q)$, onde p é a ordem do processo auto-regressivo, d é a ordem de diferenciação e q é a ordem do processo de *Moving Average*.

Um modelo ARIMA, também designado como modelo Box-Jenkins devido aos nomes dos seus autores, pode ser estendido para suportar termos de sazonalidade auto-regressivos ou de *Moving Average*. Nestes casos o modelo adopta a notação $ARIMA(p, d, q) \times (P, D, Q)$, em que P, D e Q são, respectivamente, as ordens dos processos sazonais de auto-regressão, diferenciação e *Moving Average*. Apesar de a adição de termos sazonais tornar o modelo bastante mais complexo, os mecanismos que suportam estes termos são semelhantes aos que dão resposta aos processos não-sazonais.

O processo de construção de um modelo Box-Jenkins possui três etapas bem definidas: a) identificação do modelo, b) estimação do modelo e c) validação do modelo.

Durante a etapa de identificação do modelo são efectuadas actividades que verificam se a série a analisar é estacionária e se possui algum factor de sazonalidade que deva ser considerado. A estacionariedade pode ser avaliada de forma empírica através da análise da representação gráfica da série, ou do seu gráfico de auto-correlação. Se o processo indicar que a série não é estacionária, aplica-se uma diferenciação e repete-se o processo até que a estacionariedade seja atingida. A sazonalidade é detectada através da análise dos gráficos de auto-correlação ou do periodograma da série. Depois de identificadas as ordens de diferenciação e de sazonalidade, estimam-se os valores de p e q analisando os gráficos de auto-correlação e auto-correlação parcial da série, já diferenciada [37].

Na etapa de estimação do modelo procura-se estimar os valores dos parâmetros do modelo – ver (2.39). Este é um problema de estimação não-linear bastante complexo, o que faz com que esta tarefa seja normalmente levada a cabo recorrendo a software desenvolvido especificamente para lidar com esta tarefa. Tipicamente, as técnicas mais utilizadas por estas aplicações são o método da máxima verosimilhança [38] e o método dos mínimos quadrados não-linear [28].

A etapa de validação do modelo consiste, como se pode adivinhar pelo nome, na avaliação dos resultados obtidos pelo modelo. As assumpções tomadas são testadas nesta fase, identificando-se eventuais situações que se revelem inadequadas. Geralmente basta analisar os residuais resultantes da aplicação do modelo obtido, que deverão apresentar propriedades de ruído branco [39]. Caso alguma irregularidade seja detectada é necessário voltar à etapa anterior e identificar um modelo melhor. Se nenhum problema for detectado no modelo, este poderá então ser facilmente utilizado para efectuar estimativas acerca do comportamento futuro da série que se está a considerar, embora seja necessário o recurso a uma aplicação computacional devido à sua elevada complexidade.

2.3.8 Modelo de Koyck

Não são raras as situações em que uma variável dependente é função, ou depende, da soma dos valores actuais e passados da variável independente (e do termo de erro), geralmente com pesos diferentes associados a cada um destes valores, na forma

$$y_t = \alpha + \beta_0 X_t + \beta_1 X_{t-1} + \beta_2 X_{t-2} + \cdots + \beta_s X_{t-s} + \varepsilon_t \,, \tag{2.40}$$

onde α é o termo constante, X_i são os valores actuais e passados da variável independente, β_i são os pesos associados a cada termo X_i, também chamados de *coeficientes de latência*, e ε_t é o residual ou termo de erro.

A utilização deste tipo de modelos é complicada por pelo menos três motivos. A primeira dificuldade surge logo quando é necessário escolher o número de termos passados a incluir, s, um valor que é, muitas vezes, impossível de determinar exactamente. Em segundo lugar, e partindo do pressuposto que se consegue determinar s, o número de parâmetros do modelo torna-se $s + 2$, o que se torna um entrave quando as observações existentes não são muitas, causando possíveis problemas estatísticos devido à perda de graus de liberdade. Por último, quanto maior for o valor de s, maior é o risco de as variáveis independentes se tornarem colineares.

Estes problemas motivaram os investigadores a considerar relações entre os vários valores β_i na tentativa de obter um modelo mais simples. Um modelo mais simples teria que ter menos parâmetros e estes deveriam ser mais fáceis de estimar mas, obviamente, sem nunca perder a sua estrutura durante o processo de simplificação. Um desses modelos é o apresentado por Leendert Koyck, em 1954 [44].

Koyck reescreve (2.40) com um número infinito de termos, da seguinte forma:

$$y_t = \alpha + \sum_{i=0}^{\infty} \beta_{i+1} X_{t-i} + \varepsilon_t \tag{2.41}$$

Assumindo então que

$$\frac{\beta_2}{\beta_1} = \lambda, \frac{\beta_3}{\beta_1} = \lambda^2, \ldots, \frac{\beta_{i+1}}{\beta_1} = \lambda^i, \ldots \,, \tag{2.42}$$

onde $0 \leq \lambda < 1$. Esta estrutura assume que os pesos dos termos decrescem geometricamente (ou exponencialmente, no caso contínuo) com o passar do tempo. Por esta razão, este modelo, o modelo de Koyck, é também conhecido como *modelo de latência geométrica*.

É frequente encontrar, na literatura, os termos *latência distribuída* ou *distribuição de latência*. Esta terminologia está relacionada com a similaridade entre as propriedades dos coeficientes de latência e as da função massa de probabilidade. De acordo com esta relação, podemos reescrever (2.41) tal que

$$y_t = \alpha + \beta_1 \sum_{i=0}^{\infty} w_i X_{t-i} + \varepsilon_t \tag{2.43}$$

onde os termos w_i representam a estrutura de latência. Assume-se então que a distribuição de latência possui as seguintes propriedades:

$$w_i \geq 0 \text{ , e} \tag{2.44}$$

$$\sum_{i=0}^{\infty} w_i = 1 \tag{2.45}$$

ou, traduzido em palavras, os coeficientes de latência assumem-se não-negativos e o seu somatório deve ser 1.

Uma vez que a estrutura apresentada em (2.42) implica $w_i = \lambda^i$, verifica-se que esta expressão satisfaz (2.44), mas não (2.45). Para que o somatório dos coeficientes tenha 1 como resultado é necessário redefinir a expressão de cálculo dos coeficientes. Assim

$$w_i = (1-\lambda)\lambda^i \,. \tag{2.46}$$

Se compararmos as estruturas definidas em (2.42) e em (2.46) verificamos que estas não são fundamentalmente diferentes. Por essa razão, e por ser essa a formulação mais adoptada na literatura, opta-se por (2.42). Substituindo em (2.43), obtêm-se

$$y_t = \alpha + \beta_1 X_t + \beta_1 \lambda X_{t-1} + \beta_1 \lambda^2 X_{t-2} + \cdots + \beta_1 \lambda^n X_{t-n} + \cdots + \varepsilon_t \,. \tag{2.47}$$

Aplicando um período de latência à expressão obtida e multiplicando por λ,

$$\lambda y_{t-1} = \lambda\alpha + \lambda\beta_1 X_{t-1} + \beta_1 \lambda^2 X_{t-2} + \cdots + \beta_1 \lambda^{n+1} X_{t-n-1} + \cdots + \lambda\varepsilon_{t-1} \,. \tag{2.48}$$

Se subtrairmos agora esta expressão a (2.47), obtêm-se

$$y_t - \lambda y_{t-1} = \alpha(1-\lambda) + \beta_1 X_t + \varepsilon_t - \lambda\varepsilon_{t-1} \,. \tag{2.49}$$

Definindo $\alpha^* = \alpha(1-\lambda)$ e $\varepsilon_t^* = \varepsilon_t - \lambda\varepsilon_{t-1}$, e substituindo na expressão anterior,

$$y_t = \alpha^* + \lambda y_{t-1} + \beta_1 X_t + \varepsilon_t^* \,. \tag{2.50}$$

Este procedimento foi apresentado por Koyck, sendo por isso conhecido e referido normalmente como *transformação de Koyck*. Como agora só existem três parâmetros no modelo, os problemas relacionados com o processo de previsão são largamente reduzidos [45]. A interpretação numérica destes parâmetros é também relativamente simples: β_1 está directamente relacionado com a importância da variável independente para o cálculo do valor estimado e, por isso, é mais importante para as estimativas a curto prazo; λ, por seu lado, controla a influência do valor registado num dado instante no valor observado no período seguinte; e α, obviamente, está relacionado com o nível da série.

2.4 Adequação das técnicas e métodos de previsão

Todas as técnicas apresentadas neste capítulo foram estudadas e testadas com recurso aos dados existentes, resultado do processo de recolha dos níveis de utilização das diferentes variáveis de capacidade em análise.

As primeiras duas técnicas apresentadas, *Moving Average* e *Exponential Smoothing*, são a base de grande parte dos métodos existentes, e são apresentadas mais por esta razão do que pelos resultados obtidos com as suas observações. Da mesma forma, o *Double Exponential Smoothing* é aqui descrito como forma de demonstrar o processo evolução de um método, sendo uma técnica utilizada para modelar séries temporais que não apresentam na sua decomposição termos de sazonalidade, ao contrário do verificado nos valores observados.

A existência destes termos de sazonalidade faz com que métodos estatísticos mais complexos, tais como o método de Holt-Winters, o método ARIMA, métodos de filtragem

adaptativa [57] ou métodos GARCH [58], sejam utilizados frequentemente para resolver problemas similares. No entanto, a utilização destes métodos implica forçosamente a existência de um elevado número de observações, facto que nem sempre é garantido. Neste projecto este factor foi decisivo para a escolha dos métodos a implementar na aplicação a desenvolver, pois os dados existentes – valores registados do nível de utilização das variáveis de capacidade – revelaram-se muitas vezes insuficientes. A título de exemplo tomemos o método ARIMA, que apesar de mais complexo do que o método de Holt-Winters, obteve resultados piores do que este último, precisamente porque depende de uma maior quantidade de dados para obter bons resultados.

Outros algoritmos procuram isolar os termos de uma série, possibilitando a análise individual de cada componente, ou apenas de alguns. São exemplos deste tipo de abordagem o método de Loess ou o filtro de Hodrick-Prescott [59]. Este tipo de métodos é especialmente útil quando apenas se pretendem analisar algumas das suas componentes.

Existe ainda uma categoria de métodos que assumem que existe uma relação causal entre duas (ou mais) variáveis distintas. A regressão linear é o exemplo imediato quando se fala de um método causal, o que faz deste método o mais utilizado nos processos de modelação, mas há várias alternativas tais como o modelo de Koyck, ou o método de autoregressão vectorial [60], ou os modelos ARMAX [61], uma variação dos modelos ARIMA.

Capítulo 3

Modelação

Este capítulo descreve o processo de modelação que suporta a aplicação desenvolvida, e que tem como objectivo a estimação dos níveis de alguns dos recursos necessários no futuro, ou seja, a capacidade necessária, para várias aplicações utilizadas pela Direcção de Sistemas de Informação da Sonae. Este processo é o ponto fulcral da presente dissertação. Neste capítulo será descrita a forma como este foi desenvolvido, de uma forma genérica, recorrendo-se a alguns exemplos para melhor ilustrar os mecanismos que o suportam.

Adicionalmente, são apresentados os recursos considerados pela aplicação e algumas das aplicações que destes dependem e que foram, igualmente, consideradas.

Os cálculos realizados durante esta fase do projecto foram efectuados com recurso ao ambiente de computação estatística R [56]. O R é uma ferramenta *open-source*, de utilização livre, construída sobre uma linguagem extremamente flexível com o mesmo nome. As suas principais funcionalidades incluem a manipulação de dados, o cálculo e a visualização gráfica destes.

3.1 Recursos de capacidade

Os recursos de capacidade a considerar foram definidos em conjunto com o responsável na Sonae, logo no arranque do projecto. As áreas problemáticas ou merecedoras de uma maior atenção foram o principal critério de decisão nesta etapa. Assim sendo, foram identificados dois recursos cuja monitorização foi considerada importante: o poder computacional existente, sob a forma de unidades de processamento disponíveis, e a capacidade de armazenamento, vulgo espaço em disco.

O actual modelo de negócio da Sonae é suportado por uma infra-estrutura tecnológica implementada de acordo com o modelo de arquitectura em três camadas (*three-tier*). Esta infra-estrutura permite a existência de um vasto número de aplicações de software, com as quais os colaboradores, clientes ou fornecedores interagem. Estas aplicações encontram-se distribuídas pelos vários servidores existentes, possuindo cada um destes dezenas de microprocessadores.

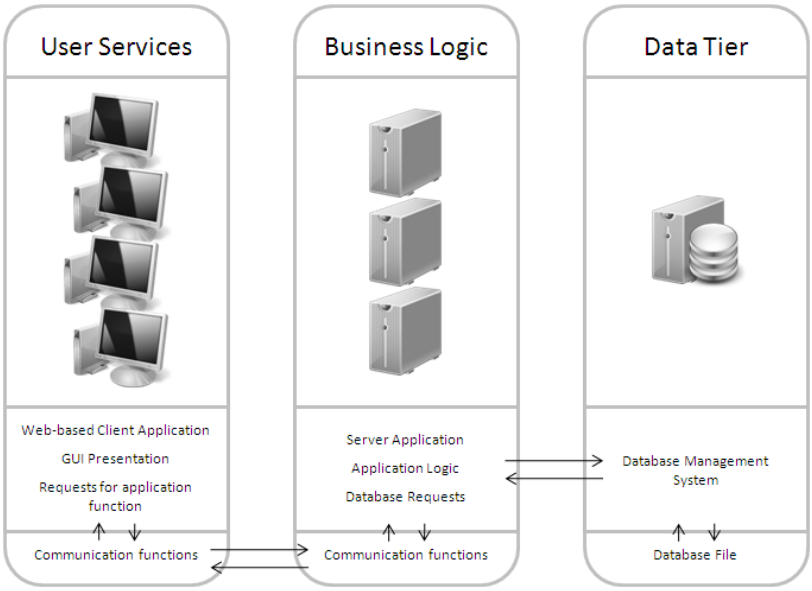


Figura 3.1 – Modelo de arquitectura de três camadas.

Como várias aplicações partilham a mesma máquina, os microprocessadores são distribuídos por estas em função do volume de cargas de trabalho, e.g. numa máquina com 50 microprocessadores, 30 podem estar atribuídos à aplicação a, 5 à aplicação b, e 10 à aplicação c, sendo os restantes 5 atribuídos dinamicamente às aplicações que deles possam precisar num determinado momento. Adicionalmente, uma aplicação poderá ainda utilizar processadores reservados para outra se necessitar destes e estes não estiverem a ser utilizados. Como é óbvio, um incremento no número de execuções de uma determinada aplicação, ou no número de pedidos das aplicações cliente, implica um aumento nas suas necessidades em termos de poder computacional. Uma vez que a Sonae é uma organização em constante expansão, facilmente se percebe que o limite máximo disponível num determinado momento se irá revelar insuficiente algures no futuro. Assim, definiu-se como uma das funcionalidades da aplicação a desenvolver a estimação do número de processadores num determinado momento no futuro.

A cada aplicação está associada uma determinada quantidade de espaço em disco, com limites bem definidos e adaptados à realidade de cada aplicação. Este espaço em disco destina-se, exclusivamente, ao armazenamento da base de dados que suporta a aplicação a que está associado. É fácil de concluir, portanto, que a necessidade de espaço em disco de cada aplicação é definida pelo ritmo de crescimento da respectiva base de dados. Como esta não apresenta um nível constante nem uma tendência de crescimento uniforme, o espaço em disco disponibilizado para cada uma das aplicações é ajustado em função da necessidade, procedendo-se a aumentos quando a base de dados cresce e a reduções quando esta é purgada. Com esta dinâmica surgem problemas óbvios, uma vez que o espaço em disco total está dependente do hardware instalado, e da quantidade que estes oferecem. Actualmente as estimativas das necessidades futuras são efectuadas de forma empírica, baseadas em experiências passadas, por parte do analista. Pretende-se que a aplicação a desenvolver seja um auxiliar a este processo, fornecendo ao analista uma base numérica que possa fundamentar ou justificar a sua decisão.

Por último, foram definidas as aplicações que seriam suportadas pela aplicação. Trata-se de aplicações utilizadas pela Sonae no decorrer das suas operações diárias, sendo estas: o ERP da empresa (Retek); o Data Warehouse (DW), que consolida a informação histórica e efectua uma série de análises sobre esta; as aplicações que suportam o Cartão Cliente (VLL e VLLC); a

Plataforma de Tratamento de Receitas Comerciais (NPRC); e o Workflow, que efectua a gestão dos processos.

3.2 Variáveis de input do modelo

A Sonae, na pessoa do responsável pelo projecto na empresa, pretende que o modelo a desenvolver receba como input variáveis expressas em termos de negócio. Neste sentido, foram consideradas algumas hipóteses como "número de utilizadores", "número de lojas", "volume de facturação" ou "número de clientes", entre outras. Das várias hipóteses existentes, a variável "número de lojas" foi a escolhida, principalmente porque, em termos logísticos, se afigura como a opção mais apelativa. O facto de se possuir informação sobre o número exacto de lojas existentes num horizonte temporal relativamente alargado, ao contrário das outras variáveis, também contribuiu para esta decisão. Não obstante a importância destes critérios, importa avaliar qual a correlação existente entre esta variável e a variação no valor das variáveis de capacidade. Este processo está detalhado na subsecção seguinte ("Construção do modelo") deste documento.

Algumas lojas, apesar de estarem associadas a insígnias diferentes, apresentam um comportamento semelhante e podem ser consideradas, ao nível da utilização de recursos de capacidade, equivalentes em termos de complexidade e funcionamento. Por esta razão, decidiu-se que o modelo a implementar deveria permitir dividir as insígnias existentes em grupos, de acordo com este critério. Nesta fase foi consultado um dos responsáveis da Sonae que lida diariamente com este tipo de informação, sendo definidos quatro grupos de insígnias, que se designaram "classes". As insígnias que fazem parte de cada classe podem variar, dependendo da aplicação em análise, mas esta divisão deverá ser sempre feita entre quatro classes. Para lidar com esta situação, definiu-se que a aplicação a desenvolver deve possuir funcionalidades que permitam que o utilizador especifique as diferentes classes de insígnias, para cada aplicação. Recorrendo à utilização de dados históricos e/ou conhecimento empírico dos especialistas nesta área no universo dos colaboradores da Sonae, é possível estabelecer facilmente uma razão de proporção entre as diferentes classes (e.g. 1 Continente = 3 Modelo), devendo esta informação ser igualmente fornecida à aplicação a ser desenvolvida.

3.3 Construção do modelo

3.3.1 Utilização do espaço em disco

Análise inicial dos dados

O primeiro passo a efectuar no processo de construção de um modelo deve ser a análise inicial dos dados, mais particularmente efectuando o desenho da série temporal que os representa [46][47]. A primeira aplicação que foi considerada foi o Retek, pela simples razão de ser aquela que utiliza mais recursos e ser fundamental para a organização.

As insígnias foram divididas pelas quatro classes, da seguinte forma:

- Classe 1: Continente;
- Classe 2: Modelo, Worten;
- Classe 3: Bonjour, Modalfa, SportZone, Vobis;

- Classe 4: BOOK.IT, Cafetarias, Farmácias, GALP, Loop, PAR, Pet&Plants, Postos de Abastecimento, Worten Gamer, Worten Mobile, Genérico Portugal, Outras Insígnias;

não sendo consideradas as restantes insígnias – e.g. Del Garden, Insco, Expedis, etc.

Existem dados registados sobre a utilização do espaço em disco desde aproximadamente 2005, mas estes só se tornam relevantes a partir do início de 2007. Podemos verificar a sua evolução na Figura 3.1.

Uma primeira análise mostra uma clara tendência de subida no nível de utilização do espaço em disco, tal como se previa, não se notando qualquer indício imediato da presença de um efeito sazonal. É notória, no entanto, a presença de valores que indiciam a existência de *outliers*, valores que diferem substancialmente daqueles que seriam espectáveis.

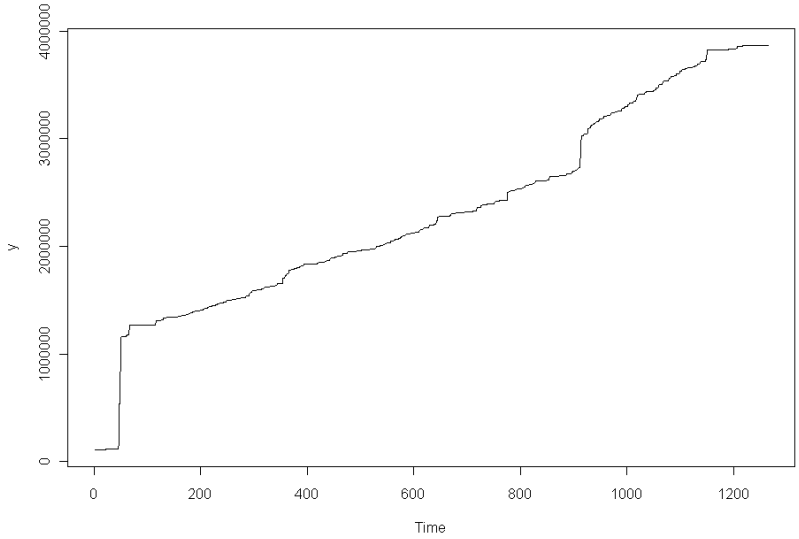


Figura 3.2 – Utilização do espaço em disco, por uma determinada aplicação. O eixo dos *xx* representa o número de dias após 01-01-2007. O eixo dos *yy* representa a quantidade de espaço em disco utilizada, em *megabytes*.

Da mesma forma, analisa-se o comportamento da série que representa a evolução no número de lojas existentes, sem no entanto considerar as inevitáveis diferenças entre cada uma delas:

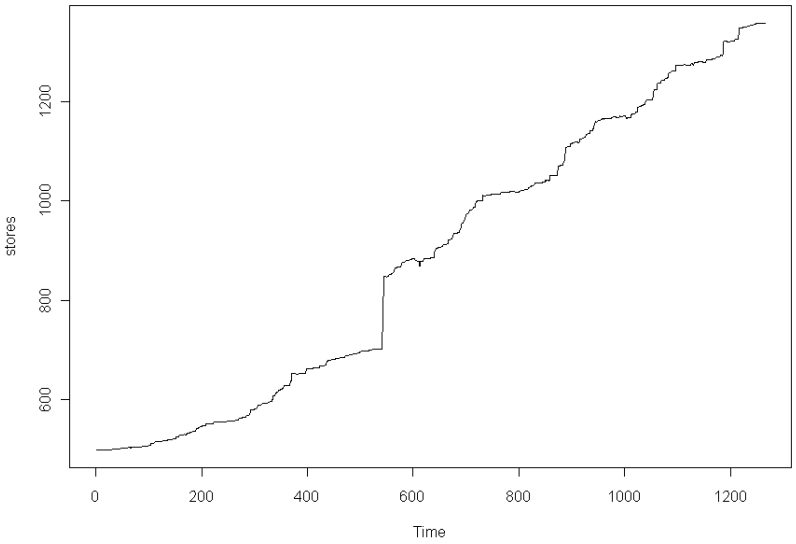


Figura 3.3 – Evolução do número total de lojas, desde 01-01-2007.

Análise da correlação

Uma análise à correlação entre esta variável e a variável em estudo (evolução do espaço em disco necessário), utilizando o coeficiente de correlação de Pearson [48] como métrica, de acordo com a expressão

$$\rho_{X,Y} = \frac{E[(X-\mu_X)(Y-\mu_Y)]}{\sigma_X \sigma_Y}, \tag{3.1}$$

produz como resultado um coeficiente de valor 0.9581284, valor que o teste de significância demonstra ser estatisticamente significativo a 5%, com um intervalo de confiança a 95% compreendido entre 0.9533575 e 0.9624207. Este resultado indica que as duas séries estão fortemente correlacionadas – existe uma associação positiva muito forte entre ambas –, tal como se pode verificar na Figura 3.4. No entanto, nesta figura é notória a presença de *outliers*, já identificados anteriormente de forma empírica. Se removermos essas observações – Figura 3.5 – os resultados obtidos são ligeiramente diferentes.

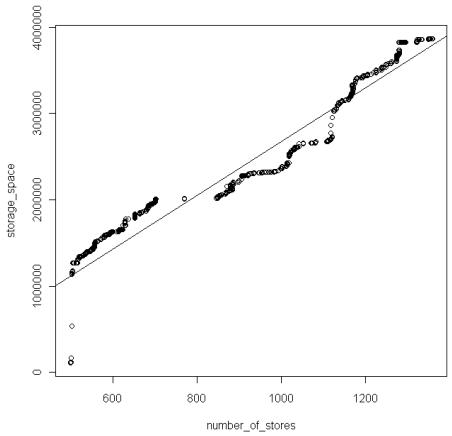


Figura 3.4 – Gráfico de dispersão da utilização do espaço em disco, em função do número de lojas.

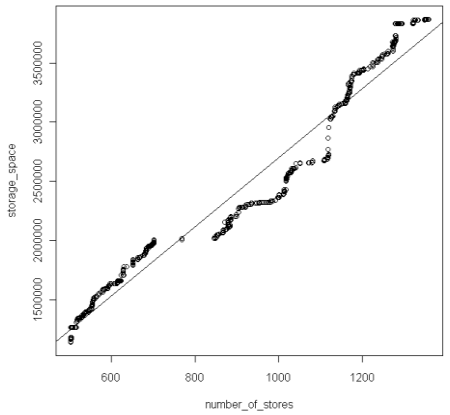


Figura 3.5 – Gráfico de dispersão da utilização do espaço em disco, em função do número de lojas (com *outliers* removidos).

Após a remoção dos *outliers* detectados, o coeficiente de correlação calculado toma o valor de 0.978283, com significância estatística a 5%, e com um intervalo de confiança a 95% compreendido entre 0.9757285 e 0.9805715, o que demonstra um incremento da correlação que, apesar de reduzido, é ainda assim assinalável.

Se procedermos à discretização do número de lojas, analisando cada classe separadamente, verifica-se que o valor do coeficiente de correlação de algumas é superior à de outras. Esta situação já era esperada, uma vez que a evolução do número de lojas de uma classe é, só por si, díspar das restantes.

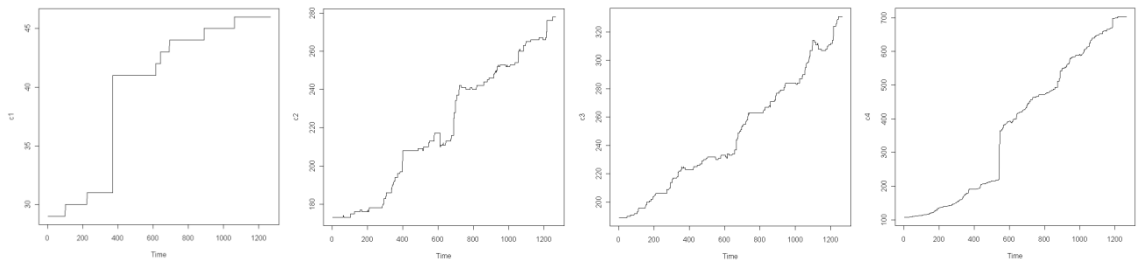


Figura 3.6 – Evolução do número de lojas de cada classe, desde 01-01-2007. Da esquerda para a direita, os gráficos correspondem às classes 1 até 4.

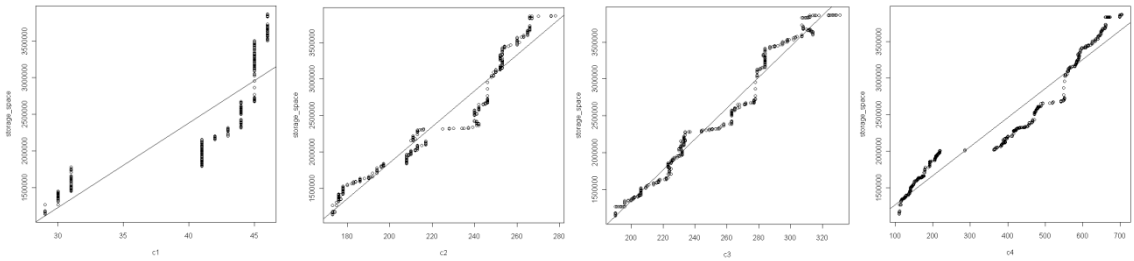


Figura 3.7 – Gráficos de dispersão da utilização de espaço em disco em função do número de lojas de cada classe, desde 01-01-2007. Da esquerda para a direita, os gráficos correspondem às classes 1 até 4.

Os coeficientes de correlação calculados entre o número de lojas de cada classe e a utilização do espaço em disco foram de 0.846408, 0.9683291, 0.9877048 e 0.9713233, para as classes 1 a 4, respectivamente. A análise destes valores indica-nos que o número de lojas da classe 3 é, quando comparado com o número de lojas das outras classes ou mesmo com o número total, uma variável que provavelmente explica melhor a evolução do espaço em disco requisitado pelas aplicações da Direcção de Sistemas de Informação da Sonae. Esta informação, só por si, seria já suficiente para cingir o resto do processo à análise apenas de duas ou três classes. Porém, um dos requisitos do projecto é que o modelo a implementar suporte a existência das quatro classes, ainda que isso implique algumas limitações. Esta análise serve o propósito, portanto, de alertar para o facto de a inclusão de todas as variáveis poder ter um impacto negativo no desempenho do modelo que se pretende construir.

Regressão Linear

Uma vez que se pretende determinar o impacto de cada loja na utilização dos recursos ao dispor do sistema, optou-se por iniciar a tentativa de modelação recorrendo a uma técnica de regressão linear, que considera o número de lojas das diferentes classes de insígnias e a quantidade de espaço em disco utilizada. Assim, considerando as quatro classes de insígnias existentes, definiu-se o seguinte modelo para tentar representar os dados:

$$y_t = \phi_1 x_{1t} + \phi_2 x_{2t} + \phi_3 x_{3t} + \phi_4 x_{4t} + \varepsilon_t \,, \tag{3.2}$$

onde y_t é a série observada, x_{it} é o número de lojas da classe i, no instante t, ϕ_i é o parâmetro que se pretende estimar, e que controla a influência de cada classe, e ε_t é o residual obtido. A expressão pode ser ainda representada adoptando a notação matricial, assumindo a forma

$$Y = \theta X + \varepsilon \,, \tag{3.3}$$

onde Y é um vector contendo n observações, θ é um vector de parâmetros a estimar, com tamanho m, X é uma matriz $n \times m$ contendo o número de lojas de cada classe no momento de cada observação, e ε é o vector de residuais.

Estas expressões estão sujeitas às restrições impostas pelos factores de proporção – se uma loja de uma insígnia, e.g. Worten, é considerada equivalente a duas lojas de uma outra insígnia, e.g. Modalfa, então é natural e expectável que o impacto da primeira seja aproximadamente o dobro de uma das segundas.

Uma vez seleccionado o modelo, procedeu-se à estimação dos seus parâmetros (θ). Para

este efeito, recorreu-se ao método dos mínimos quadrados, que procura minimizar a expressão

$$\min\left([\theta X - Y]^2\right). \tag{3.4}$$

Utilizando o ambiente de computação estatística R, obtiveram-se os seguintes resultados:

```
> v <- read.table("disk")
> c1 <- read.table("c1d")
> c2 <- read.table("c2d")
> c3 <- read.table("c3d")
> c4 <- read.table("c4d")
> c1 <- read.table("c1d")
> A <- cbind(c1=c1$V1, c2=c2$V1, c3=c3$V1, c4=c4$V1)
> B <- v$V1
> G <- matrix(nrow=4, ncol=4, byrow=TRUE,data=c(25,-50,0,0,0,15,-25,0,0,0,7,-15,0,0,0,7))
> H <- c(0,0,0,0.001)
> lsei(A=A,B=B,G=G,H=H)
$X
      c1       c2       c3       c4
8646.161 4323.080 2593.848 1210.462

$residualNorm
[1] 5.09317e-11

$solutionNorm
[1] 2.431094e+14

$IsError
[1] FALSE

$type
[1] "lsei"
```

Substituindo os valores dos parâmetros no modelo (3.2), obtemos a série representada na Figura 3.8. Podemos destacar três intervalos distintos na série obtida: num primeiro, a série apresenta valores muito próximos dos valores observados, ignorando os valores iniciais que podem ser encarados como *outliers*; num segundo período, a série obtida apresenta valores sempre superiores aos registados, embora com uma tendência de crescimento semelhante à registada; por último, na derradeira fase, a série obtida mantêm a tendência de crescimento que trazia de trás, e que é inferior à observada, o que se traduz num afastamento progressivo destas duas séries.

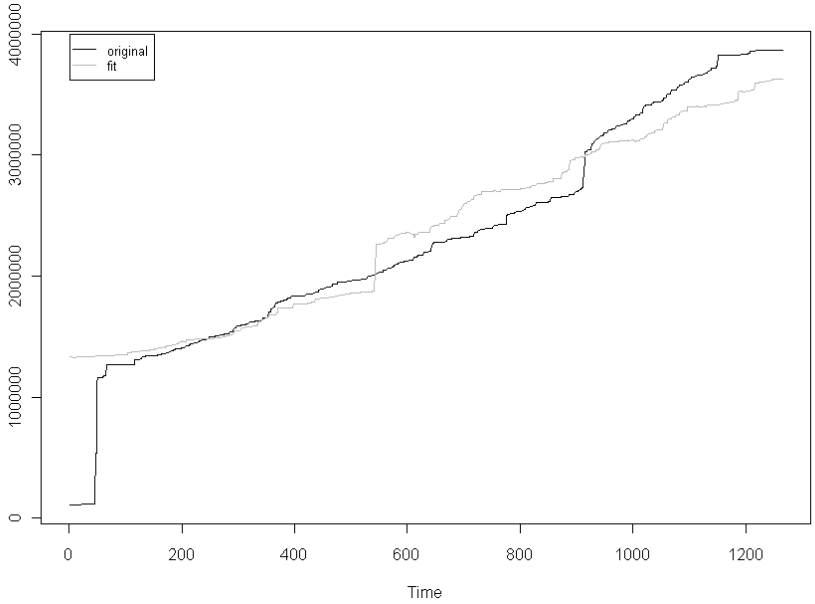


Figura 3.8 – Uma das séries representa a utilização do espaço em disco, por uma determinada aplicação, enquanto a outra representa a sua modelação de acordo com a expressão (3.1). O eixo dos *xx* representa o número de dias após 01-01-2007. O eixo dos *yy* representa a quantidade de espaço em disco utilizada, em *megabytes*.

Estes resultados são explicados pelos processos realizados pelo algoritmo de regressão linear. Este algoritmo procura, de entre uma família de funções especificada *a priori*, aquela que mais se assemelha à função alvo – as observações. Facilmente se conclui que todas as séries passíveis de serem obtidas utilizando este modelo irão apresentar um comportamento bastante semelhante, com variações de nível e de tendência dependentes apenas dos valores dos seus parâmetros.

Com este problema em mente, decidiu-se adicionar um termo adicional ao modelo, um termo dependente do tempo, e um termo constante, para além dos já existentes. Assim, o novo modelo assume a forma

$$d_t = \phi_0 + \phi_1 x_{1t} + \phi_2 x_{2t} + \phi_3 x_{3t} + \phi_4 x_{4t} + \phi_5 t + \varepsilon_t \ . \tag{3.5}$$

Aplicando novamente a expressão (3.4) com recurso ao ambiente de computação estatística R, obtiveram-se agora os seguintes resultados:

```
> v <- read.table("disk")$V1
> c1 <- read.table("c1d")$V1
> c2 <- read.table("c2d")$V1
> c3 <- read.table("c3d")$V1
> c4 <- read.table("c4d")$V1
> t = seq(500,(length(c1)+499))
> A <- cbind(d=1, c1=c1, c2=c2, c3=c3, c4=c4, t=t)
> B <- v
> G <- matrix(nrow=5, ncol=6, byrow=TRUE,data=c(0,25,-50,0,0,0,0,0,15,-25,0,0,0,0,0,7,-15,0,0,0,0,0,7,0,0,0,0,0,10,-1))
> H <- c(0,0,0,0,1,0)
> lsei(A=A, B=B, G=G, H=H)
$X
          d           c1           c2           c3           c4           t
-622585.7538    1577.9117     788.9559     473.3735     220.9076    2209.0764

$residualNorm
[1] 0

$solutionNorm
[1] 5.399039e+13

$IsError
[1] FALSE

$type
[1] "lsei"
```

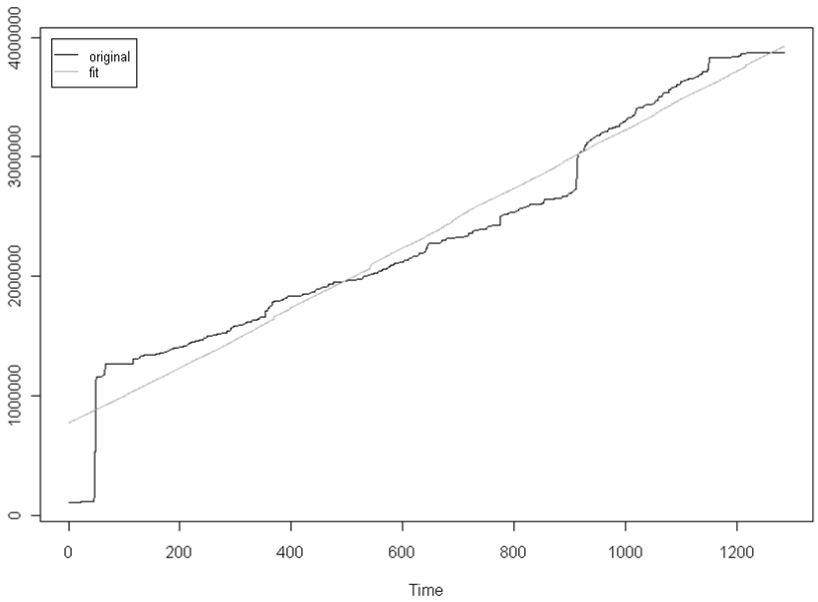


Figura 3.9 – Uma das séries representa a utilização do espaço em disco, por uma determinada aplicação, enquanto a outra representa a sua modelação de acordo com a expressão (3.5). O eixo horizontal representa o número de dias após 01-01-2007. O eixo vertical representa a quantidade de espaço em disco utilizada, em *megabytes*.

As diferenças são notórias quando se substituem os valores dos parâmetros na expressão (3.5), tal como se pode verificar na série que a representa, na Figura 3.9. Por esta razão, tomou-se a decisão de adoptar este modelo para tentar estimar os valores futuros da utilização do espaço em disco. No processo de realização de estimativas o mais importante, muitas vezes, não é a obtenção de um valor esperado, mas sim a definição de um intervalo de confiança. Por esta razão, calculou-se um intervalo de confiança a 95% para a série obtida, de acordo com a expressão

$$CI_t\big(100(1-\alpha)\big) = d_t \pm \mathrm{t}_{\frac{\propto}{2},n-k+1} \cdot \sqrt{\frac{\sum \varepsilon}{n-k}}, \tag{3.6}$$

onde $1-\alpha$ é o valor do intervalo de confiança, d_t é o valor estimado, $t_{a,b}$ é o valor da distribuição t-Student com intervalo de confiança a e b graus de liberdade, n é o número de observações existentes, k é o número de variáveis do modelo, e $\sum \varepsilon$ é o somatório dos residuais.

Outros aspectos

Para tornar a aplicação a desenvolver mais versátil, decidiu-se implementar adicionalmente um método de previsão puramente estatístico, para permitir comparações com o resultado obtido via regressão linear. O método seleccionado foi o método de Holt-Winters, escolhido principalmente pela sua versatilidade e pelos bons resultados que consistentemente demonstra quando comparado com outros métodos similares [49][50][51].

A aplicação do método de Holt-Winters é um processo relativamente simples, neste caso, bastando adoptar o processo descrito na subsecção 2.3.4 deste documento. Desta forma, a aplicação deste método produz o resultado apresentado na Figura 3.10, onde se pode verificar, para além da evolução estimada da série, os intervalos de confiança a 80% e 95% também calculados.

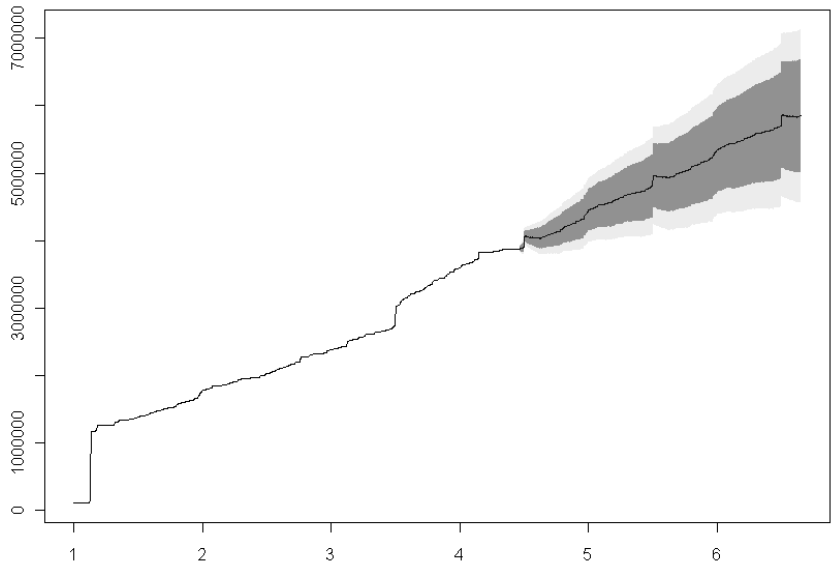


Figura 3.10 – Estimativa da utilização de espaço em disco, por uma determinada aplicação, com recurso ao método de Holt-Winters. São também representados os intervalos de confiança a 80% e 95%, para o período estimado. Cada unidade do eixo horizontal representa um período de um ano, com o primeiro valor a corresponder a 01-01-2007. O eixo vertical representa a quantidade de espaço em disco utilizada/estimada, em *megabytes*.

Considerações adicionais

O processo descrito ao longo desta secção foi efectuado utilizando os dados do Retek. Após a realização deste processo, as mesmas metodologias foram utilizadas para a análise das restantes aplicações – DW, VLL, VLLC, NPRC e Workflow – tendo sido obtidos resultados semelhantes aos que aqui foram demonstrados. Para evitar a redundância, optou-se pela não inclusão neste documento da descrição dos processos e da análise dos resultados obtidos para estas aplicações.

3.3.2 Utilização do processador

Análise inicial dos dados

Tal como no processo de construção do modelo associado à utilização do espaço em disco, inicia-se este processo analisando a série que representa a utilização do poder de processamento pela aplicação em estudo. O Retek foi a aplicação escolhida para iniciar este processo.

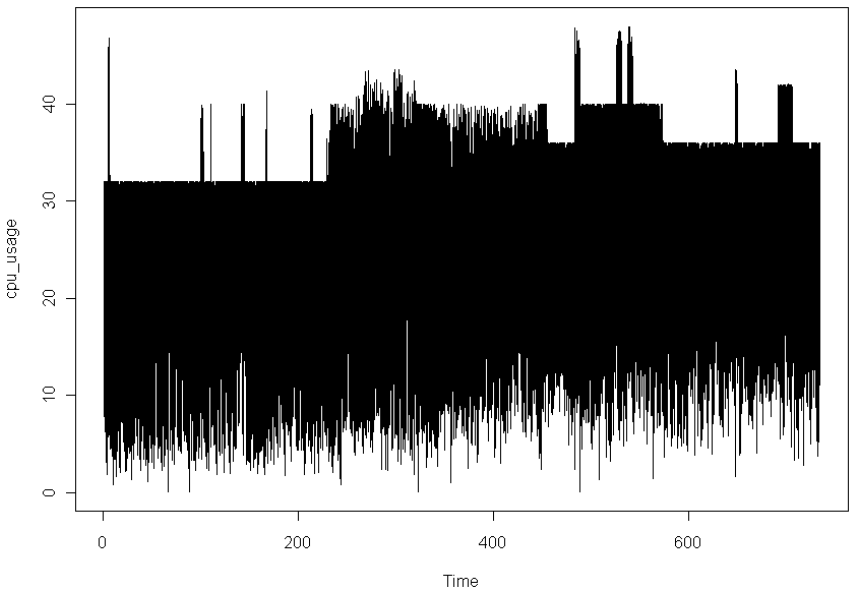


Figura 3.11 – Poder de processamento utilizado por uma determinada aplicação, entre 04-04-2008 e 14-05-2010. O eixo horizontal representa o tempo, em dias, desde a data da primeira observação. Entre cada unidade do eixo horizontal existem 288 observações, uma por cada 5 minutos do dia. O eixo vertical representa o número de microprocessadores utilizados.

Uma observação inicial é suficiente para detectar situações que se podem revelar problemáticas durante o processo de modelação. Uma delas é a presença de ciclos mais ou menos bem definidos, que parecem indicar a presença de um ou mais termos de sazonalidade. A este aspecto acrescenta-se uma tímida tendência de crescimento dos valores registados, a uma escala macroscópica. Por último, e este afigura-se como o problema mais grave, muitas das observações com valores mais elevados apresentam um comportamento suspeito. Estas observações apresentam valores idênticos durante largos intervalos de tempo, algo que não é expectável e que portanto deve ser analisado cuidadosamente.

O processo de análise iniciou-se precisamente com enfoque na resolução deste último aspecto, devido à sua especial relevância. A simples análise das observações não é suficiente para extrair conclusões fundamentadas, portanto foi necessário recorrer a outro tipo de informação. A experiência adquirida, conjugada com o conhecimento do problema em mãos,

aponta imediatamente para a necessidade de investigar a quantidade de recursos disponíveis durante os períodos temporais em causa. A análise destes dados veio, de facto, confirmar a hipótese levantada. Tal como referido anteriormente, os servidores da Direcção de Sistemas de Informação da Sonae alojam, cada um deles, múltiplas aplicações. A cada aplicação é associada uma parte dos recursos desse servidor, não podendo esta utilizar mais do que aqueles que lhe estão atribuídos. Esta última regra, porém, pode ser ignorada se as restantes aplicações com que partilha o servidor não estiverem a utilizar, ou não planeiam utilizar, os recursos que lhe estão associados. Nesta situação, estes recursos podem ser atribuídos temporariamente às aplicações que deles possam necessitar.

Durante todo o ano de 2008, a aplicação em estudo – Retek – teve à sua disposição 32 microprocessadores. Esta quantidade revelou-se escassa, de acordo com os dados registados, uma vez que em várias situações este número foi utilizada na totalidade. Uma vez que a) não havia mais recursos disponíveis (i.e. as outras aplicações necessitaram de todos os seus recursos) ou b) o administrador do sistema não sentiu necessidade de aumentar o número de recursos atribuídos à aplicação, esta nunca pôde – salvo raras excepções – ultrapassar o limite que lhe foi imposto. Durante grande parte do ano de 2009 o cenário foi substancialmente diferente, uma vez que o número de microprocessadores atribuídos ao Retek foi aumentado para 40. Se durante a primeira metade do ano este número se revelou suficiente – apesar de alguns valores esporádicos –, durante a segunda metade o cenário foi consideravelmente diferente, com um nível de utilização bem mais elevado, que voltou a revelar insuficientes os recursos existentes. O ano de 2010 trouxe uma redução no poder de processamento atribuído – para 36 microprocessadores –, algo que só veio agravar os problemas já detectados no período anterior.

Este aspecto tem graves consequências no processo de análise de necessidades de capacidade em curso, uma vez que muitas observações se tornam, por esta razão, inutilizáveis ou pouco confiáveis. Uma vez que estas observações se referem aos valores mais elevados passíveis de serem registados, é óbvio concluir que qualquer estimativa realizada com estes dados se apresentará defeituosa logo à partida. Urge, portanto, recorrer a um método que reduza os problemas a um mínimo, mas que mantenha ainda a capacidade de efectuar estimativas com a maior utilidade possível.

Uma análise aos dados existentes, mais precisamente ao histograma obtido a partir destes – Figura 3.12 –, parece indicar uma distribuição normal (ou aproximadamente normal) dos

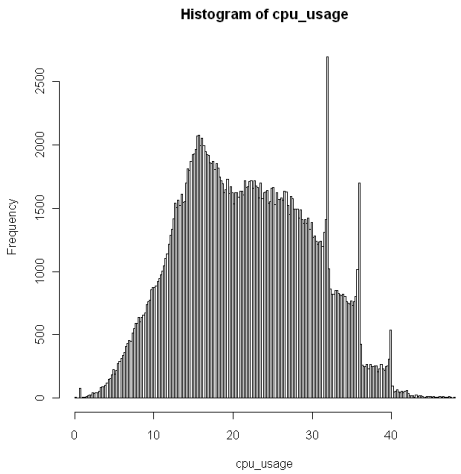


Figura 3.12 – Histograma dos valores registados da utilização do CPU, pela aplicação em estudo.

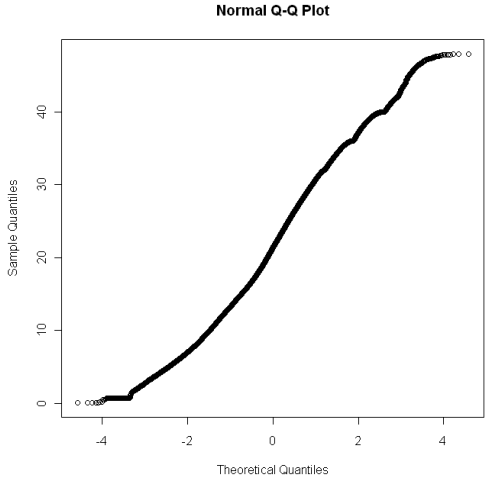


Figura 3.13 – Gráfico QQ. Permite comparar os valores registados da utilização do CPU com os esperados numa distribuição normal.

valores registados. Isto é confirmado observando o gráfico QQ (Figura 3.13) e recorrendo aos testes de normalidade de Shapiro-Wilk [52] e de D'Agostino-Pearson [53].

Análise dos percentis

Em função dos problemas detectados e dos resultados obtidos, optou-se por analisar o comportamento da série para cada dia do período de observações. As observações registadas em cada dia foram agrupadas, e foram calculados cinco valores, correspondendo cada um deles a um nível distinto de observações: os valores do 5º, 25º, 50º, 75º e 95º percentil das observações de cada dia.

Estes valores permitem minorar os efeitos das observações registadas durante os períodos de carência de poder de processamento, e que seriam superiores, ignorando 5% dos valores mais elevados de cada dia. O mesmo processo é efectuado em relação aos valores mais baixos (5º percentil), por uma questão de simetria. Saliente-se o facto de este procedimento não garantir a exclusão de todos os valores problemáticos; por esta razão, a análise dos resultados deverá ser sempre encarada com um espírito crítico. O 50º percentil corresponde à mediana dos valores registados em cada dia. Dado que o valor da mediana não se altera se o poder de processamento não estiver limitado em qualquer período, este valor é considerado na análise pelo seu grande valor estatístico. Os valores dos percentis 25 e 75 – 1º e 3º quartil, respectivamente – são também analisados, sendo fundamentalmente utilizados como valores de referência.

Depois de calcular os valores referidos, é analisada a sua evolução ao longo do período em estudo – ver Figura 3.14. É notória a similaridade entre as diferentes variáveis, com a excepção do 95º percentil, pelas razões já apresentadas, prova da normalidade da sua distribuição.

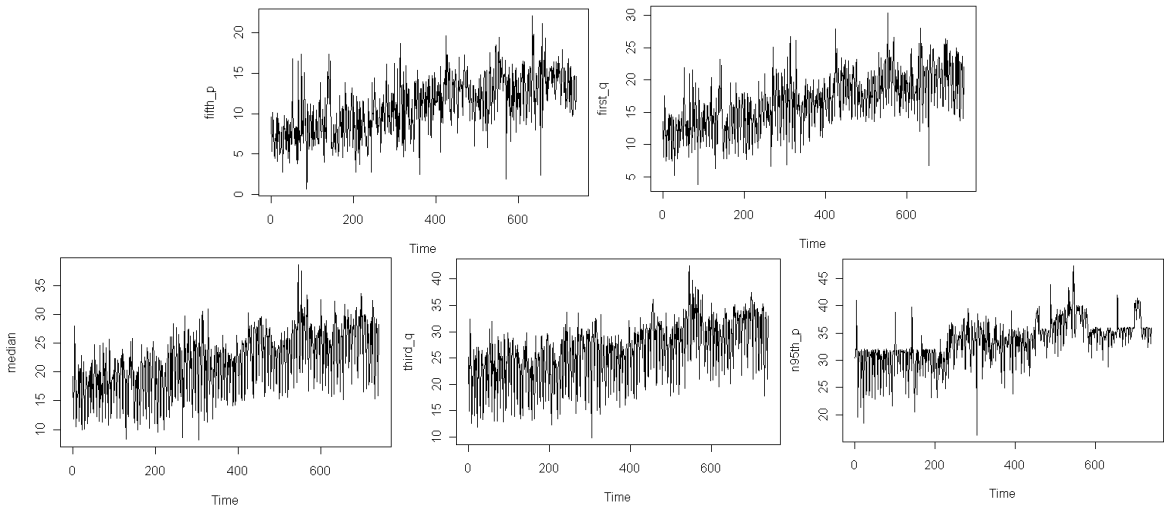


Figura 3.14 – Da esquerda para a direita, de cima para baixo, evolução do valor registado pelo 5º, 25º, 50º, 75º e 95º percentil. O eixo horizontal representa o tempo, em dias, desde a data da primeira observação. O eixo vertical representa o número de microprocessadores utilizados.

Uma análise aos gráficos de autocorrelação revela a óbvia tendência de crescimento. Estes mostram ainda uma forte relação entre os valores separados por sete períodos – uma semana –, algo que já era esperado – Figura 3.15. Por outro lado, esperava-se que os valores demonstrassem também indícios de sazonalidade com um intervalo temporal maior (trimestral, semestral ou anual), mas este pressuposto não pôde ser confirmado. Isto deve-se, com uma grande probabilidade, ao facto de o período de amostra dos dados não ser muito extenso, uma

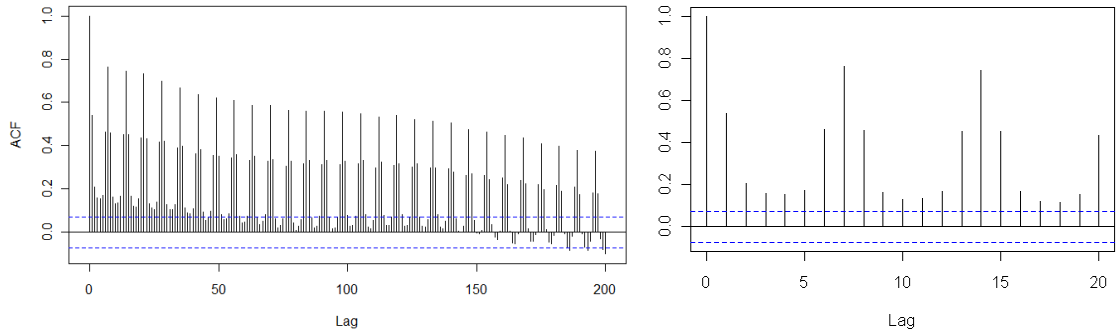


Figura 3.15 – Gráfico de autocorrelação da série que representa a evolução da mediana; à direita pode ver-se a mesma série, mas apenas para períodos inferiores a 20.

vez que só existem observações para um intervalo com pouco mais de dois anos, o que dificulta a detecção de estruturas com períodos mais alargados.

Tal como na análise das necessidades do espaço de armazenamento, demonstrada anteriormente, também aqui se pretende modelar a evolução das necessidades de poder de processamento em função do número de lojas existentes. Por esta razão, optou-se pelo mesmo processo utilizada nessa análise: dividir as lojas por grupos, em função das características relevantes para o processo de análise.

Dado que a sazonalidade detectada nas observações apresenta um período suficientemente pequeno para poder ser considerado de importância menor para a análise em curso, decidiu-se analisar unicamente a tendência de crescimento demonstrada. Assim, importa reduzir, ou retirar completamente, qualquer efeito sazonal que possa existir nas séries em análise. Um processo utilizado frequentemente com este propósito é a suavização com recurso ao algoritmo de Loess – ver secção 2.3.6. Este algoritmo possibilita a remoção dos efeitos sazonais presentes nas séries em análise, o que resulta na obtenção de um novo conjunto de séries, que representam a tendência geral de crescimento de cada uma delas. O método de Loess necessita de um valor para o seu factor de suavização; este valor não deverá ser demasiado pequeno, pois isso aumentaria a probabilidade de não excluir todos os elementos sazonais, mas também não deverá ser demasiado grande, pois quanto maior mais a curva obtida se assemelhará de uma linha recta. Por esta razão foram testados os valores 0.5, 0.75 e 1, tendo sido seleccionado o valor 0.75 após a avaliação dos resultados. As séries obtidas com a aplicação do algoritmo de Loess podem ser observadas na Figura 3.16.

As séries obtidas foram avaliadas positivamente, podendo ver-se que estas demonstram claramente a tendência de crescimento que está implícita nas séries em análise. Destaca-se apenas o resultado obtido com a série associada ao 95º percentil, que apresenta uma tendência negativa na sua parte final. Apesar de este valor fazer sentido quando se observam os dados disponíveis, deve haver sempre a preocupação em relembrar que estes valores, muito provavelmente, foram registados por defeito, uma vez que as limitações impostas não permitiram que fosse utilizada a quantidade necessária nesses períodos. Esperar-se-ia, em condições perfeitas, que esta série apresentasse uma evolução semelhante à apresentada pelas restantes.

Depois de calculadas as tendências de crescimento dos valores dos percentis em análise, foi estudada a correlação entre estes e a variação no número de lojas. Os resultados obtidos – Figura 3.17 – são francamente animadores, com valores que, na generalidade, rondam o valor 0,95, todos estatisticamente significativos a 5%.

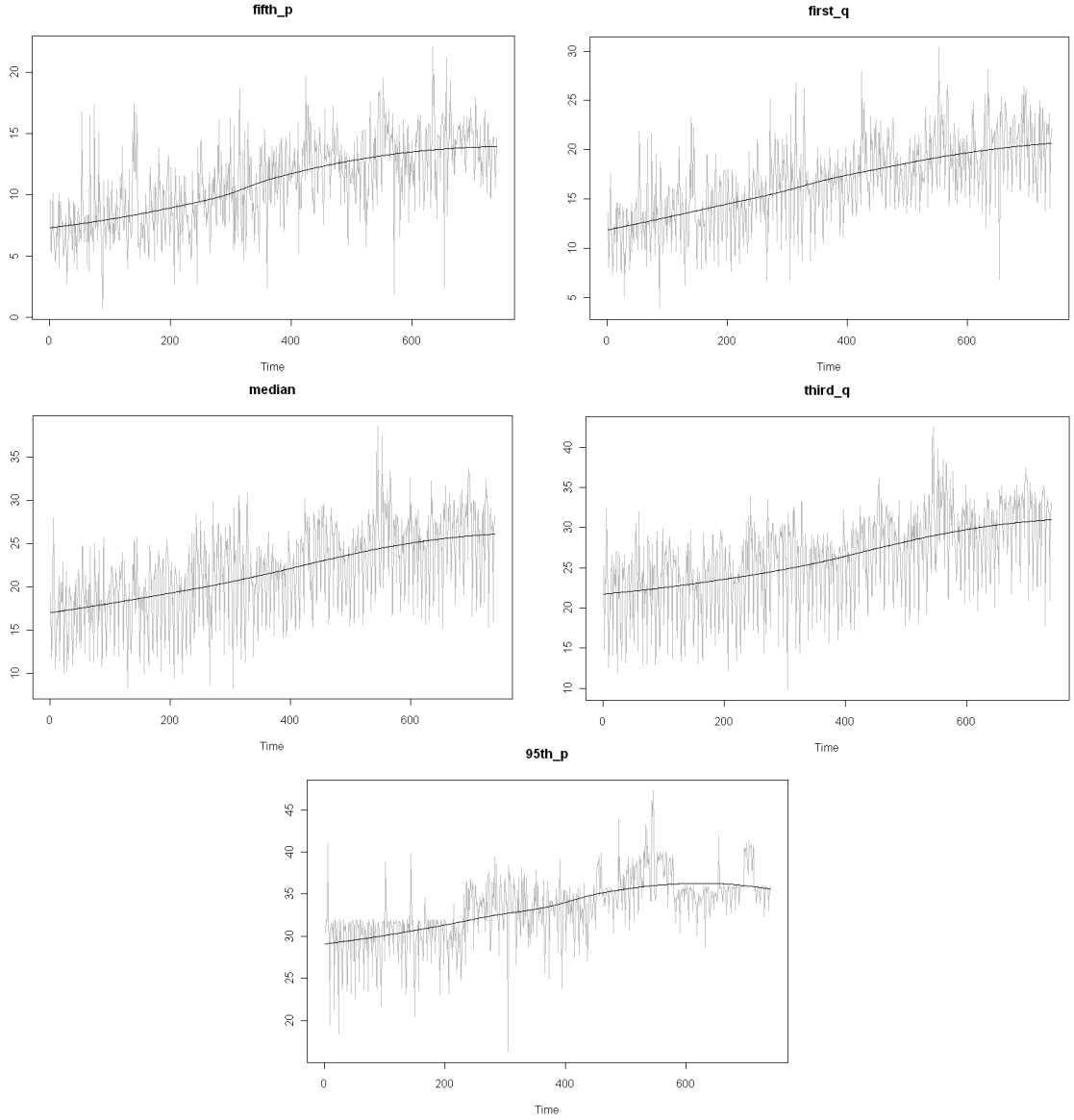
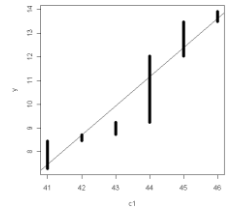
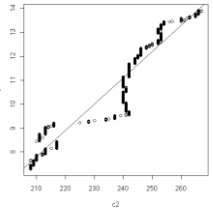
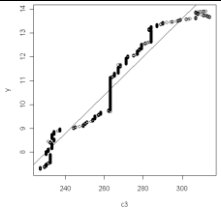
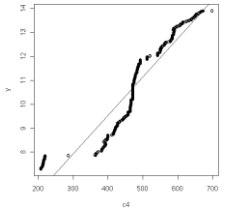


Figura 3.16 – Da esquerda para a direita, de cima para baixo: tendência de crescimento dos valores registados no 5º, 25º, 50º, 75º e 95º percentis, obtida com recurso ao método de Loess. O eixo horizontal representa o tempo, em dias, desde a data da primeira observação. O eixo vertical representa o número de microprocessadores utilizados.

Correlação	*Classe 1*	*Classe 2*	*Classe 3*	*Classe 4*
Percentil 5	 0,9510658	 0,9678532	 0,974001	 0,9546262

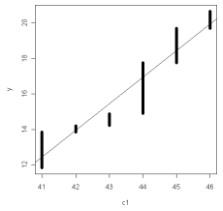
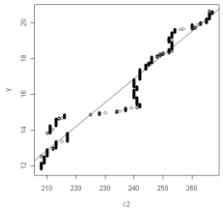
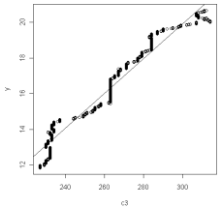
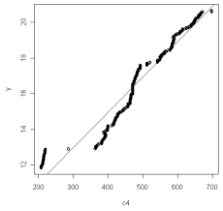
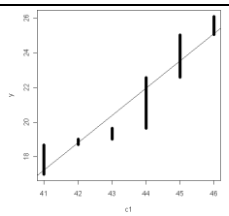
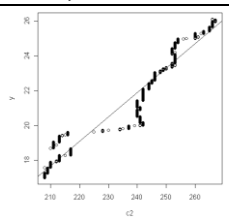
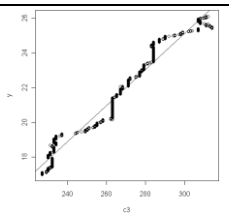
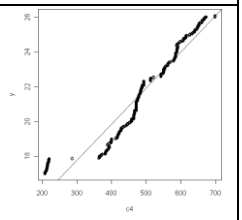
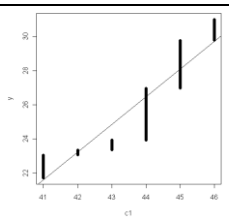
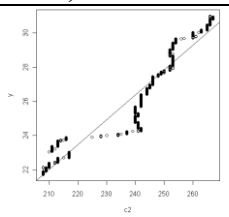
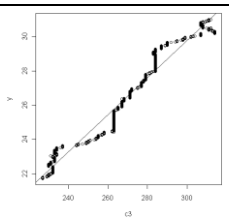
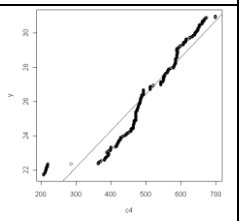
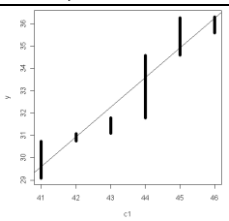
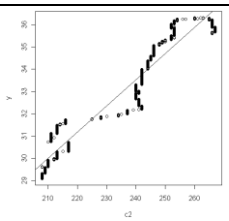
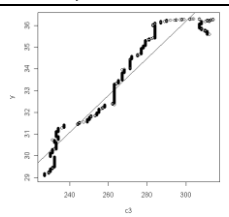
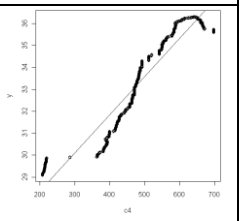
Percentil 25	 0,969935	 0,9722213	 0,9794917	 0,9690848
Percentil 50	 0,9479917	 0,9663926	 0,983231	 0,9617085
Percentil 75	 0,9362142	 0,9598563	 0,983157	 0,9516272
Percentil 95	 0,9541674	 0,9595803	 0,9588757	 0,9559877

Figura 3.17 – Gráficos de dispersão dos vários percentis da utilização do processador em função do número de lojas de cada classe, e coeficientes de correlação entre cada percentil e cada classe de lojas.

Regressão linear

Procedeu-se, então, à tentativa de modelação destas séries com recurso a uma técnica de regressão linear. O modelo seleccionado para iniciar este processo foi, à semelhança do realizado na análise da utilização do espaço em disco,

$$y_t = \phi_0 + \phi_1 x_{1t} + \phi_2 x_{2t} + \phi_3 x_{3t} + \phi_4 x_{4t} + \varepsilon_t \ . \tag{3.7}$$

Note-se a ausência propositada de um termo dependente do tempo, quando comparado com a expressão final obtida para o modelo do espaço em disco. Esta decisão está intrinsecamente relacionada com conhecimento empírico do sistema em estudo: quando não abrem novas lojas, o espaço em disco necessário continua a aumentar, enquanto as necessidades de poder de processamento se mantêm num nível constante.

Recorrendo ao ambiente de computação estatística R, calcularam-se os valores dos parâmetros do modelo, através do método dos mínimos quadrados, de forma análoga à utilizada na análise da evolução do espaço em disco. Os valores obtidos foram aplicados à expressão (3.6), obtendo-se as séries representadas na Figura 3.18.

Representando num único gráfico – Figura 3.19 – todas as séries obtidas através do processo de regressão linear, e adicionando ainda a série das observações registadas, podemos observar de uma forma mais clara a evolução estimada dos vários percentis em análise e ver mais facilmente a forma como estas séries ajudam a explicar a evolução das necessidades de poder de processamento.

Depois de calculadas estas séries, consideraram-se as quantidades de novas lojas com data de abertura agendada para qualquer dia do período que se pretende estimar. Estes novos dados permitem calcular – estimar – os valores dos vários percentis para cada dia de um determinado período no futuro. Recorrendo ao método estendido de Pearson-Tukey [54] é possível calcular a variância dos valores da utilização do CPU em cada instante do período de previsão, o que possibilita o cálculo de intervalos de confiança para esta.

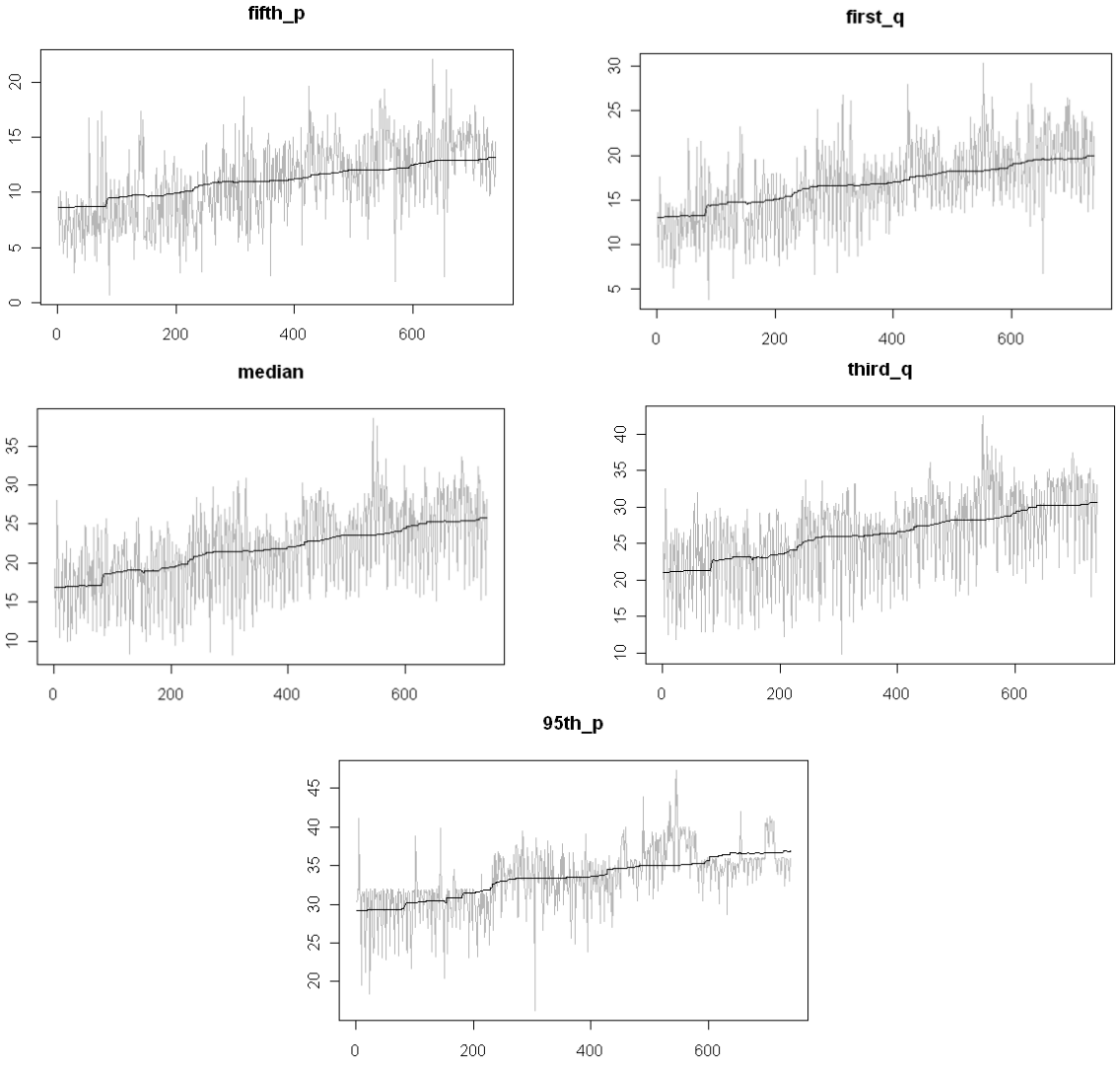


Figura 3.18 – Da esquerda para a direita, de cima para baixo: tendência de crescimento dos valores registados no 5º, 25º, 50º, 75º e 95º percentis, obtida com recurso ao método de regressão linear. O eixo horizontal representa o tempo, em dias, desde a data da primeira observação. O eixo vertical representa o número de microprocessadores utilizados.

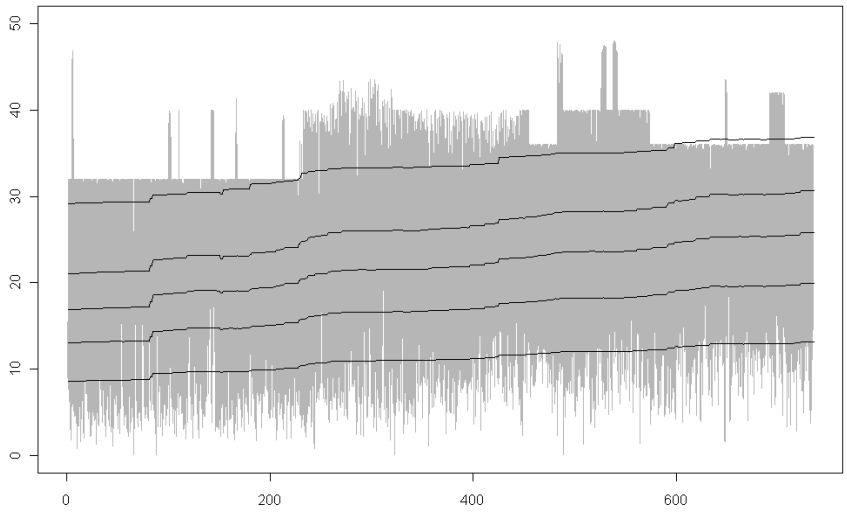


Figura 3.19 – Tendência de crescimento dos valores registados no 5º, 25º, 50º, 75º e 95º percentis, obtida através da utilização de um algoritmo de regressão linear. O eixo horizontal representa o tempo, em dias, desde a data da primeira observação. O eixo vertical representa o número de microprocessadores utilizados.

Considerações adicionais

De forma análoga à realizada no processo de análise da evolução da utilização do espaço em disco, efectuou-se todo o processo de análise da evolução da utilização do poder de processamento para apenas uma aplicação – Retek –, sendo este descrito ao longo desta secção. Infelizmente não foi possível testar o método com outras aplicações, por dificuldades de acesso aos dados destas ou porque estes, pura e simplesmente, não existem.

A aplicação das metodologias descritas nesta secção a outras aplicações permitiria validar a adequação destas, ou indicar alternativas para a resolução de eventuais problemas. Vários cenários se poderiam colocar relacionados com os dados de outras aplicações: se os dados apresentarem um comportamento similar aos do Retek, com valores que são sujeitos a limites inferiores às necessidades, os resultados serão similares aos obtidos nesta análise; por outro lado, se os valores registados forem "bem comportados", sem valores alterados pela imposição de limites, então prevê-se que o método indicado seja capaz de obter melhores resultados, embora existam técnicas mais simples que provavelmente também o farão, nomeadamente a análise apenas dos valores máximos registados; por último, se as evoluções registadas forem substancialmente diferentes, então espera-se que o processo utilizado não tenha sucesso nestes casos.

3.4 Apreciação global

A modelação da evolução da utilização do espaço em disco decorreu sem problemas de maior, tendo sido obtidos resultados semelhantes aos esperados. A adopção de um modelo de regressão linear revelou-se portanto bastante eficaz neste caso, em grande parte devido ao comportamento das séries, que apresentam tendências de crescimento bem definidas, em geral, e uma quase total inexistência de ciclos oscilatórios – termos ou indícios de sazonalidade.

Por outro lado, o processo de modelação para a evolução da utilização do poder de processamento exigiu a conjugação de diferentes técnicas na prossecução de resultados mais satisfatórios. A principal razão foi a fraca qualidade dos dados disponíveis para a análise, que apresentam "falsos" valores máximos – valores que seriam certamente superiores se não existisse um limite definido para a utilização máxima do poder de processamento. O método a que se recorreu tira partido da distribuição aproximadamente normal evidenciada pelos dados recolhidos, analisando a evolução de vários percentis da série para estimar o valor máximo passível de ser alcançado, dentro de um determinado intervalo de confiança.

Salienta-se o facto de, para ambos os modelos, não se poderem interpretar os resultados obtidos como valores exactos. Ao invés disso, estes valores devem ser sempre analisados em conjunto com o intervalo de confiança a que estão afectos, e deverão ser compreendidos e tomados em consideração todos os mecanismos inerentes ao processo de cálculo realizado, e que resultaram nos valores obtidos.

Capítulo 4

Implementação

O processo de modelação descrito no capítulo anterior foi implementado numa aplicação de software, de acordo com o requerido pela Direcção de Sistemas de Informação da Sonae, responsável pelo projecto associado ao presente documento. Neste capítulo detalham-se os vários aspectos relacionados com o processo de implementação da referida aplicação.

4.1 Linguagem e ambiente de desenvolvimento

O desenvolvimento desta aplicação obrigou à escolha de uma linguagem capaz de satisfazer os objectivos definidos pelo proponente do projecto, nomeadamente a capacidade de proporcionar uma interface agradável para o utilizador. A linguagem escolhida deve ser capaz de aceder sem dificuldades a uma base de dados Oracle, onde serão armazenadas as observações realizadas sobre as variáveis de capacidade. Requisitos adicionais como a robustez e a familiaridade com a linguagem foram também tomados em consideração no processo de escolha. Por estas razões, optou-se por implementar a aplicação em Java, tirando partido das suas principais características, particularmente o facto de ser uma linguagem orientada a objectos e possuir várias *frameworks* que possibilitam a fácil implementação de interfaces, tal como a AWT ou a SWING.

A aplicação foi desenvolvida no sistema operativo Microsoft Windows XP, e foi utilizado como IDE o Eclipse Galileo durante a fase de implementação. A aplicação recorre ainda ao ambiente de computação estatística R para efectuar os cálculos mais complexos, utilizando uma interface especificamente desenvolvida para o efeito. A versão do R utilizada foi a 2.10.1.

4.2 Arquitectura da aplicação

A aplicação foi desenvolvida com o padrão de arquitectura *Model-View-Controller* (MVC) em mente. Este padrão foi apresentado em 1979 por Trygve Reenskaug, durante a sua estadia no centro de investigação da Xerox, enquanto parte do grupo Smalltalk [55]. Este padrão de arquitectura define três conceitos:

- Modelo (*model*) – é utilizado para armazenar os dados que suportam a aplicação, bem como a lógica associado ao seu domínio.
- Vista (*view*) – é responsável pela renderização do modelo, de uma forma que possibilite a interacção com este. A vista acede aos dados do modelo e especifica como é que estes são apresentados.
- Controlador (*controller*) – traduz interacções realizadas sobre a vista para acções a serem executadas no modelo. É ainda responsável por seleccionar a vista mais adequada em função das interacções realizadas pelo utilizador e dos outputs das acções do modelo.

Assim, partindo destes conceitos, é possível representar a aplicação desenvolvida com o seguinte diagrama de arquitectura:

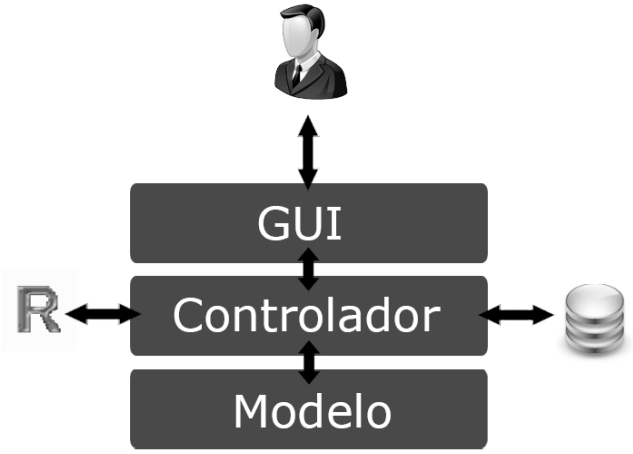


Figura 4.1 – Diagrama de arquitectura da aplicação.

É fácil verificar a importância do controlador nesta aplicação, sendo este o responsável por "fazer a ponte" entre a interface (GUI) e o modelo de dados, mas também pelo acesso à base de dados de capacidade e pela comunicação com o ambiente de computação estatística.

Na aplicação desenvolvida estes componentes estão claramente identificados. O modelo é implementado recorrendo a uma série de classes responsáveis por armazenar os diferentes tipos de dados, e por realizar uma série de acções sobre estes. Tomemos como exemplo a estrutura de classes utilizada para armazenar as observações relacionadas com o poder de processamento:

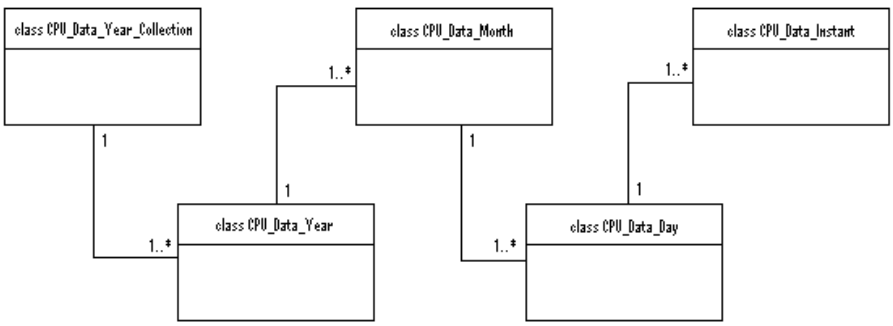


Figura 4.2 – Parte do diagrama de classes da aplicação. Na figura estão representadas as classes que armazenam os dados relacionados com a utilização do CPU e que implementam métodos que operam sobre estes.

O diagrama apresentado na Figura 4.2 é facilmente interpretado, se se tomarem em atenção os seguintes pontos: a classe *CPU_Data_Instant* é o elemento básico desta estrutura, sendo responsável por armazenar a informação relacionada com uma observação, ocorrida num determinado instante; a classe *CPU_Data_Day* possui como um dos seus atributos um conjunto de objectos *CPU_Data_Instant*, ou seja, informação relacionada com todas as informações ocorridas num determinado dia, possuindo ainda métodos que lhe permitem operar directamente com esta informação e realizar uma série de cálculos; as classes *CPU_Data_Month* e *CPU_Data_Year* têm uma estrutura similar à da classe *CPU_Data_Day*, mas estão relacionadas com os meses e os anos, respectivamente; por último, a classe *CPU_Data_Year_Collection* é, basicamente, o conjunto de todos os anos que possuem observações e que estão a ser considerados pela aplicação. Existem estruturas similares para armazenar a informação relativa à utilização do espaço em disco e às lojas existentes, sendo esta última substancialmente mais simples.

A vista da aplicação é, fundamentalmente, responsabilidade de uma única classe, que implementa uma interface recorrendo às funcionalidades e componentes disponibilizados pela framework SWING.

O controlador é também ele implementado numa única classe, sendo responsável por definir e controlar o fluxo de execução da aplicação, bem como pelo processamento das interacções entre o utilizador e a aplicação, através da interface.

A Figura 4.3 mostra o diagrama de classes da aplicação, simplificado.

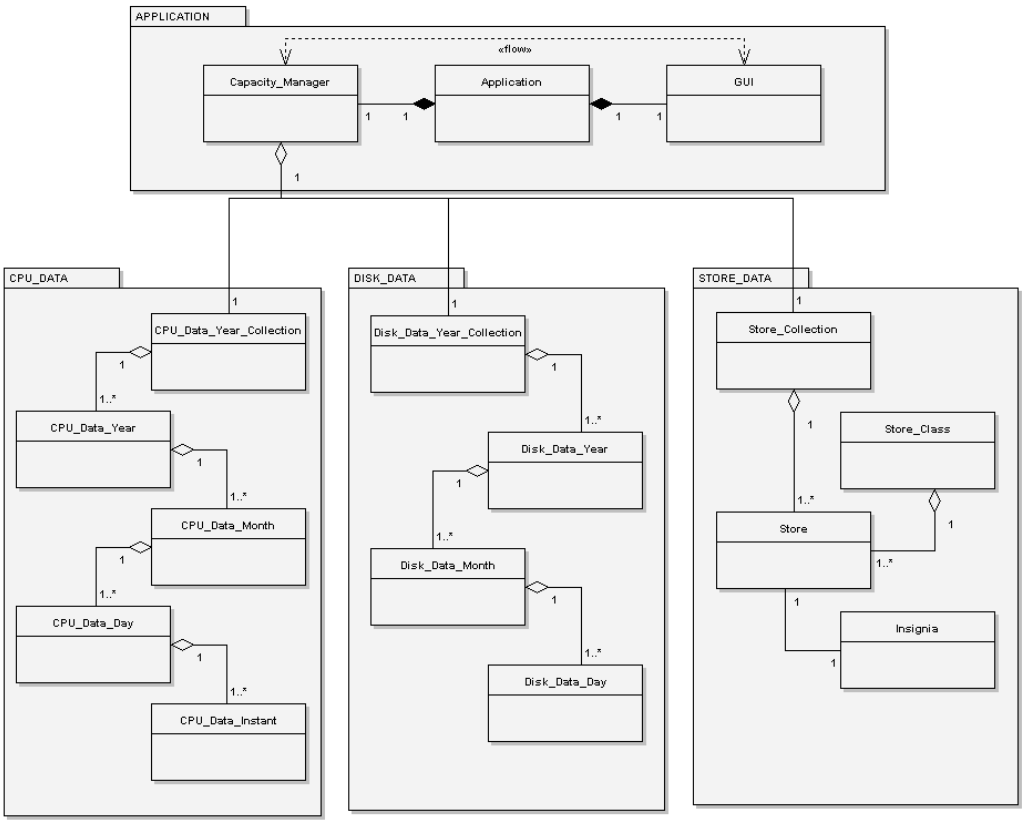


Figura 4.3 – Diagrama de classes da aplicação (simplificado).

4.3 Visão geral da aplicação

Nesta secção apresenta-se uma visão geral da aplicação implementada, com descrição dos mecanismos que lhe estão associados, acompanhada com *screenshots* recolhidos durante a sua execução para uma melhor compreensão.

Sendo uma aplicação desenvolvida em Java, esta encontra-se distribuída em formato JAR (**J**ava **AR**chive) para uma maior portabilidade. Isto permite a sua execução em ambientes e sistemas operativos diversos, tais como Microsoft Windows ou Linux. A execução desta aplicação está dependente, no entanto, da existência de uma instalação local do ambiente de computação estatística R, aplicação open-source disponibilizada em *http://www.r-project.org*.

Após lançar a aplicação, é requerida ao utilizador a sua autenticação através de um par *username-password*. Esta autenticação é requerida por uma questão de segurança, permitindo o acesso aos dados existentes na base de dados remota que suporta a aplicação.

Uma vez validada a autenticação, é apresentado ao utilizador um formulário que lhe permite especificar qual a aplicação que se pretende estudar, as datas entre as quais devem ser analisadas as observações, e o período de estimação pretendido, para cada uma das duas variáveis de capacidade consideradas pela aplicação. A Figura 4.4 mostra o formulário que é apresentado ao utilizador nesta fase.

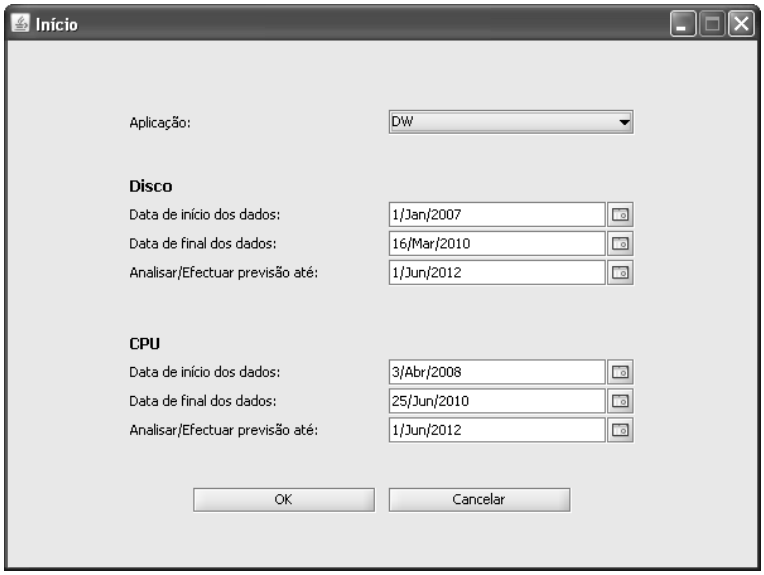


Figura 4.4 – Ecrã inicial da aplicação. Permite a definição da aplicação a estudar e dos intervalos de tempo que lhe estão associados.

Após a definição dos parâmetros requeridos e respectiva confirmação, é apresentado um segundo formulário ao utilizador – Figura 4.5 –, onde este poderá proceder à redistribuição das insígnias existentes pelas quatro classes de lojas consideradas pela aplicação. Este formulário é apresentado ao utilizador já com uma configuração por defeito, para que este processo seja o menos moroso possível. Adicionalmente, é também esperado que o utilizador especifique

factores de proporção entre as diferentes classes[3], para que o modelo a ser construído possa ter em consideração as diferenças de dimensão entre estas. Este ecrã de configuração é necessário para que a ferramenta possa suportar um maior número de aplicações, por várias razões, nomeadamente os diferentes propósitos de cada uma, o que faz com que as relações entre as insígnias possam variar.

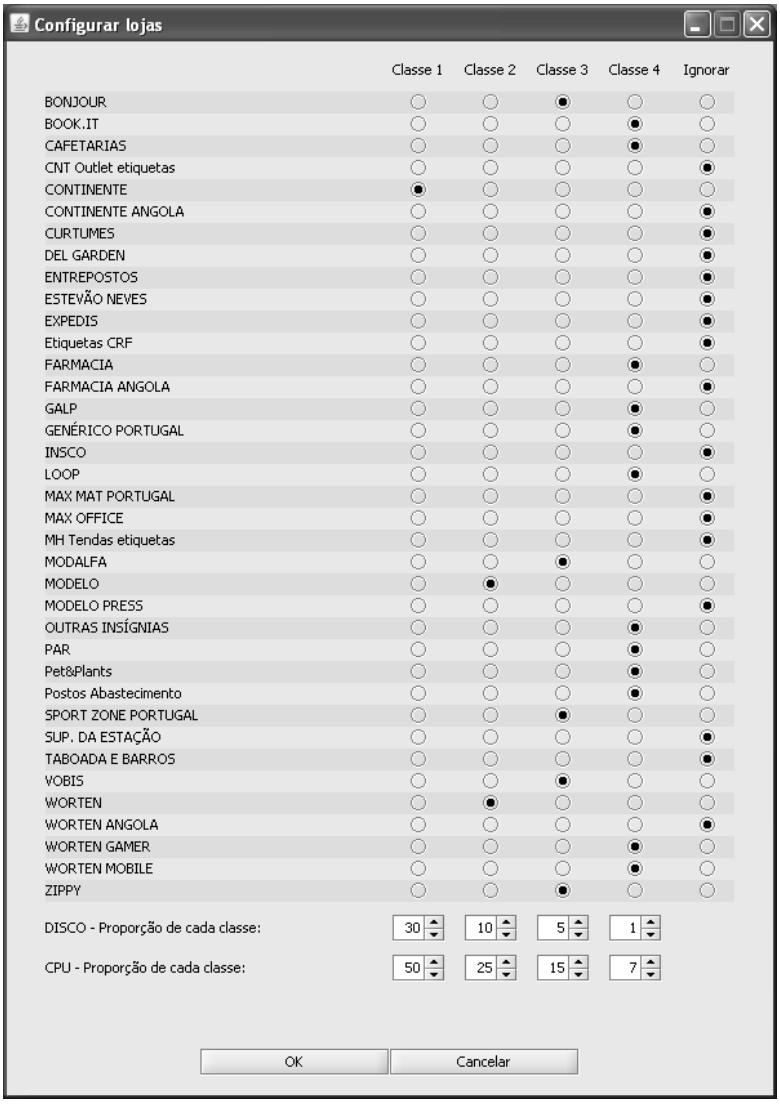


Figura 4.5 – Ecrã de configuração da aplicação. Permite o estabelecimento de relações entre as várias insígnias do portfólio da Sonae.

Enquanto o formulário representado na Figura 4.5 é apresentado ao utilizador, a aplicação lança uma segunda thread em background que inicia a transferência da informação presente na base de dados associada que suporta esta ferramenta, de acordo com o especificado pelo utilizador no ecrã inicial – Figura 4.4 –, reduzindo assim eventuais tempos de espera. Quando o utilizador valida a informação introduzida no formulário da Figura 4.5, sucede uma de duas coisas: se a informação da base de dados já foi totalmente transferida, é imediatamente

[3] Ver secção 3.2 deste documento.

efectuada uma série de cálculos e operações sobre estes; caso contrário (leia-se, se a informação ainda não tiver sido totalmente transferida da base de dados) a aplicação espera que o processo de transferência termine e só então procede à realização dos cálculos e das operações necessárias. Estas operações são:

- preenchimento das estruturas que dão corpo ao modelo da aplicação (aberturas de lojas, níveis de utilização de CPU e disco);
- a computação de valores necessários para cálculos mais complexos, e.g. mediana da utilização do poder de processamento;
- cálculo do número de lojas existentes/abertas em cada dia do período de análise;
- construção de *scripts* para serem executados no ambiente de computação estatística R, i.e. a ferramenta analisa os parâmetros introduzidos pelo utilizador e, em função destes, escreve os *scripts* que serão responsáveis pela execução dos cálculos mais complexos, recorrendo ao R;
- dá ordem de execução ao R, recolhendo os outputs da sua computação;
- utiliza estes resultados para o cálculo das séries que irão representar as previsões de evolução das variáveis de capacidade;

Uma vez concluída esta etapa de cálculos, a ferramenta apresenta ao utilizador o seu ecrã principal, onde se podem ver, entre outras coisas, as estimativas da evolução das variáveis de capacidade e os respectivos intervalos de confiança.

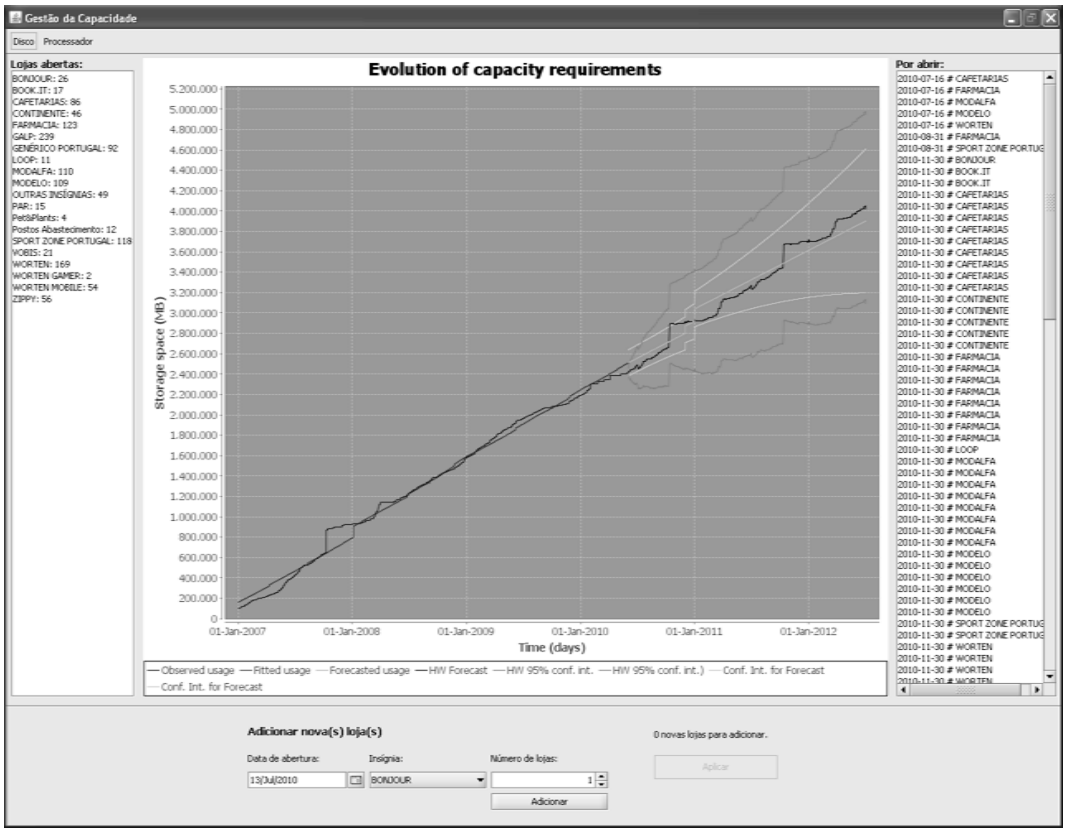


Figura 4.6 – Ecrã principal da aplicação. Nesta figura pode ser vista a análise relativa à utilização do disco.

Neste ecrã pode ser visto: ao centro, o gráfico que mostra, para a aplicação em estudo, a evolução prevista das necessidades do recurso de capacidade em causa; à esquerda, a

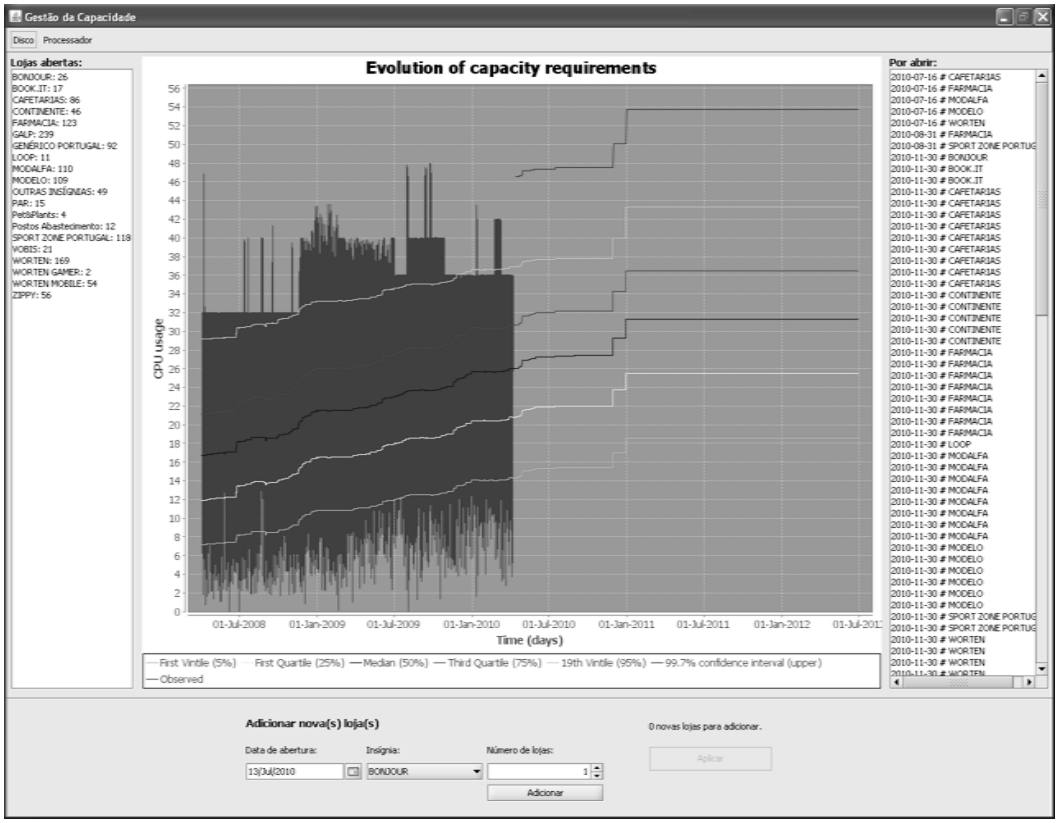


Figura 4.7 – Ecrã principal da aplicação. Nesta figura pode ser vista a análise relativa à utilização do processador.

quantidade de lojas existentes, de cada insígnia, na data final do período de observações; no lado direito do ecrã são mostradas as lojas com data de abertura contida no intervalo de tempo que se pretende estimar; em baixo, um formulário permite a introdução de novas lojas no período de estimativas, para que se possa analisar os seus impactos e as suas implicações em termos de recursos de capacidade. De notar que estas novas lojas – virtuais - não devem ser confundidas com as que estão previstas no plano de crescimento da empresa – reais – apesar de ambas serem consideradas na visualização de forma idêntica. A informação relativa às lojas "reais" deverá estar presente na base de dados que suporta a aplicação, sendo carregada automaticamente por esta em tempo de execução.

O fluxo de execução da aplicação pode ser observado, de forma simplificada, na seguinte figura:

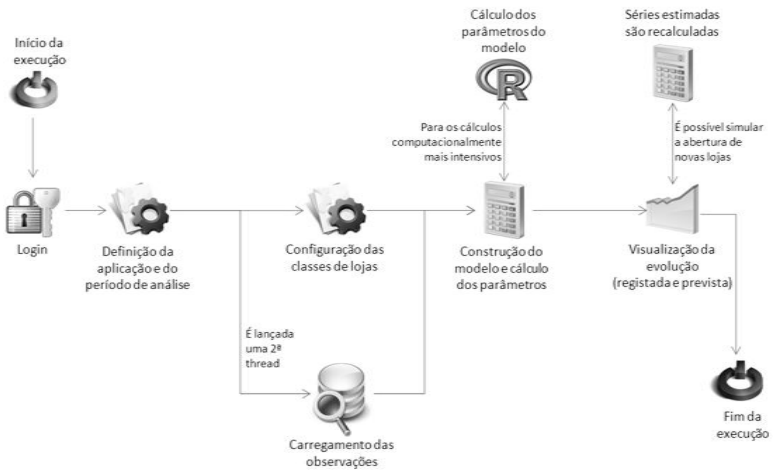


Figura 4.8 – Fluxo de execução da aplicação.

4.4 Observações e apreciação geral

O processo de implementação termina e é avaliado com a aplicação que materializa os processos e conceitos definidos durante a etapa de modelação.

Esta aplicação permite a visualização gráfica da evolução dos níveis de utilização das variáveis de capacidade em estudo – espaço em disco e poder de processamento –, e das respectivas estimativas de evolução futura, obtidas de acordo com o processo descrito no capítulo 3 desta dissertação.

Ao permitir a parametrização das variáveis de input – número de lojas – foi assumido um compromisso que sacrifica a autonomização completa da tarefa de previsão por uma maior versatilidade da ferramenta. Esta versatilidade reflecte-se num maior número de aplicações passíveis de serem analisadas com esta ferramenta.

Por último, ao possibilitar a simulação da abertura de novas lojas – em adição às já existentes no plano de crescimento da empresa – a ferramenta permite a satisfação de outro dos objectivos especificados, a avaliação da escalabilidade da infra-estrutura, na medida em que permite estimar o número de novas lojas que o sistema actual conseguirá suportar sem perdas de desempenho ou sem falhas de capacidade.

Como demonstrado, a ferramenta de previsão implementada cumpre os propósitos para o qual foi desenhada e que foram especificados no arranque do projecto. No decorrer do processo de implementação foram surgindo ideias para funcionalidades adicionais a incluir na ferramenta. Algumas destas ideias foram incluídas e estão já presentes na versão do produto disponível à data da escrita do presente documento, enquanto outras foram remetidas para futuras intervenções na aplicação.

Capítulo 5

Conclusões e trabalho futuro

O processo que tomou forma no presente projecto não se esgota com o trabalho apresentado. A Gestão da Capacidade obriga a um acompanhamento constante e a uma monitorização contínua dos recursos utilizados, não sendo suficiente a utilização de uma simples ferramenta a solução para os potenciais problemas. Desta forma, reforça-se mais uma vez a necessidade de encarar a aplicação desenvolvida durante este projecto apenas como uma ferramenta que o agente de decisão terá ao seu dispor para que possa actuar de uma perspectiva mais informada. Este utilizador terá de estar consciente das vantagens e desvantagens que possam advir da utilização da referida aplicação.

Neste capítulo sumarizam-se os resultados obtidos com a ferramenta de previsão desenvolvida no projecto descrito ao longo deste documento, apontando-se os pontos passíveis de serem melhorados com vista a um potencial incremento da qualidade do produto final.

5.1 Satisfação dos objectivos

Os resultados obtidos com a ferramenta de previsão desenvolvida foram, de uma forma geral, considerados bastante positivos. São reconhecidas as limitações das estimativas realizadas, sendo algumas consequência do método adoptado, e outras da qualidade dos dados disponíveis para análise.

A verificação e validação dos métodos desenvolvidos foi um elemento que não recebeu a devida atenção. Isto deve-se, fundamentalmente, ao reduzido tempo de duração do projecto. Este tempo – 4 meses – tornou impossível a realização de testes com dados reais, pois as variações dos níveis de utilização num período de tempo desta dimensão não são suficientes para revelar eventuais carências do modelo implementado. Uma alternativa a esta situação seria a divisão das observações existentes em dois grupos: um para a construção do modelo e outro para o seu teste. No entanto, salvo algumas excepções, os dados armazenados em histórico cobriam apenas cerca de dois anos completos de utilização, um valor um pouco reduzido para poder recorrer a este processo sem qualquer dificuldade. De facto, em algumas situações, este período de amostra reduzido foi suficiente para produzir resultados acertados, mas isto ocorreu

apenas nas séries que apresentavam um comportamento claro, com uma tendência de crescimento constante e sem qualquer efeito cíclico ou aleatoriamente oscilatório. Nas outras situações, de uma forma geral, a curva obtida tendia a afastar-se lentamente da série das observações à medida que esta progredia no tempo.

O facto de as observações existentes serem relativamente reduzidas tem ainda impacto noutros aspectos. Métodos como o Box-Jenkins, que procuram detectar e identificar flutuações sazonais que possam estar implicitamente presentes nas observações, apresentaram uma performance muito baixa durante o período de análise. O conhecimento empírico transmitido pelos colaboradores da Sonae que foram consultados indiciava um claro efeito sazonal com periodicidade anual, no entanto não foi possível confirmar este pressuposto apenas com a análise dos dados existentes. O método de Box-Jenkins, por exemplo, necessita de pelo menos dois períodos completos de dados, em condições ideais, para conseguir efectuar uma tentativa de modelação. No entanto, como já foi mostrado, as condições não eram as ideais neste caso, e por essa razão os resultados obtidos com este método ficaram aquém do esperado. De facto, apesar de ser um método extremamente complexo, alguns dos seus resultados foram inclusivamente piores do que os obtidos com recurso ao método de Holt (*Double Exponential Smoothing*), mesmo tendo em conta a relativa simplicidade deste último.

O facto de não ter sido possível testar a metodologia utilizada na análise da evolução da utilização do poder de processamento faz com que não seja possível a sua validação, devendo os seus resultados ser cuidadosamente analisados, particularmente se forem relativos a outras aplicações que não o Retek.

A aplicação desenvolvida foi também avaliada satisfatoriamente. Os seus requisitos foram cumpridos na integralidade, resultando num produto final que é simples de utilizar e cujos resultados são facilmente interpretáveis. A versatilidade da aplicação, e o facto de poder ser utilizada para analisar vários sistemas diferentes é sem dúvida um ponto positivo do produto implementado. Esta funcionalidade, no entanto, impediu que a aplicação apresentasse um comportamento completamente autónomo, tornando-a dependente de um input do utilizador para cumprir o seu propósito – ver secção 3.2.

5.2 Trabalho futuro

Concluído o tempo disponível para o desenvolvimento do projecto, foram identificados os elementos que, por uma razão ou por outra, poderiam ser melhorados.

O primeiro elemento a merecer a atenção prende-se com a estimativa da utilização do processador. É necessário testar as metodologias adoptadas com os dados provenientes de outras aplicações, para assim se poder validar o processo e aferir a sua qualidade.

A necessidade de conciliar o estudo de metodologias de previsão e a sua aplicação aos dados existentes, com a implementação de uma aplicação de software que as utiliza, num período de tempo relativamente reduzido, limitou o número de alternativas que poderiam ter sido utilizadas e testadas para a resolução do projecto. As metodologias seleccionadas e estudadas foram aquelas que, de uma forma generalizada, são as mais utilizadas para lidar com a problemática da gestão da capacidade. No entanto há muitas outras técnicas que poderiam ter sido consideradas durante a fase de análise e que não o foram unicamente por razões logísticas. Teria sido extremamente interessante comparar os resultados obtidos pelo método clássico de regressão linear com os resultantes da aplicação de uma rede neuronal desenvolvida para lidar com este problema. Da mesma forma, seria também interessante testar a utilização de

algoritmos genéticos na determinação dos valores dos parâmetros de um modelo, por exemplo, por oposição a técnicas como o método dos mínimos quadrados. Em suma, seria importante o teste de metodologias que adoptem abordagens distintas daquelas utilizadas pelos algoritmos estudados.

Neste momento, a aplicação desenvolvida possui apenas suporte para a análise de dois tipos de recursos – poder de processamento e espaço de armazenamento. Um melhoramento futuro poderia passar pela inclusão de suporte para outras variáveis de capacidade, tais como largura de banda ou memória RAM, entre outros.

Futuramente pretende-se que a ferramenta seja capaz de gerar relatórios que indiquem, de forma intuitiva e para cada variável de capacidade, qual o nível de utilização dos recursos previsto em vários horizontes temporais distintos. Este relatório apresentará a informação de uma forma visual, recorrendo a um esquema de cores tipo semáforo para representar a percentagem de utilização da variável de capacidade em causa (e.g. menos de 70% – verde, entre 70% e 90% – amarelo, mais de 90% – vermelho).

Um outro ponto que merece a atenção e que poderá vir a ser implementado futuramente visa eliminar a necessidade de recorrer ao input do utilizador para efectuar a distribuição das insígnias pelas classes, e a razão de proporção entre estas. Este procedimento poderá ser eliminado com a implementação de um algoritmo de *clustering* que analise dados relevantes para as aplicações em estudo (e.g. volumes de facturação para o DW, dimensões das gamas de produtos para o Retek) e que efectue autonomamente o processo de alocação das insígnias às classes. Este processo foi estudado brevemente durante a fase de implementação, mas a sua incorporação foi rejeitada devido às restrições temporais a que o projecto estava sujeito.

A performance da aplicação, em termos de tempos de execução, pode ainda ser bastante optimizada. A etapa de cálculo é um ponto crítico da ferramenta, e o grande volume e complexidade das operações realizadas nesta fase originam tempos de espera que ascendem a cerca de um minuto, ou até mais, no caso das aplicações com um grande número de observações. Foram identificadas claramente as operações que mais contribuem para este factor, sendo a optimização destas um dos objectivos prioritários de uma previsível revisão futura.

Por último, o facto de a aplicação depender de uma instalação local do ambiente de computação estatística R poderá merecer a atenção no futuro. A implementação das rotinas e dos algoritmos utilizados pelo R no código da aplicação foi uma hipótese descartada durante a fase de desenvolvimento, uma vez que estas se encontram de tal forma optimizadas que uma eventual transposição destas para Java acarretaria uma inevitável deterioração da performance. Teriam de ser realizados testes de performance mais detalhados, para avaliar a exequibilidade destes cálculos sem recurso a uma aplicação externa, partindo do pressuposto que o desempenho será sempre inferior nestes casos.

Referências

[1] Microsoft Corporation, “Application Architecture for .NET: Designing Applications and Services”, *MSDN Library*. Disponível em http://msdn.microsoft.com/en-us/library/ /ee817664.aspx. Acesso em 26/Janeiro/2010.

[2] W. Eckerson, “Three-Tier Client/Server Architecture: Achieving scalability, performance, and efficiency in client server applications", em *Open Information Systems, Janeiro de 1995.*

[3] “Capacity and Performance Management: Best Practices White Paper”, Cisco Systems, Outubro de 2005.

[4] H. Dixon, *Excel 2007 – Beyond the Manual*, cap. 9, pp. 171-186, Apress, Março de 2007.

[5] M. Roughan, *“An Application of Martingales to Queueing Theory (PhD. Thesis)”*, Department of Applied Mathematics, University of Adelaide, 1994.

[6] B. Urgaonkar et al, “An Analytical Model for Multi-Tier Internet Services and its Applications”, em *Proceedings of the 2nd IEEE International Conference on Autonomic Computing*, Seattle, Junho de 2005.

[7] R. Cooper, *Introduction to Queueing Theory, Second Edition*, cap. 1, North Holland, New York, 1981.

[8] M. Zukerman, *Introduction to Queueing Theory and Stochastic Teletraffic Models*, pp. 71-75, 2000.

[9] J. Beasley, “Operations Research Notes”. Disponível em http://people.brunel.ac.uk/ /~mastjjb/jeb/or/contents.html. Acesso em 29/Janeiro/2010.

[10] D. Villela et al, “Provisioning Servers in the Application Tier for E-Commerce Systems”, em *Proceedings of IWQoS’04*, Montreal, 2004.

[11] D. Menasce et al, *Capacity Planning and Performance Modeling: from mainframes to client-server systems*, Prentice-Hall, 1994.

[12] T. Kachigan, “A Multi-Dimensional Approach to Capacity Planning”, em *Proc. of CMG Conference 1980*, Boston, 1980.

[13] J. Rolia, V. Vetland, “Correlating Resource Demand Information with ARM Data for Application Services”, em *Proc. of the ACM Workshop on Software and Performance*, 1998.

[14] T. Kelly, "Detecting Performance Anomalies in Global Applications", em *2nd Workshop on Real, Large Distributed Systems*, 2005.

[15] C. Stewart et al, "Exploiting Nonstationarity for Performance Prediction", em *Proc. of EuroSys'2007*, Lisboa, Março de 2007.

[16] Office of Government Commerce, *Best Practices for Service Delivery*, The Stationery Office, 2003.

[17] J. Clark, "Everything you wanted to know about ITIL in less than one thousand words (White Paper)", Connect Sphere Ltd, Outubro de 2007.

[18] R. Seery, "Minimize IT Risk with a Business Focused Capacity Plan", SAS. Disponível em http://www.bettermanagement.com/library/library.aspx?l=5797. Acesso em 27/ Janeiro/2010.

[19] C. Molloy, "ITIL Capacity Management Deep Dive", em *Proceedings of the Computer Measurement Group's 2005 International Conference*, 2005.

[20] R. Kaminski, Y. Ding, "Business Metrics and Capacity Planning", BMC Software, 2004.

[21] J. Hull, *Options, Futures and Other Derivatives – 7th Edition*, Prentice-Hall, 2008.

[22] J. Siegel et al, *International Encyclopedia of Technical Analysis*, Vision Books, 2006.

[23] R. Edwards, W. Bassetti e J. Magee, *Technical Analysis of Stock Trends - 8th Edition,* cap. 36, pp. 477-482, CRC Press, 2001.

[24] R. Hecht-Nielsen, *Neurocomputing*, Addison-Wesley, 1990.

[25] C. Holt, "Forecasting Seasonals and Trends by Exponentially Weighted Moving Averages", em *O.N.R. Research Memorandum 52*, Carnegie Institute of Technology, 1957.

[26] R. Brown, *Statistical Forecasting for Inventory Control*, McGraw-Hill, New York, 1959.

[27] J. McDonald, *Handbook of Biological Statistics, 2nd Ed*, Sparky House Publishing, Baltimore, Maryland, 2009. Versão online disponível em http://udel.edu/~mcdonald/ /statregression.html. Acesso em 11/Junho/2010.

[28] D. Marquardt, "An Algorithm for the Least-Squares Estimation of Nonlinear Parameters", em *SIAM Journal of Applied Mathematics*, pp. 431-441, Junho de 1963.

[29] C. Chatfield, Mohammad Yar, "Holt-Winters forecasting: some practical issues", em *Journal of the Royal Statistical Society. Series D: The Statistician*, vol. 37, pp.129-140, 1988.

[30] NIST/SEMATECH*, e-Handbook of Statistical Methods*, National Institute of Standards and Technology. Disponível em http://www.itl.nist.gov/div898/handbook/. Acesso em 20/Março/2010.

[31] C. Adcock, "Choosing linear approximations to a non-linear model: a case study", em *Journal of the Royal Statistical Society. Series D: The Statistician*, vol. 35, pp. 245-249, 1986.

[32] G. Seber, C. Wild, *Nonlinear regression*, cap. 2, pp. 21-32, Wiley-Interscience, Setembro de 2003.

[33] S. Weisberg, *Applied Linear Regression, 3rd Edition*, cap. 2-3, pp. 19-68, Wiley-Interscience, Fevereiro de 2005.

[34] G. Box, G. Jenkins, *Time series analysis: forecasting and control*, Holden-Day, 1970.

[35] G. Eshel, "The Yule Walker Equations for the AR Coefficients". Disponível em http://www-stat.wharton.upenn.edu/~steele/Courses/956/ResourceDetails/YWSourceFiles /YW-Eshel.pdf. Acesso em 13/Abril/2010.

[36] J. Durbin, "Efficient Estimation of Parameters in Moving Average Models", *Biometrika*, vol. 46, pp. 306-316, 1959.

[37] T. Ozaki, "On the Order Determination of ARIMA Models", em *Applied Statistics*, vol. 26, pp. 290-301, 1977.

[38] J. Dufour, "Estimation of ARMA models by maximum likelihood", McGill University, Fevereiro de 1981.

[39] J. Kleijnen, "White noise assumptions revisited: regression metamodels and experimental designs in practice", em *Proc. of the 38th Conference on Winter Simulation*, pp. 107-117, Monterey, California, 2006.

[40] W. Cleveland, "Robust Locally weighted regression and smoothing scatterplots", em *Journal of the American Statistical Association*, vol. 74 (368), pp. 829-836, 1979.

[41] W. Cleveland, E. Grosse e M. Shyu, "Local regression Models", em *Statistical Models in S*, pp. 309-376, Chapman and Hall, New York, 1992.

[42] C. Loader, *Local Regression and Likelihood*, Springer, Julho de 1999.

[43] C. Chatfield, *The analysis of time series: an introduction, 6th Edition*, p. 76, CRC Press, Londres, 2004.

[44] L. Koyck, *Distributed lags and investment analysis*, North-Holland Pub. Co., Amesterdão, 1954.

[45] D. Salvatore, D. Reagle, *Schaum's Outline of Statistics and Econometrics, Second Edition*, cap. 8, pp.181-205, McGraw-Hill Professional, New York, Outubro de 2001.

[46] C. Chatfield, *Time-series forecasting*, CRC Press, Londre, 2001.

[47] S. Makridakis, S. Wheelwright e R. Hyndman, *Forecasting: methods and applications*, Wiley, Dezembro de 1997.

[48] H. Soper et al, "On the distribution of the correlation coefficient in small samples. Appendix II to the papers of 'Student' and R. A. Fisher. A cooperative study", em *Biometrika*, vol. 11, pp. 328-413, Biometrika Trust, 1917.

[49] C. Brandon, J. Jarrett e S. Khumawala, "A comparative study of the forecasting accuracy of Holt-Winters and economic indicator models of earnings per share for financial decision making", em *Managerial Finance*, vol. 13, pp. 10-15, 1993.

[50] J. Taylor, "A comparison of univariate time series methods for forecasting intraday arrivals at a call center", em *Management Science*, vol. 54, pp. 253-265, Fevereiro de 2008.

[51] J. Armstrong, F. Collopy, "Error measures for generalizing about forecasting methods: empirical comparisons", em *International Journal of Forecasting*, vol. 8, pp. 69-80, 1992.

[52] S. Shapiro, M. Wilk, "An analysis of variance tests for normality (complete samples)", em *Biometrika*, vol. 52, pp. 591-611, 1965.

[53] R. D'Agostino, A. Belanger, e R. D'Agostino Jr., "A suggestion for using powerful and informative tests of normality", em *The American Statistician*, vol. 44, Novembro de 1990.

[54] D. Keefer, S. Bodily, "Three-point approximations for continuous random variables", em *Management Science*, vol. 29, pp. 595-609, Maio de 1983.

[55] T. Reenskaug, "Thing-Model-View-Editor: an example from a planningsystem", Maio de 1979.

[56] The R Development Core Team, *"R: A language and environment for statistical computing"*, The R Foundation for Statistical Computing, Vienna, 2009.

[57] D. Montgomery, L. Contreras, "A note on forecasting with adaptive filtering", em *Operational Research Quarterly*, vol. 28, pp. 87-91, Pergamon Press, Reino Unido, 1977.
[58] R. Engle, "GARCH 101: The use of ARCH/GARCH models in applied econometrics", em *Journal of Economic Perspectives*, vol. 15(4), pp. 157-168, 2001.
[59] C. Leser, "A simple method of trend construction", em *Journal of the Royal Statistical Society, Series B (Methodological)*, vol. 23, pp. 91-107, 1961.
[60] J. Robertson, E. Tallman, "Vector Autoregressions: Forecasting and reality", em *Economic Review*, pp. 4-18, Federal Bank of Atlanta, 1999.
[61] H. Yang, C. M. Huang, C. L. Huang, "Identification of ARMAX model for short term load forecasting: an evolutionary programming approach", em *IEEE Transactions on Power Systems*, vol. 11(1), pp. 403-408, Fevereiro de 1996.