

MESTRADO

MÉTODOS QUANTITATIVOS PARA A DECISÃO ECONÓMICA E EMPRESARIAL

TRABALHO FINAL DE MESTRADO

RELATÓRIO DE ESTÁGIO

INTEGRAÇÃO DE MÉTODOS PREDITIVOS EM SISTEMAS DE INFORMAÇÃO NA QUIDGEST

DANIEL FILIPE MONTEIRO TOMÁS

OUTUBRO – 2018

MESTRADO EM
MÉTODOS QUANTITATIVOS PARA A DECISÃO
ECONÓMICA E EMPRESARIAL

TRABALHO FINAL DE MESTRADO
RELATÓRIO DE ESTÁGIO

INTEGRAÇÃO DE MÉTODOS PREDITIVOS EM SISTEMAS
DE INFORMAÇÃO NA QUIDGEST

DANIEL FILIPE MONTEIRO TOMÁS

ORIENTAÇÃO:

PROF. CARLOS J. COSTA
JOÃO PAULO CARVALHO

OUTUBRO - 2018

Agradecimentos

Primeiramente gostaria de agradecer ao ISEG pela minha formação académica e em especial para a Coordenação do Mestrado, pelo apoio, preocupação e incentivo ao longo de todo o processo de estágio.

Ao professor Carlos J. Costa agradeço os conselhos, sugestões e pela disponibilidade em todo o processo de realização do relatório.

À Quidgest – Consultores de Gestão, SA e em especial ao Dr. João Paulo Carvalho agradeço a oportunidade de estágio e os conselhos e esclarecimentos de dúvidas.

Agradeço a todos os colaboradores com quem tive a oportunidade de trabalhar, pela ajuda, incentivo e boa disposição que se viveu ao longo do estágio.

A todos os meus amigos, agradeço o apoio que recebi ao longo deste processo e também pela paciência e compreensão.

Por fim, à minha família. Aos meus avós que sempre depositaram muita confiança em mim, que conseguiria atingir os meus objetivos, mas em especial aos meus pais que me ofereceram a possibilidade de cumprir os meus objetivos académicos, sem eles nunca teria chegado aqui.

Resumo

A utilização de estudo de séries temporais para previsão é particularmente relevante, juntamente com a necessidade de os sistemas de informação terem ferramentas de *Business Intelligence*. A sua utilização em contexto empresarial permite de alguma forma ajustar a estratégia do negócio de uma forma interativa. O objetivo deste trabalho consiste em propor uma solução que permita que sistemas de informação disponham de componentes de *data science*, mais especificamente para a análise de séries temporais. A abordagem metodológica enquadra-se no que se designa *design science research*. Partindo da revisão da literatura, foi apresentada uma solução conceptual. Esta solução foi implementada e permitiu validar a exequibilidade da proposta. Esta solução permitiu ainda comparar várias abordagens para análise de séries temporais.

Abstract

The use of time series study for forecasting is particularly relevant, along with the need for information systems to have Business Intelligence tools. Its use in business context allows in some way to adjust the strategy of the business in an interactive way. The objective of this work is to propose a solution that allows information systems to have data science components, more specifically for the analysis of time series. The methodological approach followed is the design science research. This resulted in a conceptual proposal that was implemented. The implementation allows to validate the feasibility of the proposal. This solution also allowed to compare several approaches for time series analysis. The results presented show significant differences in function of the analyzed realities. Forecasting errors associated with incidents are lower than orders or billing. The relevant result of this report presented here was the creation of a solution that can be used by managers for decision making and integrable in information systems.

Índice

AGRADECIMENTOS	4
RESUMO	5
ABSTRACT	6
ÍNDICE	7
ÍNDICE DE TABELAS	9
ÍNDICE DE FIGURAS	10
ACRÓNIMOS	11
1. INTRODUÇÃO	12
1.1. Enquadramento	12
1.2. Empresa – Quidgest – Consultores de Gestão, SA	12
1.2.1. Genio.....	13
1.2.2. Balanced Scorecard	13
1.2.3. Quigenio	13
1.3. Objectivo.....	14
1.4. Abordagem Metodológica.....	14
1.5. Contribuições	15
1.6. Estrutura do TFM.....	15

2.	REVISÃO DA LITERATURA	16
2.1.	<i>Business Intelligence e Business Analytics</i>	16
2.2.	Integração de sistemas	18
2.3.	Séries temporais.....	19
3.	PROPOSTA DE SOLUÇÃO CONCEPTUAL	24
4.	TRABALHO EMPÍRICO	26
4.1.	Arquitetura da solução.....	26
4.2.	Ferramentas informáticas (software) utilizadas	27
4.3.	Bibliotecas e outras tecnologias	28
4.4.	Extração de dados.....	31
5.	APRESENTAÇÃO DE DADOS E DISCUSSÃO DE RESULTADOS	34
6.	CONCLUSÕES E TRABALHOS FUTUROS	45
	REFERÊNCIAS BIBLIOGRÁFICAS	48
	ANEXOS	52
	INSTALAÇÃO E CONFIGURAÇÃO DO R SERVICES NO SQL SERVER	60

Índice de Tabelas

Tabela 1 - Descrição das séries temporais	31
Tabela 2 - Síntese das previsões da faturação	37
Tabela 3 - Comparação dos modelos - faturação	39
Tabela 4 - Síntese das previsões das encomendas	41
Tabela 5 - Comparação dos modelos - encomendas	41
Tabela 6 - Síntese das previsões para os incidentes	42
Tabela 7 - Comparação dos modelos – incidentes	43

Índice de Figuras

Figura 1 – Esquema da integração da linguagem programação para <i>data science</i> com um SGBD e os seus resultados	25
Figura 2 - Esquema da integração do R com o SQL Server	26
Figura 3 – <i>Stored procedure</i> com código R	29
Figura 4 - Ambiente de desenvolvimento do Visual Studio para relatórios	30
Figura 5 - Faturação mensal	38
Figura 6 - Decomposição da série Faturação	38
Figura 7 – Modelo de previsão utilizando regressão linear	39
Figura 8 - Erros de previsão	40
Figura 9 - <i>Layout</i> dos relatórios	43
Figura 10 - Relatório integrado num sistema criado na Quidgest	44
Figura 11 - <i>Stored Procedure</i> do gráfico da série temporal – Faturação	52
Figura 12 - <i>Stored procedure</i> do gráfico decomposição da série temporal – Faturação	53
Figura 13 - <i>Stored procedure</i> do gráfico com previsão da série temporal – Faturação (ARIMA)	54
Figura 14 - <i>Stored procedure</i> do gráfico com previsão da série temporal – Faturação (Holt-Winters)	55
Figura 15 - <i>Stored procedure</i> do gráfico com previsão da série temporal – Faturação (Alisamento Exponencial)	56
Figura 16 - <i>Stored procedure</i> do gráfico com previsão da série temporal – Faturação (Regressão Linear)	57
Figura 17 - <i>Stored procedure</i> da conclusão de qual o melhor modelo da série temporal – Faturação	58
Figura 18 - <i>Stored procedure</i> do sumário dos dados da série temporal – Faturação	58

Figura 19 - <i>Stored procedure</i> das previsões obtidas com todos os modelos para a série temporal – Faturação	59
Figura 20 - Página inicial do <i>SQL Server Installation Center</i>	60
Figura 21 - Seleção do tipo de instalação do SQL Server	61
Figura 22 - Seleção das funcionalidades do SQL Server	61
Figura 23 - Resultado inicial do comando <i>sp_configure 'external scripts enabled'</i>	62
Figura 24 - Resultado intermédio do comando <i>sp_configure 'external scripts enabled'</i>	62
Figura 25 - Esquema do funcionamento interno do <i>R Services</i>	63
Figura 26 – <i>Stored procedure</i> de validação da integração	63

Acrónimos

ARIMA – *Auto Regressive Integrated Moving Average*

BA – *Business Analytics*

BI – *Business Intelligence*

ERP – *Enterprise resource planning*

SGBD – Sistema gerenciador de bases de dados

SQL – *Structured Query Language*

1. Introdução

1.1. Enquadramento

A conjuntura empresarial obriga as empresas a ser cada vez mais competitivas. Os mercados são cada vez mais exigentes, sendo imprescindível antecipar tendências. Mas mais que tendências é indispensável prever cada vez com maior precisão. Tem que se prever procura, encomendas, vendas, custos de matérias-primas ou de equipamentos. A utilização de métodos de previsão é cada vez mais utilizada pelas empresas. Com efeito, a sua utilização permite às empresas definirem estratégias para aumentarem as suas vendas e identificar potenciais clientes. Neste contexto, procura-se perceber quais as ferramentas mais indicadas para previsão de faturação, encomendas e incidentes.

1.2. Empresa – Quidgest – Consultores de Gestão, SA

A Quidgest foi criada em 1988. É uma empresa multinacional de origem portuguesa, de consultoria e desenvolvimento de sistemas de informação de gestão que aposta na investigação em engenharia de software. A Quidgest está presente para além de Portugal, em Timor-Leste, Moçambique, Macau, Marrocos e Alemanha, investindo com grande sucesso na sua internacionalização. A empresa tem atualmente mais de 120 colaboradores. As suas áreas de ação são: gestão da Quidgest, investigação e desenvolvimento, marketing e comunicação, gestão de recursos humanos, gestão financeira, gestão patrimonial, gestão bancária, gestão documental e processos de negócio, gestão de sistemas de saúde, projetos especiais e consultoria de negócio.

A Quidgest tem três grandes plataformas de trabalho e funcionamento da empresa de modo a controlar as várias áreas de gestão. As plataformas de maior dimensão e são: Genio, *Balanced Scorecard* e Quigenio.

1.2.1. Genio

O Genio é um software de geração automática de código desenvolvido pela Quidgest desde 1991. Tem como objetivo aumentar a produtividade no desenvolvimento de sistemas de informação, reduzindo custos e tempo na geração de código. A Quidgest produz mais de 150 milhões de linhas de código por mês com o Genio. Foi com este software que a empresa desenvolveu mais de 200 sistemas diferentes. Permite a qualquer utilizador sem conhecimentos de programação criar um sistema de informação em poucos minutos.

1.2.2. Balanced Scorecard

O *Balanced Scorecard* Quidgest é um sistema de informação de apoio à Gestão Estratégica que tem como principal objetivo garantir a eficácia do planeamento, da comunicação, da execução, da monitorização e do controlo da estratégia organizacional. Ferramenta ideal para a operacionalização da Gestão Estratégica: eficaz na implementação da estratégia organizacional e eficiente, ao focar-se naquilo que é fundamental para a organização. Permite mobilizar os recursos mais adequados e evitar, deste modo, os desperdícios, reduzindo ainda os custos de circulação de informação.

1.2.3. Quigenio

A plataforma Quigenio é uma ERP, que armazena as informações sobre a empresa, quer de clientes quer dos colaboradores, no seu *template* tem alguns grafismos com dados relevantes sobre incidentes, faturação, possíveis negócios e etc. Também é o software que os colaboradores usam para registar ações e incidentes.

1.3. Objetivo

O objetivo deste trabalho consiste em propor uma solução que permita que sistemas de informação disponham de componentes de *data science*, mais especificamente para a análise de séries temporais.

A solução consiste na integração de uma linguagem de programação para *data science* com um Sistema gerenciador de bases de dados (SGBD) para desenvolvimento de relatórios sobre séries temporais, os quais são apresentados nos sistemas de informação como auxiliares à tomada de decisão.

1.4. Abordagem Metodológica

A abordagem metodológica de sistemas de informação pode seguir duas vertentes: *natural science* ou *design science*. A primeira tem sido a abordagem dominante desde a sua existência, é uma vertente de investigação centrada em métodos tradicionais, baseada em domínios físicos e biológicos, tendo duas fases de desenvolvimento – descoberta e justificação. Contudo esta abordagem não é a mais apropriada para a temática deste trabalho. Com efeito, *design science* começou a ganhar popularidade no início da década de noventa (March & Smith, 1995, Nunamaker, Chen & Purdin, 1991, Walls, Widmeyer & Sawy, 1992), centra-se em domínios da gestão e da engenharia (nomeadamente informática) para resolver problemas. Diferencia-se de *natural science*, porque ao contrário de tentar compreender a realidade o *design science* tenta criar algo para a utilidade humana (Simon, 1969).

Segundo Ken Peffers (2008), o processo de *Design science* tem seis passos: motivação e identificação do problema, definição dos objetivos para a solução, desenvolvimento e *design*, demonstração, avaliação e comunicação (Peffers, Tuunanen, Rothenberger, & Chatterjee, 2007). De referir que nem todos os passos têm que ser obrigatoriamente respondidos.

Neste trabalho, a motivação e identificação do problema foi - como integrar *data science* em sistemas de informação.

A definição dos objetivos para a solução passou por escolher uma linguagem programação para *data science* (R, Python, Matlab e etc) que fosse integrável num SGBD (SQL Server, Oracle, My SQL e etc) para criar ferramentas em sistemas de informação. O desenvolvimento e *design* consistiu em criar *stored procedures* armazenados no SGBD, desenvolvidos com a linguagem de programação para *data science*, em seguida desenvolver relatórios para serem apresentados em sistemas de informação. Demonstração, avaliação e comunicação passaram por avaliar a praticabilidade dos relatórios e os seus resultados, isto é, verificar se os modelos criados faziam sentido e por último mostrar os resultados à organização.

1.5. Contribuições

Para além da contribuição prática, que se traduziu na resolução de um problema concreto. Este TFM (Trabalho de fim de curso), teve como resultado a realização de trabalho que foi apresentado na conferência portuguesa de sistemas de informação realizada em outubro de 2018 em Santarém, e publicado nas respetivas atas (Tomás, Costa, Gaivão & Carvalho, 2018).

1.6 Estrutura do TFM

Este relatório, depois de uma breve introdução, apresenta a revisão da literatura. Em seguida é descrito o método utilizado. Depois, os resultados são apresentados, sendo, por fim, realizada uma breve discussão e conclusão.

2. Revisão da Literatura

2.1. *Business Intelligence e Business Analytics*

O uso de aplicações/ferramentas de *Business Intelligence* (BI) tem-se tornado cada vez mais uma realidade e uma “obrigação” para as empresas e em especial para as empresas que desenvolvem sistemas de informação, para se manterem competitivas. O interesse pelo BI surgiu durante a década de 90, apesar de já existir desde a década de 60, mas com outras designações, nomeadamente sistemas de suporte à decisão (Power, D. J. 2010). De acordo com Boris Evelson da empresa Forrester Research, BI é um conjunto de metodologias, processos, arquiteturas e tecnologias, que transformam dados em bruto em informação relevante. Permite aos utilizadores tomarem decisões que os colocam à frente dos seus competidores (Evelson, Boris 2008).

Business Analytics (BA), é o processo de transformação de dados em ações através de análise de dados para a tomada de decisão e resolução de problemas no contexto empresarial (Matthew J. Liberatore & Wenhong Luo 2010), cobre as mais variadas áreas tais como: finanças, vendas, marketing, recursos humanos, redes sociais e muitas outras. As informações obtidas com o BA, tais como: tendências, sazonalidade, relações e outros padrões permitem às empresas otimizar e automatizar os seus processos produtivos.

O BI é normalmente descrito como relatórios, alertas, *dashboards* e *scorecards* já o BA é descrito como análises estatísticas, análises quantitativas, *data mining* e métodos preditivos (Evans & Lindner, 2012).

Segundo Muhmmad I. Nofal, nos dias de hoje com uma economia tão competitiva e num contexto de BI e *Enterprise resource planning* (ERP) complexos, tornaram-se numa ferramenta estratégica importante, com impacto direto no sucesso de qualquer projeto. Os sistemas de informação integrados

com ferramentas BI têm transformado as organizações para melhor, permitindo aos gestores compreenderem melhor a informação que têm ao seu dispor (Muhmmad I. Nofal & Zawiyah M. Yusof 2013). De maneira geral, a integração de sistemas de informação com soluções de BI, realiza-se com os dados do sistema armazenados num SGBD, que depois são tratados e armazenados internamente na base de dados ou então externamente através de softwares ou linguagens de programação através de scripts automáticos, para serem expostos como relatórios, *dashboards* ou *scoreboards*.

As ferramentas de BI podem retirar informação de ERP e de outros sistemas de informação, sendo possível disponibilizar relatórios nos ERP para ajudar os utilizadores a melhorar as tomadas de decisão. Contudo é comum o BI desenvolver relatórios com dados transacionáveis de ERP. O BI tem sido criticado pelo nível dos relatórios ser estratégico e não operacional o que cria uma perda de desempenho. Para resolver este problema é possível integrar as capacidades do BI no ERP através dos seguintes níveis de integração: dados, processos de negócio, aplicação e integração de utilizadores (Russman. Seymour & Belle, 2017).

Para um total aproveitamento do potencial do BI, é necessário integrá-lo num sistema central que suporte a gestão de organizações, como é o caso dos ERP em que as organizações geralmente os utilizam como peça chave da infraestrutura de TI. O principal argumento para a integração de ferramentas de BI com ERP tem foco na obtenção de informação em tempo real para os utilizadores. Outro argumento para esta integração é a consistência e comunicações mais amplas com os clientes (Polkowski, Constantin, Raducu, & Blidaru, 2016).

2.2. Integração de sistemas

Para Wilhemlm Hasselbring as unidades organizacionais possuem 3 camadas: a arquitetura do negócio, a arquitetura da aplicação e a arquitetura tecnológica. A arquitetura do negócio define a estrutura organizacional e os *workflows* das regras e processos de negócio. Trata-se de um nível conceptual expressado em termos significativos para os utilizadores de sistemas de informação. A arquitetura da aplicação define a implementação dos conceitos do negócio em termos de aplicação em ambiente empresarial. A principal função desta camada é a de ligação entre a arquitetura de negócio e a arquitetura tecnológica. A investigação em sistemas de informação aponta para preencher a lacuna existente entre o negócio e a tecnologia, o que requer cooperação interdisciplinar entre o domínio da aplicação e a informação da tecnologia. A arquitetura tecnológica define a informação e comunicação de infraestruturas. Com estes níveis é possível definir para cada tipo de organização qual o melhor método de organização tais como - integração vertical ou integração horizontal (Hasselbring, 2000, Caetano, & Costa, 2014).

As organizações dependem do uso de sistemas de informação para serem eficientes e manterem-se competitivas. Por outro lado, os trabalhadores necessitam deles para completarem as suas tarefas diariamente. As organizações têm usado os sistemas de informação para aumentarem a sua produtividade e os lucros, diminuindo os custos. Com esta constante procura de melhorar resultados as organizações necessitam que os sistemas comuniquem entre si e por isso é necessário que estejam integrados (por exemplo, Costa, Antunes & Dias, 2002). O principal objetivo da integração de sistemas passa por assegurar que os sistemas são capazes de comunicar entre si de forma correta e que não exista redundância. A metodologia adotada para resolver este problema assenta em três métodos: método qualitativo, investigação de casos de estudo e abordagem de recolha de dados (Sekgweleo, Billawer, & Hamunyela, 2016).

Nos dias de hoje, com o desenvolvimento tecnológico, é perceptível a rápida difusão de aparelhos eletrônicos tais como os *smarthpones*, *tablets* e etc. Tradicionalmente, as organizações implementam sistemas de informação para resolver problemas internos. Como resultado, a informação fica dispersa pela organização sem que exista comunicação. A forma mais comum de integração de sistemas é a integração pelos dados. Esta integração ocorre devido a muitas das bases de dados das organizações conterem dados similares entre elas, sem que exista comunicação (Mohamed, Mahadi, Miskon, Haghshenas & Adnan 2013). Um caso paradigmático é o do software *Web/z*, que integra dados para sistemas de informação de imunização numa escola de enfermagem. Dizem ter utilizado com sucesso em 3144 clientes incluindo escolas, clínicas e hospitais (Galemore 2011).

2.3. Séries temporais

Os estudos sobre séries temporais passam pela identificação das principais características de grupos de observações feitas sequencialmente ao longo do tempo relacionadas com determinadas realidades. No trabalho aqui apresentado, foram utilizados os seguintes modelos para estudo das séries temporais: Modelos ARIMA (*Auto-Regressive Integrated Moving Average*), Alisamentos Exponenciais, *Holt-Winters* e Regressões Lineares.

Os métodos modernos para previsão usando séries temporais têm como ideia fundamental que o passado permite de alguma forma antever o futuro. Portanto, a questão de fundo da previsão está relacionada com a forma como se interpreta a informação codificada em eventos passados e o método de extrapolação dessa informação para o futuro.

Os desenvolvimentos teóricos na análise de séries temporais começaram com o advento dos processos estocásticos (Doob, 1934). A abordagem clássica à previsão usando séries temporais é baseada na regressão (Yule, 1927 e Walker,

1931). O modelo standard de regressão envolve a especificação de uma relação paramétrica entre um conjunto de variáveis explanatórias (ou exógenas) e um conjunto de variáveis dependentes (ou endógenas). A estimação dos parâmetros do modelo pode ser realizada de diversas maneiras, com origem no trabalho pioneiro de Gauss em 1809 com o método dos mínimos quadrados - *Theoria Motus Corporum Coelestium in Sectionibus Conicis Solem Ambientum*.

A primeira aplicação real de modelos autorregressivos para previsão remonta ao trabalho de G. U Yule (Yule, 1927) e J. Walker (Walker, 1931) nas décadas de 1920 e 1930. Durante esse período, a média móvel foi introduzida para remover flutuações periódicas nas séries temporais, por exemplo, flutuações devido à sazonalidade. Herman Wold (1954) introduziu os modelos ARMA (*Auto-Regressive Moving Average*) para séries estacionárias, mas não conseguiu derivar uma função de verossimilhança para permitir a estimação da máxima verossimilhança (ML) dos parâmetros. Foi só em 1970, que Box e Jenkins (1970) apresentaram o procedimento completo de modelação para séries individuais: especificação, estimativa, diagnóstico e previsão. Atualmente, os chamados modelos de Box-Jenkins são talvez os mais usados e muitas técnicas usadas para previsão e ajuste sazonal podem ser rastreadas até esses modelos. Ao introduzir diferenciação nas variáveis dependentes, resolvendo assim o problema da estacionariedade da série, obtemos um modelo ARIMA sem sazonalidade (Box & Jenkins, 1970). ARIMA é o acrónimo de *Auto Regressive Integrated Moving Average* e o modelo pode ser escrito da seguinte forma:

$$y'_t = c + \phi_1 y'_{t-1} + \dots + \phi_p y'_{t-p} + \theta_1 \varepsilon_{t-1} + \dots + \theta_q \varepsilon_{t-q} + \varepsilon_t,$$

Chamamos a este modelo ARIMA(p,d,q), com o p a corresponder à ordem da parte autorregressiva, d à ordem de diferenciação e q à ordem da parte médias móveis.

As técnicas desenvolvidas por Box & Jenkins nos anos 70 tiveram um grande sucesso e impacto, em parte devido a uma metodologia simples e de fácil aplicação para a previsão em diversos contextos académicos ou empresariais. Contudo, outros autores preferiam uma abordagem mais estrutural à previsão. Por exemplo, (Harrison & Stevens, 1976) tiveram algum sucesso na formulação de um modelo linear Markoviano num contexto da estatística Bayesiana. Usando o filtro de Kalman, Harrison & Stevens especificaram um modelo linear dinâmico com parâmetros que variam no tempo para resolver o problema da não-estacionariedade.

Uma generalização dos modelos autorregressivos é o chamado ARCH, acrónimo de *autoregressive conditional heteroscedasticity*. Estes modelos foram introduzidos por (Engle, 1982) com o objetivo de capturar a não-estacionariedade do momento de segunda ordem, ou seja, a variância condicional ou volatilidade. Posteriormente, o modelo GARCH (*generalized autoregressive conditional heteroscedasticity*), desenvolvido por (Bollerslev, 1986), representa a variância dos erros como função dos termos autorregressivos.

Outra generalização dos modelos autorregressivos foi aceitar modelos multivariados, entre os quais os modelos VAR (*Vector Autoregressive*) que se tornaram populares. VAR são modelos lineares multivariáveis capazes de capturar a dinâmica conjunta de múltiplas séries temporais. No seu trabalho pioneiro, Sims (1980) propôs a substituição de modelos macroeconómicos de larga escala nos anos 60 por VAR. Estas técnicas são aplicáveis somente para séries temporais estacionárias. No entanto, as séries temporais, nomeadamente as séries de interesse para a Economia, exibem uma tendência crescente sugerindo não-estacionariedade, isto é, a presença de uma raiz unitária.

Os modelos de Alisamento Exponencial surgiram nos finais dos anos 50 e permitiram criar melhores modelos de previsão – Brown (1956). Holt, (1957).

Winters (1960). Este modelo produz pesos médios ponderados das observações passadas, com os pesos a decaírem exponencialmente à medida que a série se torna mais antiga, ou seja, quanto mais recentes são as observações maior será o seu peso para influenciar a previsão. Este tipo de modelo permite prever uma grande variedade de séries temporais e rapidamente, o que se torna uma vantagem para o mercado empresarial nos dias de hoje. No caso multivariado, verificou-se que as séries temporais não estacionárias podem ter uma raiz unitária comum. Estas séries temporais são chamadas de séries temporais co-integradas e podem ser usadas nos chamados modelos de correção de erros dentro de relacionamentos de longo prazo e dinâmicas de curto prazo são estimadas. Os modelos de Regressão linear para séries temporais são modelos de Regressão linear comuns, mas as variáveis dependentes são a sazonalidade e a tendência da série.

Outra classe de modelos, os TAR (*threshold nonlinear ARMA*), propostos por (Tong, 1990), foram aplicados com sucesso à previsão de variáveis relevantes em Economia. Decompondo séries temporais usando wavelets, (Lineesh & John, 2010) usaram os modelos ARMA e TAR para fazer previsão de cada componente da decomposição. Em Krishnamurthy e Yin (2002), combinaram-se cadeias de Markov ocultas com modelos AR onde os parâmetros do modelo AR variavam de acordo com a realização da cadeia de Markov.

Nas últimas décadas, a previsão usando redes neuronais tem obtido bastante popularidade, devido ao seu desempenho e capacidade de classificação (Zhang, 2012). As redes neuronais têm diversas características únicas que as tornam apelativas: 1) são *data-driven*; 2) não requerem um modelo explícito subjacente; 3) flexíveis, universais e capazes de lidar com modelos complexos.

No que diz respeito a aplicações práticas, tem havido aplicações desde os modelos financeiros (Khairalla, & Ning, 2017), a telecomunicações (Puente,

Hernandez & Salcedo, 2009).). Porém, com frequência, utilizam-se abordagens híbridas (Pati & Shukla, 2015) ou com recurso a redes neuronais (Ji, Yu, Guo, & Zhang, 2016).

3. Proposta de Solução Conceptual

Através da revisão da literatura e tendo em conta o processo de negócio de uma empresa que desenvolve sistemas de informação, foi possível chegar à conclusão que a solução passa pela integração de linguagens de programação para *data science* em sistemas de gestão de bases de dados, de forma a criar soluções analíticas avançadas em sistemas de informação. Consiste num processo semelhante ao da integração pelos dados, em que as comunicações entre a base dados e a linguagem de programação ocorrem sob a base de dados. A integração começa com o desenvolvimento de *queries* T-SQL, para depois serem tratadas e transformadas pelas linguagens programação para *data science*. Através desta integração é possível desenvolver *stored procedures* quer para o armazenamento de modelos preditivos, quer para armazenamento de dados. Contudo, é boa prática testar inicialmente no ambiente da linguagem de programação para *data science*, para aferir se os gráficos e modelos estão a funcionar, isto porque os SGBDs não têm possibilidade de mostrar os resultados na sua totalidade, podendo escapar algum erro.

O passo seguinte passa pelo desenvolvimento da solução no sistema de informação. Aqui existem pelo menos duas possibilidades: criar aplicações web ou criar relatórios. Os relatórios acabam por ser a solução mais simples isto porque a maioria dos SGBDs têm serviços de *Reporting* já associados e por isso mais simples de incorporar nos sistemas. A incorporação foi realizada através de um *iframe* na página HTML destinada aos relatórios, também era possível integrar os relatórios diretamente na aplicação, no entanto algumas das funcionalidades seriam perdidas. Os relatórios têm um campo para seleccionar o número de meses que o utilizador pretende prever, menus com os possíveis modelos de previsão disponíveis, contêm uma breve explicação do modelo, gráfico com a previsão, sumário dos dados e erros de previsão. Estas funcionalidades são desenvolvidas através de funções de uma linguagem de

programação para *data science* e as previsões são geradas de forma automática, selecionando a melhor possibilidade para cada modelo. Foram desenvolvidos relatórios para fazer previsões sobre faturação, número de incidentes e encomendas.

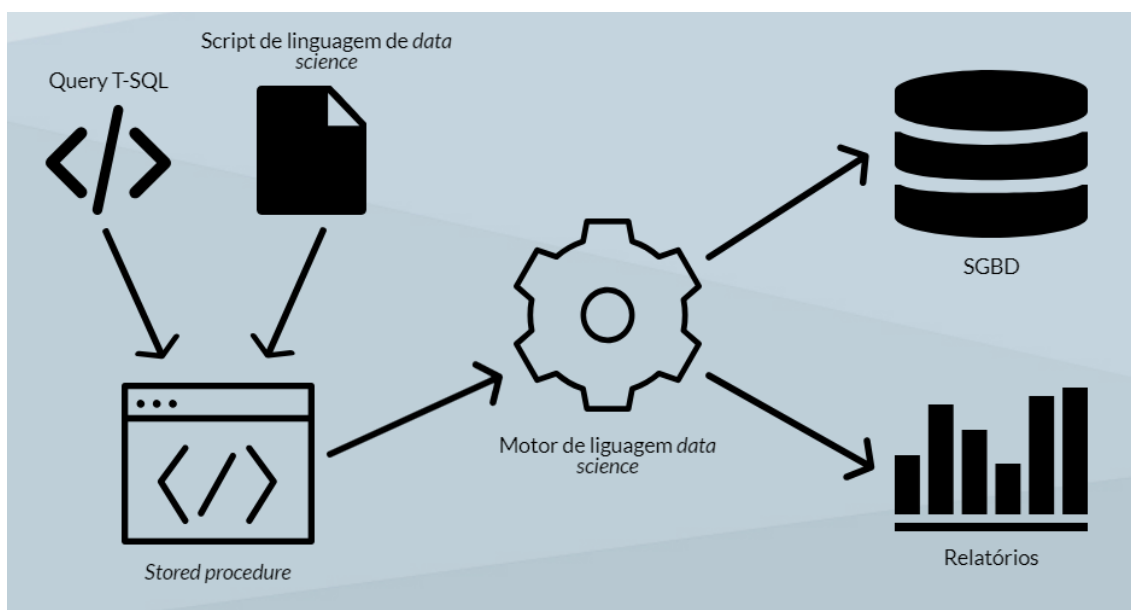


Figura 1 – Esquema da integração da linguagem programação para *data science* com um SGBD e os seus resultados

4. Trabalho Empírico

4.1 Arquitetura da solução

Os dados fornecidos pela empresa (dados do Quigenio) em SQL Server, foram extraídos inicialmente para o R, através do package RODBC, permite executar *queries* SQL no ambiente R. Numa fase posterior esses dados passaram a ser tratados no SQL Server com a integração do R (*R Services*). Após a integração, procedeu-se à criação de *stored procedures* que podem conter gráficos, tabelas, sumários para depois serem usados na criação de relatórios.

Foi realizado um estudo sobre métodos de previsão com R, especialmente em séries temporais. Dos modelos apresentados descreveu-se a razão da escolha dos melhores modelos, que se mostraram mais fiáveis e eficientes, passando ao desenvolvimento de relatórios. Os relatórios foram desenvolvidos em Visual Studio, dada a facilidade e rapidez com que se relaciona com o SQL Server (são ambos softwares da Microsoft) e devido à familiaridade que os colaboradores da Quidgest têm com este software.

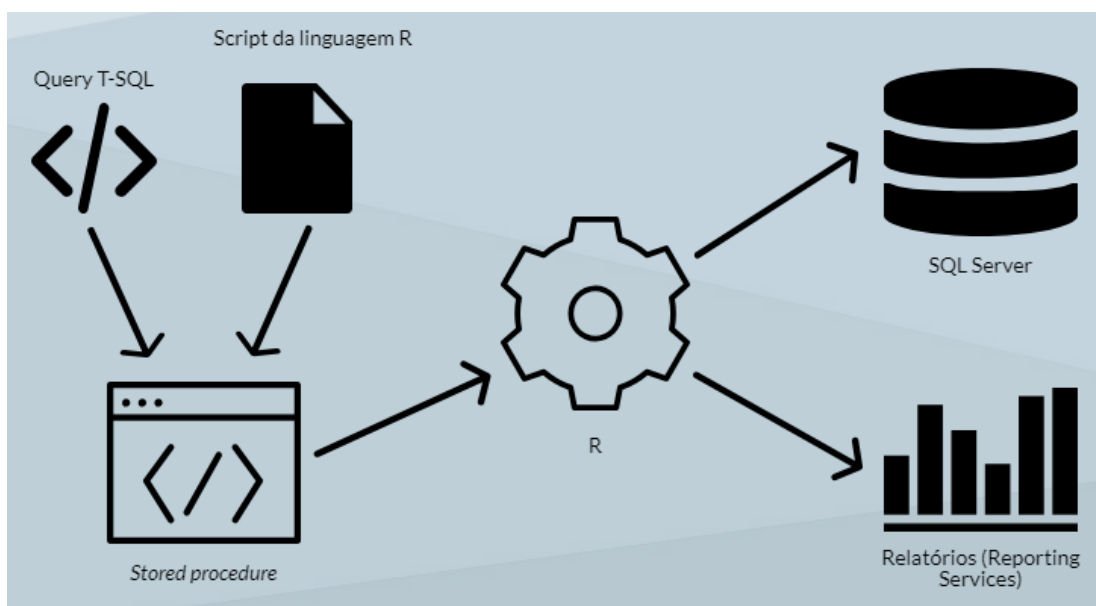


Figura 2 - Esquema da integração do R com o SQL Server

4.2. Ferramentas informáticas (software) utilizadas

De entre as principais ferramentas informáticas utilizadas são de destacar o R, o Microsoft SQL Server, o Microsoft Visual Studio e o Genio. R é uma linguagem e também um ambiente de desenvolvimento integrado para cálculos estatísticos e gráficos. O R é também altamente expansível com o uso de *packages*, que são bibliotecas para sub-rotinas específicas ou áreas de estudo específicas. A linguagem R é largamente usada entre estatísticos e analistas de dados para desenvolver software de estatística e análise de dados. Pesquisas e levantamentos com profissionais da área mostram que a popularidade do R aumentou substancialmente nos últimos anos.

O Microsoft SQL Server é um sistema gerenciador de base de dados relacional (SGBD) desenvolvido pela Microsoft. Uma base de dados tem como principal função armazenar e recuperar dados solicitados por outras aplicações de software, seja no mesmo computador ou em execução em outro computador através de uma rede (incluindo a Internet).

Microsoft Visual Studio é um ambiente de desenvolvimento integrado (IDE) da Microsoft para desenvolvimento de software especialmente dedicado à .NET Framework e às linguagens Visual Basic, C, C++, C# e J#. As linguagens com maior frequência nessa plataforma são: VB.NET (Visual Basic.Net) e o C#.

Para este trabalho o software serviu apenas para desenvolver relatórios, através do *Reporting Services* do SQL Server.

4.3. Bibliotecas e outras tecnologias

No contexto referido na secção anterior é ainda de destacar os *packages* do R, *stored procedures* e relatórios.

Para o caso de estudo, os *packages* do R frequentemente utilizados são: dplyr, forecast, ggplot2 e scales. O dplyr é um *package* orientado para manipulação e tratamento de dados de forma simples. O forecast é um *package* usado para o estudo de séries temporais, tem um conjunto de funções para modelos preditivos e para o cálculo de erros de previsão. O ggplot2 é o *package* mais utilizado pelos cientistas de dados para representação gráfica. A função autoplot() do ggplot2 foi criada de forma apropriada para objetos ts (séries temporais), enquanto que o *package* scales tem como objetivo melhorar a representação dos eixos dos gráficos, muitas das vezes a escala não é a mais apropriada.

O *R Services* permite aplicar no SQL Server a grande maioria das funcionalidades e *packages* do R, o que possibilita a exploração e tratamento estatístico/gráfico em apenas um ambiente de desenvolvimento. Para que o código R seja suportado no SQL Server é necessário instalar e posteriormente ativar o *R Services*. Após a instalação do *R Services*, ficamos com uma nova funcionalidade no SQL Server o *sp_execute_external_script*, que permite usar código R ou Python num *stored procedure*.

Na sua estrutura básica começa-se por definir o nome do procedimento – CREATE PROCEDURE nomeDoProcedimento AS BEGIN; em seguida colocamos o *query* que queremos armazenar e finalizamos com END GO. Depois basta correr o procedimento e fica armazenado na base de dados.

Para o executar basta escrever EXEC nomeDoProcedimento. Neste caso, em que queremos usar a linguagem R, existem algumas mudanças na estrutura.

Em vez de usarmos *queries* usamos EXEC *sp_execute_external_script* e em seguida usamos os seguintes parâmetros: @language, @script, @input_data_1 e @parallel. Acabando com WITH RESULT SETS ((campo1 int,...)).

```
CREATE PROC nomeDoProcedimento
AS
BEGIN
EXEC sp_execute_external_script
@language = N'R',
@script = N' library()
            df = InputDataSet;
            ...
            OutputDataSet = data.frame();',
@input_data_1 = 'SELECT campo1,campo2,... FROM servidor.dbo.tabela',
@parallel = 1
WITH RESULT SETS ((campo1 int, campo2 varchar(50), ...))
```

Figura 3 – *Stored procedure* com código R

O parâmetro @language indica a linguagem que o *external script* (*sp_execute_external_script*) se encontra, podendo assumir os valores R e P (R ou Phyton).

O parâmetro @script é o local em que se coloca o código R, geralmente inclui a ativação de *packages*, tratamento de dados e por fim armazenamento de resultados.

O parâmetro @input_data_1 especifica a *query* SQL que vai ser usada no parâmetro @script. Por fim o parâmetro @parallel tem o objetivo de diminuir a duração do processamento do *stored procedure*.

Os parâmetros têm associadas componentes que são importantes referir. Ao definirmos uma *query* para o @input_data_1 estamos a criar um objeto no parâmetro @script denominado InputDataSet, que contem os dados da *query*

em formato *data frame* (formato tabular), ou seja, tem o papel de elo de ligação entre a linguagem SQL com a linguagem R. Ao armazenar os resultados pretendidos num objeto no parâmetro @script, devemos armazenar os resultados num objeto denominado OutputDataSet, nome que o *external script* reconhece como resultado final, para ser apresentado no SQL Server. Não esquecendo que o objeto deve ser do tipo *data frame*, isto porque o SQL Server não é capaz de interpretar outro tipo de formato.

Após a criação dos *stored procedures*, passamos ao desenvolvimento de relatórios no Visual Studio, para isso teremos que instalar o *Reporting Services* do SQL Server e instalar a extensão do Visual Studio - *Microsoft Reporting Services Projects*, que permite conectar o SQL Server com o Visual Studio na criação de relatórios. Para criar um relatório basta criar um novo projeto *Business Intelligence – Reporting Services*, em seguida escolher a fonte dos dados, inserir os *stored procedures* em formato tabela ou imagem e por fim fazer o design do relatório. Após criar o relatório deve ser publicado no servidor local, que é previamente criado ao instalar o *Reporting Services* do SQL Server. Desta forma é possível implementar estes relatórios na plataforma Genio da Quidgest, para depois serem usados no desenvolvimento de sistemas de informação.

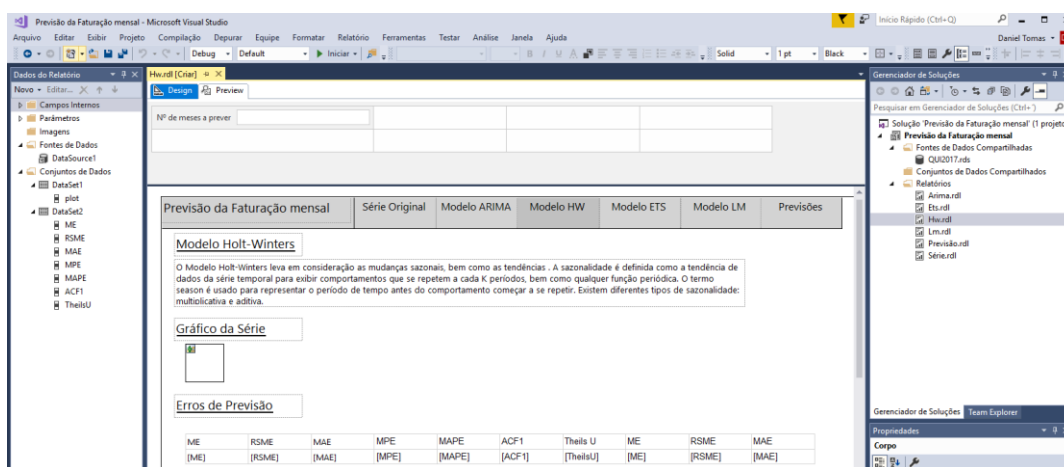


Figura 4 - Ambiente de desenvolvimento do Visual Studio para relatórios

Caso de estudo – Implementação de previsões de séries temporais em sistemas desenvolvidos na Quidgest

4.4. Extração de dados

A extração dos dados foi feita através do SQL Server, que é o sistema gerenciador de bases de dados relacional que a empresa usa. O tratamento dos dados passou por remover os valores nulos e linhas com dados incompletos através do *R Services*. Em seguida, somar mensalmente a quantidade ou valor desde janeiro de 2010 até dezembro de 2017.

Foram criadas três séries – N^o de incidentes mensais, faturação mensal (em €), valor mensal de encomendas (em €), transformando os dados de formato *data frame* para *ts* (série temporal), objeto esse que fica com as datas e valores agregados, simplificando a tarefa do R.

As três séries consideradas foram: faturação, encomendas e incidentes. Na faturação, cada linha corresponde ao valor total faturado pela empresa aos seus clientes. Valor total das encomendas entradas no sistema a pedido dos clientes. Um incidente corresponde a qualquer erro/não conformidade encontrada nos sistemas. Número de observações: 96 meses (8 anos).

Nome da série	Descrição	Valores
Faturação	Faturação mensal em €	12 332 € a 1 640 869 €
Encomendas	Encomendas mensais em €	8 992 € a 4 969 474 €
Incidentes	N ^o de incidentes mensais	20 a 514

Tabela 1 - Descrição das séries temporais

Os dados fornecidos pela empresa encontravam-se armazenados no SQL Server, foram extraídos inicialmente para o R, através do package RODBC, que permite executar *queries* SQL no ambiente R e numa fase posterior esses dados passaram a ser tratados no SQL Server com a integração do R. Após a configuração do *R Services*, procedeu-se à criação de *stored procedures* que podem conter gráficos, tabelas, sumários e outros elementos, depois usados para a criação de relatórios.

Foi realizado um estudo sobre métodos de previsão com R, especialmente em séries temporais, referindo alguns dos seus métodos e modelos preditivos (Hyndman, R.J., & Athanasopoulos, G., 2018). Os relatórios foram desenvolvidos no Visual Studio, têm um campo para selecionar o número de meses que o utilizador pretende prever, menus com os modelos de previsão disponíveis, uma breve explicação do modelo, gráfico com a previsão, sumário dos dados e erros de previsão, tudo isto desenvolvido com funções do R que geram estas previsões de forma automática, selecionando a melhor possibilidade para cada modelo. Foram desenvolvidos relatórios para fazer previsões para faturação, número de incidentes e encomendas.

Para realizar previsões foram escolhidos os modelos – ARIMA, Alisamentos Exponenciais (Holt Winters incluído) e Regressão Linear com variáveis explicativas sazonalidade e tendência. Estes modelos foram os modelos escolhidos por terem sido os melhores a prever algumas das séries de teste. Ainda foi testado um modelo de Redes neuronais, mas tendo em conta o tempo necessário para modelar corretamente as séries em comparação com os outros modelos foi excluído.

Como termo de comparação, para calcular o melhor modelo e realizar a previsão são necessários os seguintes tempos para os modelos: ARIMA – em média 20 segundos; Holt-Winters – em média 15 segundos; Alisamento exponencial – em média 18 segundos; Regressão Linear – em média 12 segundos; Redes neuronais – em média 2-3 minutos.

Especificações do computador utilizado:

Sistema operativo – Windows 10 Enterprise 64 bits

Processador – Intel(R) Core(TM)2 Quad CPU Q9400 2.66GHz

Memória RAM – 4.00 GB

O critério usado para escolher o melhor modelo foi o menor erro absoluto médio em percentagem (MAPE), o qual, para a avaliação de modelos de séries temporais é considerado o melhor, tendo em conta que as séries apresentam apenas valores positivos (Hyndman, R.J., & Athanasopoulos, G., 2018). No entanto, deixamos à disposição do decisor/gestor um conjunto de erros de previsão diferentes para outro tipo de avaliação do modelo, tais como: erro médio, erro médio absoluto entre outros. Para escolher de forma automática o melhor modelo, o R compara o MAPE de todos os modelos e, quando o menor MAPE for encontrado, surge uma mensagem – “Observando os erros absolutos médios em percentagem, é possível concluir que o melhor modelo é o modelo: ...”.

$$\text{MAPE} = \frac{1}{n} \sum_{i=1}^n \frac{|Y_i - \hat{Y}_i|}{Y_i}$$

5. Apresentação de Dados e Discussão de Resultados

A plataforma escolhida para a criação de relatórios foi o Visual Studio, utilizando o *Reporting Services* do SQL Server para disponibilizar os relatórios localmente ou na rede. Os relatórios são automáticos na exibição dos gráficos e tabelas, o utilizador apenas tem que definir o período de tempo que pretende prever.

O utilizador, ao abrir o relatório, tem uma barra no topo da página que serve como menu, tendo seis opções:

- Série Original, é a opção que vem pré-selecionada quando o utilizador abre o relatório, contém gráficos da série e decomposição como também um sumário da série;
- Modelo ARIMA, onde se obtêm as previsões e erros de previsão para o Modelo ARIMA;
- Modelo HW, onde se obtêm as previsões e erros de previsão para o Modelo Holt-Winters;
- Modelo ETS, onde se obtêm as previsões e erros de previsão para o Modelo Alisamento Exponencial;
- Modelo LM, onde se obtêm as previsões e erros de previsão para o Modelo de Regressão Linear;
- Previsões, onde surgem os valores de todos os modelos previstos, com a conclusão de qual o melhor modelo.

Nos modelos será apresentado ao utilizador um *input* para ser definido o número de meses que pretende prever, numa secção do relatório em cima do menu. Para alterar o número de meses a prever, o utilizador pode alterar o valor e pressionar o botão *View Report*, que se encontra no canto superior direito do relatório. É possível alternar entre menus, sem perder o que foi selecionado

anteriormente, basta selecionar o botão de retroceder. Se o utilizador pretender guardar a página do relatório em que se encontra, tem a opção de o guardar em Word, Excel, PowerPoint, PDF, *Data Feed*, e imprimir.

Os relatórios foram construídos de forma a serem de fácil compreensão e úteis para os gestores da empresa. Para isso, em cada uma das abas tem uma explicação básica sobre os modelos. Não foram considerados testes para a aprovação dos modelos, dado que as funções automáticas do *package forecast* garantem na maioria dos casos o melhor modelo possível para a série.

Na aba Série Original, é possível observar um sumário da série com os quartis, mínimo, média e máximo, que dá ao gestor uma ideia inicial de como é o comportamento dos dados, em seguida uma representação gráfica da série e um gráfico com a decomposição da série, com as componentes sazonalidade, tendência e série irregular, que permitem obter informações importantes sobre os comportamentos passados e a encontrar explicações para o porquê desses comportamentos.

Na aba Modelo ARIMA, após a seleção do número de meses a prever, temos uma explicação breve do que é um modelo ARIMA, gráfico da série com a previsão já incluída. Para além da previsão, contém 2 níveis de intervalos de confiança a 80 e 95% para os valores previstos e, ainda, um gráfico de barras com os erros de previsão da série – ME (erro médio), RMSE (raiz do erro médio), MAE (erro absoluto médio), MPE (erro médio em percentagem), MAPE (erro absoluto médio em percentagem) e ACF1 (Teste de precisão de previsão). Nos restantes modelos a estrutura é semelhante.

Na aba Previsões, temos uma tabela com a série prevista para os próximos 12 meses com todos os modelos e em baixo, uma conclusão sobre qual o melhor modelo. O critério para definir qual o melhor modelo nos relatórios desenvolvidos, é o menor erro absoluto médio em percentagem (MAPE), no entanto, este pode ficar ao critério do gestor, pelo que estão incluídas mais seis opções para além do MAPE.

O package forecast fornece 2 tipos de funções para prever Alisamentos exponencias: `ets()` para alisamentos exponenciais de forma geral e `HoltWinters()` para alisamentos exponencias do tipo Holt-Winters. Para criar um modelo com ambas as funções basta inserir no argumento de cada função um objeto do tipo `ts`. Para a função `ets()` o R seleciona, entre diversos modelos possíveis, o modelo que minimiza a função verossimilhança para a série temporal selecionada. Já para a função `HoltWinters()`, o R seleciona, entre diversos modelos possíveis, o modelo que minimiza o erro quadrático médio.

Estes modelos são gerados automaticamente pelas funções do package forecast: `auto.arima()`, para os modelos ARIMA; `HoltWinters()`, para os modelos de Holt-Winters; `ets()`, para os modelos de Alisamento Exponencial; `tslm()`, para os modelos de Regressão Linear. Para realizar a previsão, basta fazer `forecast(modelo, h=nº de períodos a prever)`. Para criar um gráfico simples, basta introduzir `autoplot(forecast)`.

Na Tabela 2, são apresentadas as previsões para a série faturação.

Data	ARIMA	Holt-Winters	Alisamento Exponencial	Regressão Linear
jan/18	109510	149402	102964	110677
fev/18	286983	387130	456034	323346
mar/18	224514	336529	359829	290294
abr/18	173163	243017	214994	198373
mai/18	221552	320676	304486	259635
jun/18	422612	441890	423699	325633
jul/18	337460	389022	288489	265294
ago/18	283667	367434	305334	275736
set/18	527731	597749	641129	449387
out/18	466886	484930	397520	331393
nov/18	602874	615185	526978	448612
dez/18	1068540	1284600	1674240	1200060

Tabela 2 - Síntese das previsões da faturação

O gráfico da Figura 5 apresenta os valores mensais de faturação.

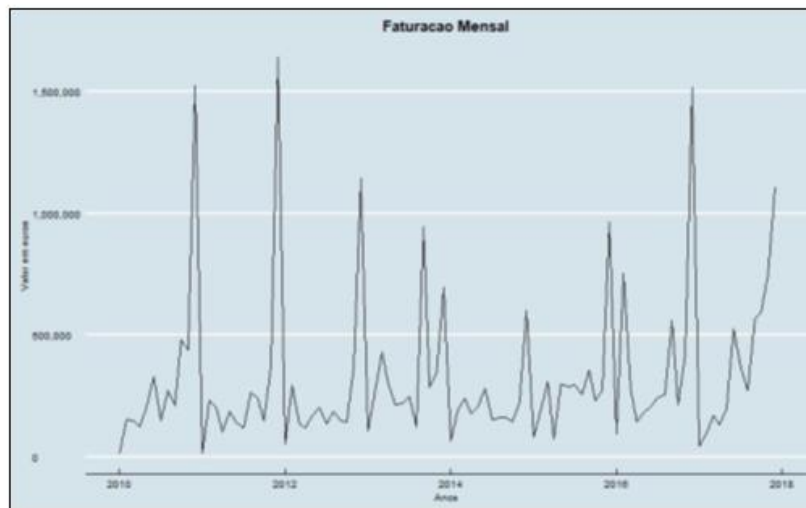


Figura 5 - Faturação mensal

O gráfico da Figura 6 mostra a decomposição da faturação mensal, separando sazonalidades e tendências.

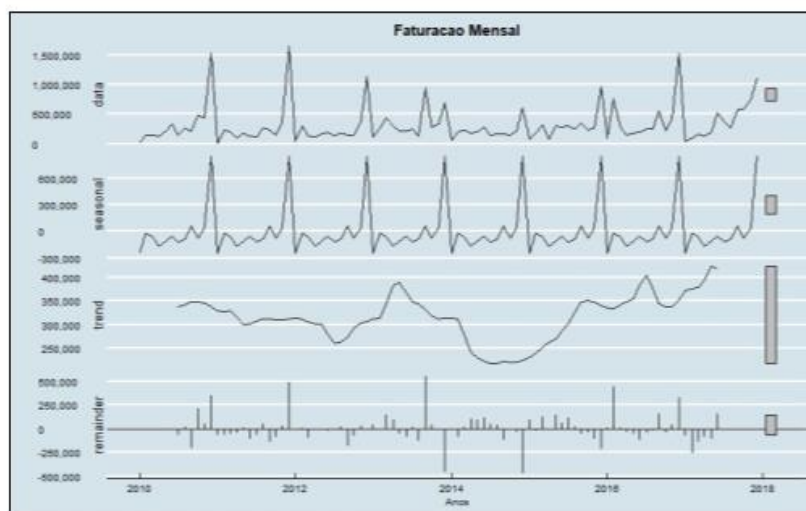


Figura 6 - Decomposição da série Faturação

O gráfico da Figura 7 mostra o modelo de previsão com regressão linear.

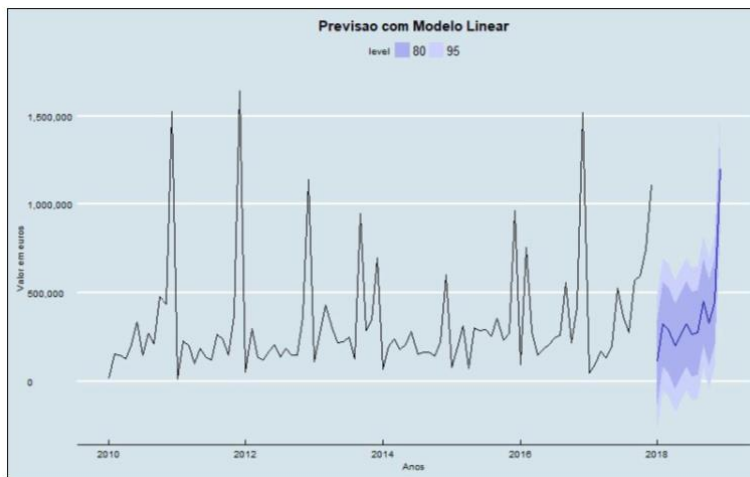


Figura 7 – Modelo de previsão utilizando regressão linear

A Tabela 3 permite comparar os modelos quanto aos erros de previsão, utilizando diversas métricas atrás referidas.

Erros de						
Previsão	ME	RMSE	MAE	MPE	MAPE	ACF1
ARIMA	3312,63	203551,72	126495,9	-49,99	70,1	0,14
Holt-Winters	33232,5	199552	123282	0,9	47,121	0,12
Alisamento						
Exponencial	6484,58	155154,62	87813,62	-51,77	76,16	0,11
Regressão						
Linear		0 161415,81	102243,3	-18,58	37,68	0,16

Tabela 3 - Comparação dos modelos - faturação

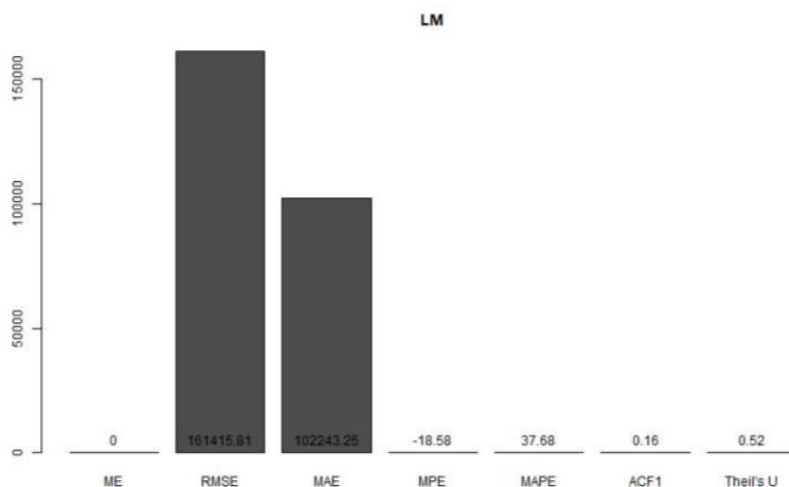


Figura 8 - Erros de previsão

Seguindo o critério de escolha – menor erro médio absoluto em percentagem (MAPE) – chegou-se à conclusão que o modelo que melhor previu a faturação da Quidgest foi o modelo de Regressão linear com 37,68.

A tabela que se segue apresenta os resultados síntese das previsões das encomendas utilizando os métodos de ARIMA, Holt-Winters, alisamento exponencial e regressão linear.

Data	ARIMA	Alisamento		Regressão
		Holt-Winters	Exponencial	Linear
jan/18	369821,8	593005,8	315186,7	450972
fev/18	369821,8	517462,7	197221,3	379656,8
mar/18	369821,8	523462,7	545605,6	577443,7
abr/18	369821,8	549955,7	387676,8	429116,4
mai/18	369821,8	559667,9	340927,4	434064,9
jun/18	369821,8	577014,8	621255	683522,1
jul/18	369821,8	565807	430794,6	415651,7
ago/18	369821,8	589985,6	209251,4	316061
set/18	369821,8	669007,1	229238,2	361667,4
out/18	369821,8	596458,7	217387,6	370092,3
nov/18	369821,8	788013,2	672872,8	437032,2
dez/18	369821,8	1120820,9	1472095,1	1095571,8

Tabela 4 - Síntese das previsões das encomendas

A Tabela 5 permite comparar os modelos quanto aos erros de previsão, utilizando para tal, métricas referidas atrás.

Erros de						
Previsão	ME	RMSE	MAE	MPE	MAPE	ACF1
ARIMA	69401,97	574714,2	200345,6	-123,99	161,39	-0,07
Alisamento						
Exponencial	19443,33	521606,6	210694,2	-131,23	161,84	-0,06
Holt-Winters	44785,17	583814,8	249280,5	-98,16	145,36	-0,05
Regressão						
Linear	0	521404,9	232693,4	-126,99	203,61	-0,08

Tabela 5 - Comparação dos modelos - encomendas

Devido à variação elevada dos dados (vários picos em meses com valores baixos em média), não foi possível efetuar uma boa previsão de encomendas. Contudo, seguindo o critério de escolha – menor erro médio absoluto em percentagem (MAPE) – chegou-se à conclusão que o modelo que melhor previu o valor em encomendas da Quidgest foi o modelo Holt-Winters com 145,36. Em trabalhos futuros, seria ideal usar outros modelos ou outras abordagens.

A tabela que se segue apresenta os resultados síntese das previsões dos incidentes, utilizando os métodos de ARIMA, Holt-Winters, Alisamento exponencial e Regressão linear.

Data	ARIMA	Holt- Winters	Alisamento Exponencial	Regressão Linear
jan/18	500	464	496	469
fev/18	465	426	471	471
mar/18	387	379	414	435
abr/18	280	290	380	394
mai/18	344	356	355	386
jun/18	359	342	294	370
jul/18	417	348	302	395
ago/18	349	245	257	347
set/18	412	317	270	394
out/18	312	313	458	434
nov/18	336	340	392	418
dez/18	274	284	293	368

Tabela 6 - Síntese das previsões para os incidentes

A Tabela 7 permite comparar os modelos quanto aos erros de previsão, utilizando, para tal, diversas métricas atrás referidas.

Erros de						
Previsão	ME	RMSE	MAE	MPE	MAPE	ACF1
ARIMA	-0,09	65,91	45,89	-8,54	26,58	-0,01
Alisamento						
Exponencial	-1,16	72	48,45	-9,71	29,02	0,2
Holt-Winters	1,95	72,38	52,32	-8,89	31,44	0,1
Regressão						
Linear	0	80,34	61,06	-18,17	40,04	1,21

Tabela 7 - Comparação dos modelos – incidentes

Seguindo o critério de escolha – menor erro médio absoluto em percentagem (MAPE) - chegou-se à conclusão que o modelo que melhor previu o nº de incidentes da Quidgest é o modelo de ARIMA com 26,58.

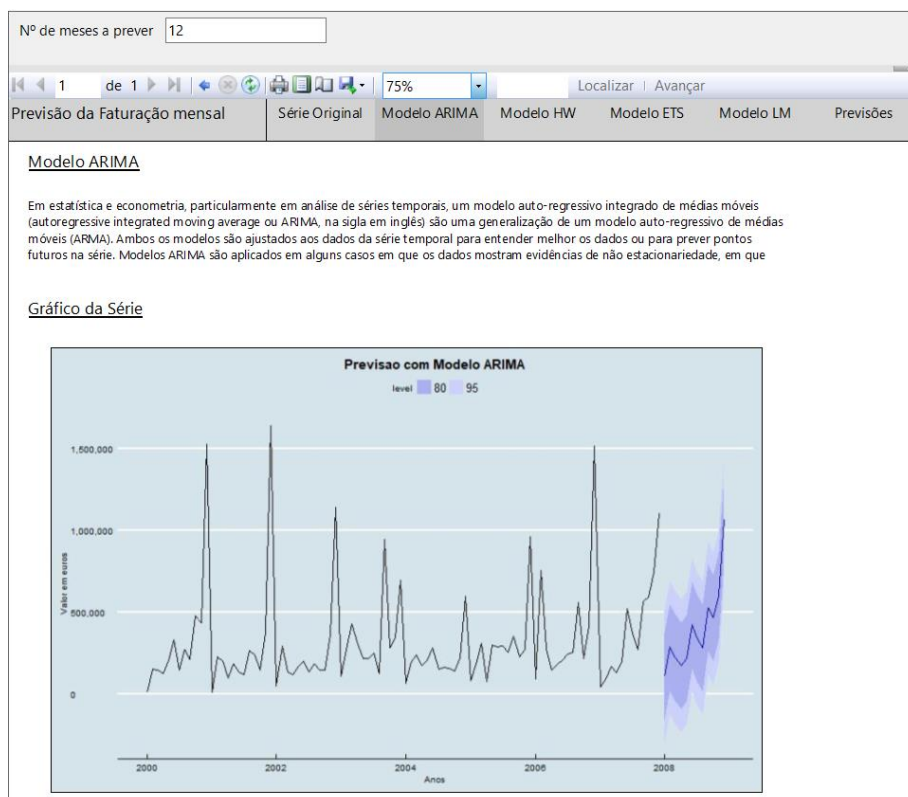


Figura 9 - Layout dos relatórios

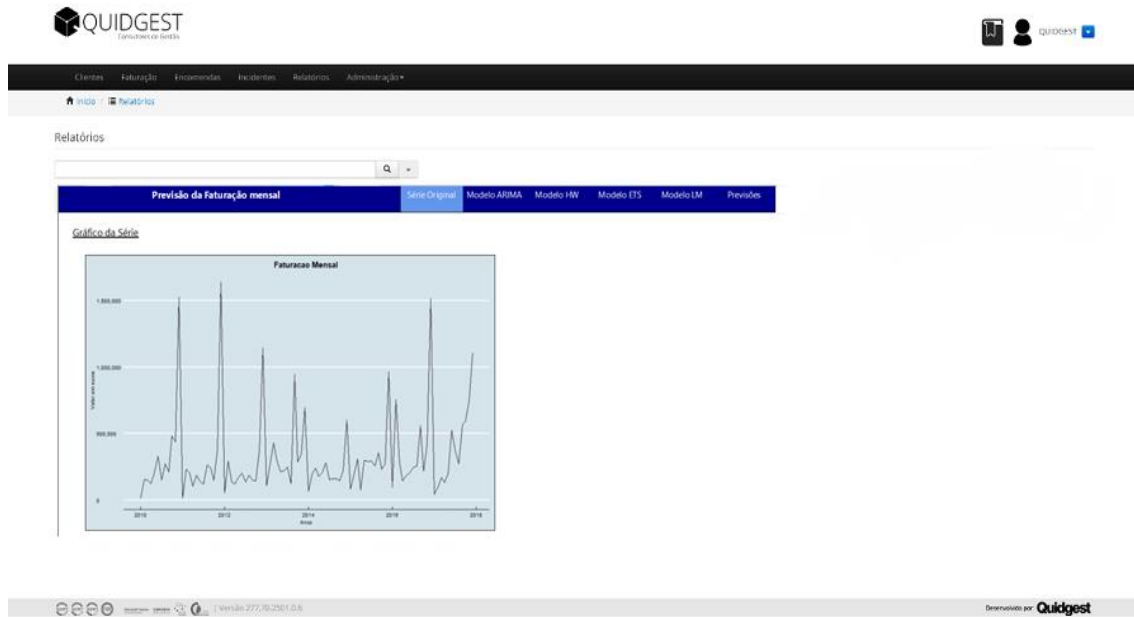


Figura 10 - Relatório integrado num sistema criado na Quidgest

Os testes realizados, embora não sejam significativos para permitir tirar conclusões, permitiram verificar que de acordo com as diversas realidades empresariais, existem técnicas que se adequam mais que outras. Esse facto vai em linha de conta com a literatura. A abordagem seguida permitiu testar a solução proposta, sendo possível verificar que a utilização de gráficos é especialmente relevante.

Quanto à validação da abordagem conceptual utilizada, os resultados também permitiram verificar a sua adequabilidade, tal como em outras situações anteriores (exemplo, Mohamed, et al. 2013 e Galemore 2011).

6. Conclusões e Trabalhos Futuros

Estudo de séries para previsão é particularmente relevante. A sua utilização em contexto empresarial permite, de alguma forma, ajustar a estratégia do negócio. Neste contexto, o trabalho reportado resultou de um estudo concreto em que se procurou identificar qual a melhor forma de prever recorrendo a vários métodos clássicos no contexto da econometria com base em relatórios desenvolvidos com o *R Services* do SQL Server para a sua integração em sistemas de informação. Os resultados apresentados demonstram haver diferenças significativas em função das realidades analisadas. Os erros de previsão associados a incidentes são menores do que nas encomendas ou na faturação. Este aspeto poderá decorrer do facto de estas últimas estarem associadas à resposta tática ou operacional.

Desta forma, a elaboração de metas para a empresa torna-se algo mais realista, motivando os trabalhadores para alcançarem esses objetivos e promovendo a competitividade entre áreas. O marketing é a área que porventura beneficia mais com as previsões, já que, sabendo de antemão os comportamentos sazonais das vendas, pode elaborar ações de marketing de forma mais assertiva.

Espera-se que num futuro próximo seja possível aumentar o número de modelos e diferentes dados a explorar, como por exemplo modelos VAR/VEC e alguns dos modelos mais usados em *Data Mining* como por exemplo *Random Forests*, *SVM*, *Extreme Gradient Boost* e etc.

Um dos resultados relevantes do trabalho realizado foi a produção de relatórios. Estes relatórios, para além de serem importantes como suporte ao trabalho empírico, são significativos como complementos ao sistema de *business intelligence*. As vantagens dos relatórios/ferramentas desenvolvidos com *R Services* são: facilidade com que são inseridos em sistemas desenvolvidos pela

Quidgest; reaproveitamento dos *stored procedures* desenvolvidos para os relatórios na criação de tabelas na base de dados, onde são armazenadas as análises anteriores; rapidez de execução, já que o SQL Server usa uma edição do R criada pela Microsoft, onde o código de algumas funções/*packages* do R foi otimizado, passando esses *packages* a serem escritos em C /C++ em vez de Fortran; Visualização dos resultados do tipo *data frame* na consola do SQL Server, permite verificar se existe algum erro inesperado. No entanto existem desvantagens tais como: não é possível verificar como estão os gráficos na consola do SQL Server, apenas nos relatórios, o que torna o processo de correção bastante mais demorado caso exista algum erro. Os gráficos não são interativos, o *R Services* apenas consegue mostrar ficheiros do tipo imagem (exemplo .jpeg), pelo que por vezes não é perceptível qual o valor que as séries estão a assumir, necessitando de uma tabela auxiliar com os valores; o *R Services* não suporta múltiplas versões do R ao mesmo tempo, apenas uma. Caso exista a necessidade de mudança de versão, só é possível através da alteração da pasta do *R Services* manualmente; por vezes a instalação dos *packages* não é permitida, pelo que alguns *packages* têm de ser instalados por manualmente na pasta *library* do *R Services*; alguns *packages* não estão disponíveis.

Para uma melhor performance dos relatórios e para uma maior diversidade de modelos e técnicas, seria ideal armazenar os modelos em ficheiros Rdata, por exemplo de mês a mês, em vez de recalculá-los a cada pedido do utilizador. A sugestão seria substituir o código da criação do modelo pela leitura de um modelo já criado anteriormente. Com esta alteração cada relatório demoraria um tempo menor a carregar e assim já seria possível usar modelos mais exigentes para o computador.

Em trabalhos futuros seria interessante explorar as capacidades dos *packages* do R, shiny e rmarkdown, para desenvolvimento de aplicações web/relatórios sem necessitar de conhecimentos em HTML, CSS ou Javascript. Com esta aplicação web já seria possível contornar o problema da falta de interatividade presente no *R Services*, já que nas aplicações desenvolvidas com o shiny e rmarkdown é possível utilizar todos os *packages* disponíveis no R.

Referências Bibliográficas

- Bollerslev, T. (1986) Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, 31(3), 307–327.
- Box, G & Jenkins, G (1970). *Time Series Analysis: Forecasting and Control*. San Francisco: Holden-Day.
- Brown, R G. (1956). *Exponential Smoothing for Predicting Demand*. Cambridge, Massachusetts: Arthur D. Little Inc. p. 15.
- Caetano, T. V., & Costa, C. J. (2014). Data Warehousing num contexto de Sistemas Integrados. In *Atas da Conferência da Associação Portuguesa de Sistemas de Informação* (Vol. 12, pp. 186-199).
- Costa C.J., Antunes P., Dias J.F. (2002) Integrating Two Organizational Systems through Communication Genres. In: Arbab F., Talcott C. (eds) *Coordination Models and Languages. COORDINATION 2002. Lecture Notes in Computer Science*, vol 2315. Springer, Berlin, Heidelberg
- Doob, J. (1934). "Stochastic Processes and Statistics". Proceedings of the National Academy of Sciences of the United States of America. 20 (6): 376–379.
- Engle, R.F. (1982) Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation. *Econometrica*, 50(4), 987–1007.
- Evans, J. R., & Lindner, C. H. (2012). Business analytics: the next frontier for decision sciences. *Decision Line*, 43(2), 4-6.
- Evelson, B., & Norman, N. (2008). Topic overview: Business intelligence. *Forrester research*, 61.
- Galemore, C. A. (2011). "Immunization information system-progress on integration of school nurses: a multi-state roundtabl", *NASN School Nurse*, 83-86.
- Harrison, P.J. Stevens, C.F. (1976) "Bayesian Forecasting," *Journal of the Royal Statistical Society Series B*, 38(3), 205-247

- Hasselbring, W. (2000). "Information system integration", *Communications of the ACM*, 43 (6), 32-38
- Hyndman, R.J., & Athanasopoulos, G. (2018) *Forecasting: principles and practice*, 2nd edition, OTexts: Melbourne, Australia. OTexts.com/fpp2
- Holt, C. (1957). "Forecasting Trends and Seasonal by Exponentially Weighted Averages". Office of Naval Research Memorandum. 52. reprinted in Holt, Charles C. (January–March 2004). "Forecasting Trends and Seasonal by Exponentially Weighted Averages". *International Journal of Forecasting*. 20 (1): 5–10. doi:10.1016/j.ijforecast.2003.09.015.
- Ji, S. Yu, H., Guo, Y. & Zhang, Z. (2016). Research on sales forecasting based on ARIMA and BP neural network combined model. In *Proceedings of the 2016 International Conference on Intelligent Information Processing (ICIIP '16)*. ACM, New York, NY, USA, , Article 41 , 6 pages. DOI: <https://doi.org/10.1145/3028842.3028883>
- Khairalla, MA. & Ning., M. (2017). Financial Time Series Forecasting Using Hybridized Support Vector Machines and ARIMA Models. In *Proceedings of the 2017 International Conference on Wireless Communications, Networking and Applications (WCNA 2017)*, Srikanta Patnaik (Ed.). ACM, New York, NY, USA, 94-98. DOI: <https://doi.org/10.1145/3180496.3180613>
- Krishnamurthy, V. & Yin, G.G. (2002) Recursive algorithms for estimation of hidden Markov models and autoregressive models with Markov regime. *IEEE Transactions on Information Theory*, 48(2), 458–476.
- Liberatore, M. J., & Luo, W. (2010). The analytics movement: Implications for operations research. *Interfaces*, 40(4), 313-324.
- Lineesh, M.C. & John, C.J. (2010) Analysis of non-stationary time series using wavelet decomposition. *Nature and Science*, 8(1), 53–59.
- March, S., & Smith, G.(1995) Design and Natural Science Research on Information Technology. *Decision Support Systems*, 15.

- Mohamed, N., Mahadi, B., Miskon, S., Haghshenas, H. & Adnan, H. (2013). "Information System Integration: A Review of Literature and a Case Analysis", *Mathematics and Computers in Contemporary Science*, World Scientific and Engineering Academy and Society (WSEAS), Editors: Reinhard Neck, 68-77
- Nofal, M. I., & Yusof, Z. M. (2013). Integration of business intelligence and enterprise resource planning within organizations. *Procedia Technology*, 11, 658-665.
- Nunamaker, J.F., Chen, M., and Purdin, T.D.M. (1991) Systems Development in Information Systems Research. *Journal of Management Information Systems*, 7, 3.
- Pati, J. & Shukla, K. (2015). A Hybrid Technique for Software Reliability Prediction. In *Proceedings of the 8th India Software Engineering Conference (ISEC '15)*. ACM, New York, NY, USA, 139-146.
DOI=<http://dx.doi.org/10.1145/2723742.2723756>
- Peffer, K., Tuunanen, T., Rothenberger, M. A., & Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of management information systems*, 24(3), 45-77.
- Polkowski, Z., Constantin, D., Raducu, A. & Blidaru, E. (2016). "The Integration of BI, ERP and CRM Systems", *Research Bulletin the Jan Wyzkowski University. Studies in Technical Sciences* 2016(5), 49-63.
- Power, D. J. (2007). A brief history of decision support systems. DSSResources.COM, World Wide Web, <http://DSSResources.com/history/dsshhistory.html>, version, 4..
- Puente, F, Hernandez, C. & Salcedo, O. (2009). Comparative analysis of time series techniques ARIMA and ANFIS to forecast Wimax traffic. In *Proceedings of the 7th International Conference on Advances in Mobile*

- Computing and Multimedia* (MoMM '09). ACM, New York, NY, USA, 277-281.
DOI=<http://dx.doi.org/10.1145/1821748.1821801>
- Russman. R, Seymour. L, & Belle. J (2017). "Integrating BI Information into ERP Processes - Describing Enablers", Proceedings of the 19th International Conference on Enterprise Information Systems (ICEIS 2017) - Volume 1, 241-248.
- Sekgweleo, T., Billawer, J. & Hamunyela, L. (2016). "Integration Framework for Information Systems Coexistence within Organization", *Management and Organizational Studies*, Vol 3 , No. 1, 17-26.
- Simon, H. (1969) *The sciences of the artificial*. Cambridge MA: MIT Press.
- Sims, C. (1980) "Macroeconomics and Reality," *Econometrica*, Vol. 48, No. 1, pp. 1–48, January.
- Tomas, D, Costa, C., Gaivao, J.P. & Carvalho J.P. (2018) "Series Temporais para a Previsão de Incidentes, encomendas e faturação" *CAPSI 2018 Atas da 18 Conferencia da Associação Portuguesa de Sistemas de Informação* ; Santarém 12 e 13 de Outubro.
- Tong, H. (1990) *Non-Linear Time Series: A Dynamical System Approach*, Clarendon Press, Oxford, UK.
- Walker, G. (1931) "On Periodicity in Series of Related Terms", *Proceedings of the Royal Society of London*, Ser. A, Vol. 131, 518–532.
- Walls, J., Widmeyer, G., and El Sawy, (1992) O. Building an Information System Design Theory for Vigilant EIS. *Information Systems Research*, 3, 1.
- Winters, P. R. (1960). "Forecasting Sales by Exponentially Weighted Moving Averages". *Management Science*. 6 (3): 324–342. doi:10.1287/mnsc.6.3.324.
- Wold, H. (1954) *A Study in the Analysis of Stationary Time Series*, Second revised edition, Almqvist & Wiksell
- Yule, G. (1927) "On a Method of Investigating Periodicities in Disturbed Series, with Special Reference to Wolfer's Sunspot Numbers", *Philosophical Transactions of the Royal Society of London*, Ser. A, Vol. 226, 267–298.]

Zhang, G.P. (2012) Neural networks for time-series forecasting, in *Handbook of Natural Computing*, Rozenberg, G., Bäck, T. and Kok, J. (eds), Springer, Berlin, Germany, pp. 461–477.

Anexos

```
--Gráfico da série temporal
ALTER PROCEDURE plot_serie
AS
BEGIN
SET NOCOUNT ON;
EXEC sp_execute_external_script
    @language = N'R'
    ,@script = N'library("forecast")
                library("ggplot2")
                library("scales")
                library("ggthemes")

df = InputDataSet;
faturas = ts(df,frequency=12,start=c(2010,1));
image_file = tempfile();
jpeg(filename=image_file, width=800, height=500);
plot=autoplot(faturas, main="Faturacao Mensal",xlab="Anos",ylab="Valor em euros")+
  scale_y_continuous(labels = comma)+
  theme_economist()+
  theme(plot.title = element_text(hjust=0.5));
print(plot);
dev.off();
OutputDataSet = data.frame(data=readBin(file(image_file,"rb"),what=raw(),n=1e6));'
    ,@input_data_1 = N'select SUM(VALORTOT) AS VALOR
                    from QUIfactu
                    WHERE DATA>'20091231'' AND VALORTOT>0
                    GROUP BY YEAR(DATA), MONTH(DATA)
                    ORDER BY YEAR(DATA), MONTH(DATA)'
    ,@parallel=1
WITH RESULT SETS ((Plot varbinary(max)));
END
GO
```

Figura 11 - *Stored Procedure* do gráfico da série temporal – Faturação

```

--Gráfico da série temporal decomposta
--Série original, componente sazonal, componente tendência e componente irregular
alter PROCEDURE plot_decomposicao
AS
BEGIN
SET NOCOUNT ON;
EXEC sp_execute_external_script
    @language = N'R'
    ,@script = N'library("forecast")
                library("ggplot2")
                library("scales")
                library("ggthemes")

df = InputDataSet;
faturas = ts(df,frequency=12,start=c(2010,1));
dfaturas = decompose(faturas)
image_file = tempfile();
jpeg(filename=image_file, width=800, height=500);
plot=autoplot(dfaturas,main="Faturacao Mensal",xlab="Anos")+
    scale_y_continuous(labels = comma)+
    theme_economist()+
    theme(plot.title = element_text(hjust=0.5));
print(plot);
dev.off();
OutputDataSet = data.frame(data=readBin(file(image_file,"rb"),what=raw(),n=1e6));'
    ,@input_data_1 = N'select SUM(VAlORTOT) AS VALOR
                    from QUIfactu
                    WHERE DATA>'20091231' AND VALORTOT>0
                    GROUP BY YEAR(DATA), MONTH(DATA)
                    ORDER BY YEAR(DATA), MONTH(DATA)'

    ,@parallel=1
    WITH RESULT SETS ((Plot varbinary(max)));
END
go

```

Figura 12 - *Stored procedure* do gráfico decomposição da série temporal – Faturação

```

--Gráfico da previsão do modelo Arima
ALTER PROC plot_arima
@h nvarchar(max)
AS
BEGIN
SET NOCOUNT ON;
EXEC sp_execute_external_script
    @language = N'R'
    ,@script = N'library("forecast")
                library("ggplot2")
                library("scales")
                library("ggthemes")

df = InputDataSet;
serie = ts(df,frequency=12,start=c(2010,1));
ARIMA = auto.arima(serie)
ARIMAforecast = forecast(ARIMA, h=h)
image_file = tempfile();
jpeg(filename=image_file, width=800, height=500);
plot=autoplot(ARIMAforecast,main="Previsao com Modelo ARIMA", xlab="Anos", ylab="Valor em euros")+
  scale_y_continuous(labels = comma)+
  scale_x_continuous(breaks= pretty_breaks())+
  theme_economist()+
  theme(plot.title = element_text(hjust=0.5))
print(plot);
dev.off();
OutputDataSet = data.frame(data=readBin(file(image_file,"rb"),what="raw()",n=1e6));'
    ,@input_data_1 = N'select SUM(VALORTOT) AS VALOR
                    from QUIfactu
                    WHERE DATA>'20091231' AND VALORTOT>0
                    GROUP BY YEAR(DATA), MONTH(DATA)
                    ORDER BY YEAR(DATA), MONTH(DATA)'
    ,@parallel=1
    ,@params=N'@h nvarchar(max)'
    ,@h=@h
    WITH RESULT SETS ((plot varbinary(max)));
END
go

```

Figura 13 - *Stored procedure* do gráfico com previsão da série temporal – Faturação (ARIMA)

```

--Gráfico da previsão do modelo Holt-Winters
ALTER PROCEDURE plot_hw
@h nvarchar(max)
AS
BEGIN
SET NOCOUNT ON;
EXEC sp_execute_external_script
@language = N'R'
,@script = N'library("forecast")
           library("ggplot2")
           library("scales")
           library("ggthemes")

df = InputDataSet;
serie = ts(df,frequency=12,start=c(2010,1));
HW = HoltWinters(serie);
HWforecast = forecast(HW, h=h);
image_file = tempfile();
jpeg(filename=image_file, width=800, height=500);
plot=autoplot(HWforecast, main="Previsao com Modelo Holt-Winters", xlab="Anos", ylab="Valor em euros")+
  scale_y_continuous(labels = comma)+
  scale_x_continuous(breaks= pretty_breaks()+
  theme_economist()+
  theme(plot.title = element_text(hjust=0.5));
print(plot);
dev.off();
OutputDataSet = data.frame(data=readBin(file(image_file,"rb"),what=raw(),n=1e6));'
,@input_data_1 = N'select SUM(VALORTOT) AS VALOR
                  from QUIfactu
                  WHERE DATA>'20091231'' AND VALORTOT>0
                  GROUP BY YEAR(DATA), MONTH(DATA)
                  ORDER BY YEAR(DATA), MONTH(DATA)'
,@parallel=1
,@params=N'@h nvarchar(max)'
,@h=@h
WITH RESULT SETS ((Plot varbinary(max)));
END
GO

```

Figura 14 - *Stored procedure* do gráfico com previsão da série temporal – Faturação (Holt-Winters)

```

--Gráfico da previsão do modelo Alisamento exponencial
ALTER PROCEDURE plot_ets
AS
BEGIN
SET NOCOUNT ON;
EXEC sp_execute_external_script
    @language = N'R'
    ,@script = N'library("forecast")
                library("ggplot2")
                library("scales")
                library("ggthemes")

df = InputDataSet;
tabela = ts(df,frequency=12,start=c(2010,1));
ETS = ets(tabela);
ETSforecast = forecast(ETS, h=12);
image_file = tempfile();
jpeg(filename=image_file, width=800, height=500);
a=autoplot(ETSforecast, main="Previsao com Modelo ETS", xlab="Anos", ylab="Valor em euros")+
  scale_y_continuous(labels = comma)+
  scale_x_continuous(breaks= pretty_breaks())+
  ylim(0,2000000)+
  theme_economist()+
  theme(plot.title = element_text(hjust=0.5));
print(a);
dev.off();
OutputDataSet = data.frame(data=readBin(file(image_file,"rb"),what=raw(),n=1e6));'
    ,@input_data_1 = N'select SUM(VALORTOT) AS VALOR
                        from QUIfactu
                        WHERE DATA>'20091231'' AND VALORTOT>0
                        GROUP BY YEAR(DATA), MONTH(DATA)
                        ORDER BY YEAR(DATA), MONTH(DATA)'
    ,@parallel=1
WITH RESULT SETS ((plot varbinary(max)));
END
go

```

Figura 15 - *Stored procedure* do gráfico com previsão da série temporal – Faturação (Alisamento Exponencial)


```

--Gráfico com a previsão do modelo Regressão Linear
alter PROCEDURE plot_lm
@h nvarchar(max)
AS
BEGIN
SET NOCOUNT ON;
EXEC sp_execute_external_script
@language = N'R'
,@script = N'library("forecast")
          library("ggplot2")
          library("scales")
          library("ggthemes")

df = InputDataSet;
tabela = ts(df,frequency=12,start=c(2010,1));
LM = tslm(tabela~trend+season);
LMforecast = forecast(LM, h=h);
image_file = tempfile();
jpeg(filename=image_file, width=800, height=500);
a=autoplot(LMforecast ,main="Previsao com Modelo Linear", xlab="Anos", ylab="Valor em euros")+
  scale_y_continuous(labels = comma)+
  scale_x_continuous(breaks= pretty_breaks())+
  theme_economist()+
  theme(plot.title = element_text(hjust=0.5));
print(a);
dev.off();
OutputDataSet = data.frame(data=readBin(file(image_file,"rb"),what="raw()",n=1e6));'
,@input_data_1 = N'select SUM(VALORTOT) AS VALOR
                  from QUIfactu
                  WHERE DATA>'20091231' AND VALORTOT>0
                  GROUP BY YEAR(DATA), MONTH(DATA)
                  ORDER BY YEAR(DATA), MONTH(DATA)'
,@parallel=1
,@params=N'@h nvarchar(max)'
,@h=@h
WITH RESULT SETS ((plot varbinary(max)));
END
go

```

Figura 16 - *Stored procedure* do gráfico com previsão da série temporal – Faturação (Regressão Linear)

```

--Conclusão de qual o modelo que minimiza o erro absoluto médio em percentagem (MAPE).
--Contem os modelos Arima, Holt-Winters, Alisamento exponencial e regressão linear
ALTER PROCEDURE selectmodelo
AS
BEGIN
SET NOCOUNT ON;
EXEC sp_execute_external_script
@language = N'R'
,@script = N'library("TTR")
library("forecast")
library("tseries")

df = InputDataSet;
serie = ts(df,frequency=12,start=c(2010,1));
HW = HoltWinters(serie);
HWForecast = forecast(HW, h=12);
ARIMA = auto.arima(serie);
ARIMAForecast = forecast(ARIMA, h=12);
ETS = ets(serie);
ETSForecast = forecast(ETS, h=12);
LM = tslm(serie-trend+season);
LMForecast = forecast(LM, h=12);
EARIMA = summary(ARIMA)
EHW = accuracy(HWforecast$fitted,serie)
EETS = summary(ETS)
ELM = accuracy(LM$fitted.values,serie);
(if (EARIMA[5]<EHW[5] & EARIMA[5]<EETS[5] & EARIMA[5]<ELM[5]){
x="Observando os erros absolutos medios em percentagem, podemos concluir que o melhor modelo segundo este criterio (o menor), e o modelo: ARIMA"
}
else if(EHW[5]<EARIMA[5] & EHW[5]<EETS[5] & EHW[5]<ELM[5]){
x="Observando os erros absolutos medios em percentagem, podemos concluir que o melhor modelo segundo este criterio (o menor), e o modelo: Holt-winters"
}
else if(EETS[5]<EHW[5] & EETS[5]<EARIMA[5] & EETS[5]<ELM[5]){
x="Observando os erros absolutos medios em percentagem, podemos concluir que o melhor modelo segundo este criterio (o menor), e o modelo: Alisamento Exponencial"
}
else if(ELM[5]<EHW[5] & ELM[5]<EETS[5] & ELM[5]<EARIMA[5]){
x="Observando os erros absolutos medios em percentagem, podemos concluir que o melhor modelo segundo este criterio (o menor), e o modelo: Regressão Linear"
})

OutputDataSet = data.frame(x);'
,@input_data_1 = N'select SUM(VALORTOT) AS VALOR
from QUIFactu
WHERE DATA='20091231' AND VALORTOT>0
GROUP BY YEAR(DATA), MONTH(DATA)
ORDER BY YEAR(DATA), MONTH(DATA)'

,@parallel=1
WITH RESULT SETS ((Conclusão varchar(max)));
END
GO

```

Figura 17 - *Stored procedure* da conclusão de qual o melhor modelo da série temporal – Faturação

```

--Sumário dos dados da série - Mínimo, 1ºQuartil, Mediana, Média, 3ºQuartil, Máximo
alter PROCEDURE sumario_serie
AS
BEGIN
SET NOCOUNT ON;
EXEC sp_execute_external_script
@language = N'R'
,@script = N' df = InputDataSet;
min=round(min(df),0);
quantile=round(quantile(df[,1]),0);
media=round(mean(df[,1]),0);
OutputDataSet = data.frame(min,quantile[2],quantile[3],media,quantile[4],quantile[5]);'
,@input_data_1 = N'select SUM(VALORTOT) AS VALOR
from QUIFactu
WHERE DATA='20091231' AND VALORTOT>0
GROUP BY YEAR(DATA), MONTH(DATA)
ORDER BY YEAR(DATA), MONTH(DATA)'
,@parallel=1
WITH RESULT SETS ((Mínimo varchar(max),[1º Quartil] varchar(max), Mediana varchar(max),Média varchar(max),[3º Quartil] varchar(max),Máximo varchar(max)));
END
GO

```

Figura 18 - *Stored procedure* do sumário dos dados da série temporal – Faturação

```

--Tabela com as os valores das previsões dos modelos Arima, Holt-Winters, Alisamento exponencial e Regressão Linear
ALTER PROCEDURE previsao
AS
BEGIN
SET NOCOUNT ON;
EXEC sp_execute_external_script
    @language = N'R'
    ,@script = N'library("TTR")
                library("forecast")
                library("tseries")

df = InputDataSet;
serie = ts(df,frequency=12,start=c(2000,1));
HW = HoltWinters(serie);
HWforecast = forecast(HW, h=12);
ARIMA = auto.arima(serie);
ARIMAforecast = forecast(ARIMA, h=12);
ETS = ets(serie);
ETSforecast = forecast(ETS, h=12);
LM = tslm(serie~trend+season);
LMforecast = forecast(LM, h=12);
OutputDataSet = data.frame(ARIMAforecast[4],HWforecast[4],ETSforecast[2],LMforecast[2]);'
    ,@input_data_1 = N'select SUM(VALORTOT) AS VALOR
                    from QUIfactu
                    WHERE DATA>'20091231'' AND VALORTOT>0
                    GROUP BY YEAR(DATA), MONTH(DATA)
                    ORDER BY YEAR(DATA), MONTH(DATA)'
    ,@parallel=1
WITH RESULT SETS ((Previsão_ARIMA varchar(max),Previsão_HW varchar(max),Previsão_ETS varchar(max), Previsão_LM varchar(max)));
END
go

```

Figura 19 - *Stored procedure* das previsões obtidas com todos os modelos para a série temporal – Faturação

Instalação e configuração do *R Services* no SQL Server

Os requisitos mínimos para instalar o *R Services* são os seguintes - 1 GB de memória RAM, 6 GB de disco disponíveis e um processador de 2.0 GHz.

Para instalar o *R Services* numa instância de SQL Server é necessário abrir o SQL Server Installation Center e seleccionar a primeira opção do menu *Installation* - *New SQL Server stand-alone installation or add features to an existing installation*.

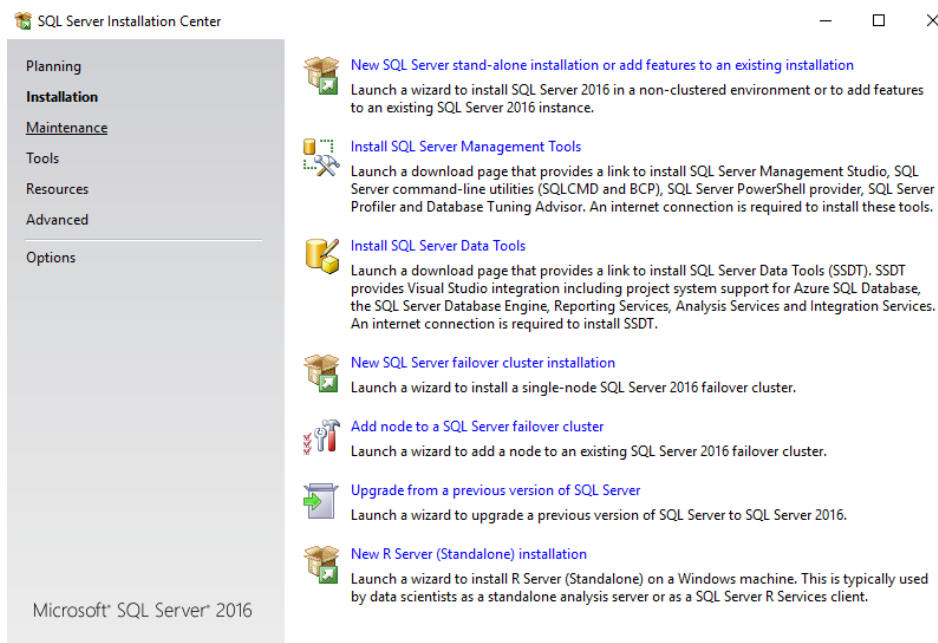


Figura 20 - Página inicial do *SQL Server Installation Center*

Em seguida seleccionamos se pretendemos instalar um novo SQL Server ou adicionar ferramentas a uma instância SQL Server já existente. Para efeitos demonstrativos vão ser apresentadas as opções referentes a uma nova instalação de SQL Server.

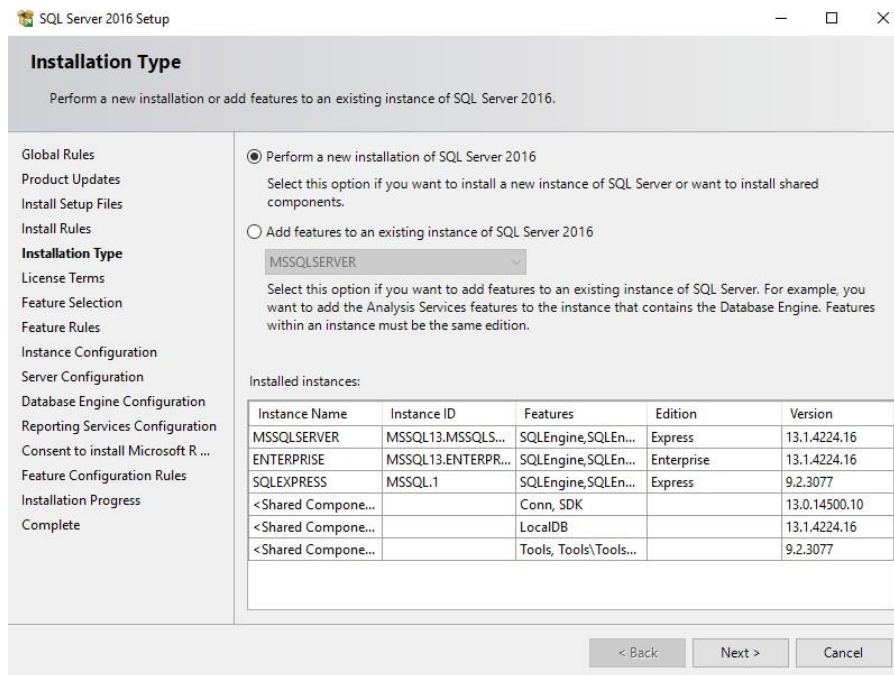


Figura 21 - Seleção do tipo de instalação do SQL Server

Para o próximo passo da instalação do *R Services* é necessário selecionar o campo – *R Services (in-Database)*. Para criar os relatórios desenvolvidos ao longo deste trabalho é necessário selecionar as opções marcadas abaixo.

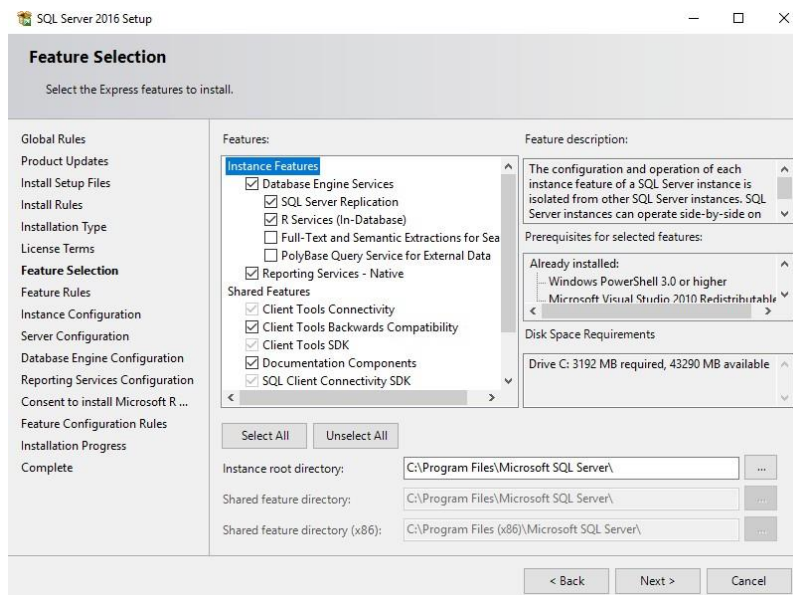


Figura 22 - Seleção das funcionalidades do SQL Server

Os passos seguintes passam pela escolha do nome do SQL Server, palavra passe do mesmo e aceitar e confirmar as seleções realizadas anteriormente.

Após a instalação estar completa é necessária a configuração do *R Services*. Para isso é preciso ter instalado o *Microsoft SQL Server Management Studio* que permite aceder aos servidores SQL Server instalados no computador ou na rede. Ao fazer login no servidor instalado abrimos uma folha de *script sql* em *New Query*. Colocamos o seguinte código na folha – *sp_configure 'external scripts enabled'*, e fazemos F5. Devemos obter o seguinte output:

name	minimum	maximum	config_value	run_value
external scripts enabled	0	1	0	0

Figura 23 - Resultado inicial do comando *sp_configure 'external scripts enabled'*

O *R Services* ainda não está cativo, por isso colocamos o seguinte código na folha - *sp_configure 'external scripts enabled', 1 reconfigure with override*, e fazemos F5. Devemos obter o seguinte output:

name	minimum	maximum	config_value	run_value
external scripts enabled	0	1	1	0

Figura 24 - Resultado intermédio do comando *sp_configure 'external scripts enabled'*

Para completar a configuração é necessário desligar o *Microsoft SQL Server Management Studio* e fazer *restart* aos serviços do SQL Server, nomeadamente o *Launchpad* que é o serviço do SQL Server que suporta e faz a integração da linguagem R com o SQL Server, por outra palavras, é o que permite criar os *stored procedures*.

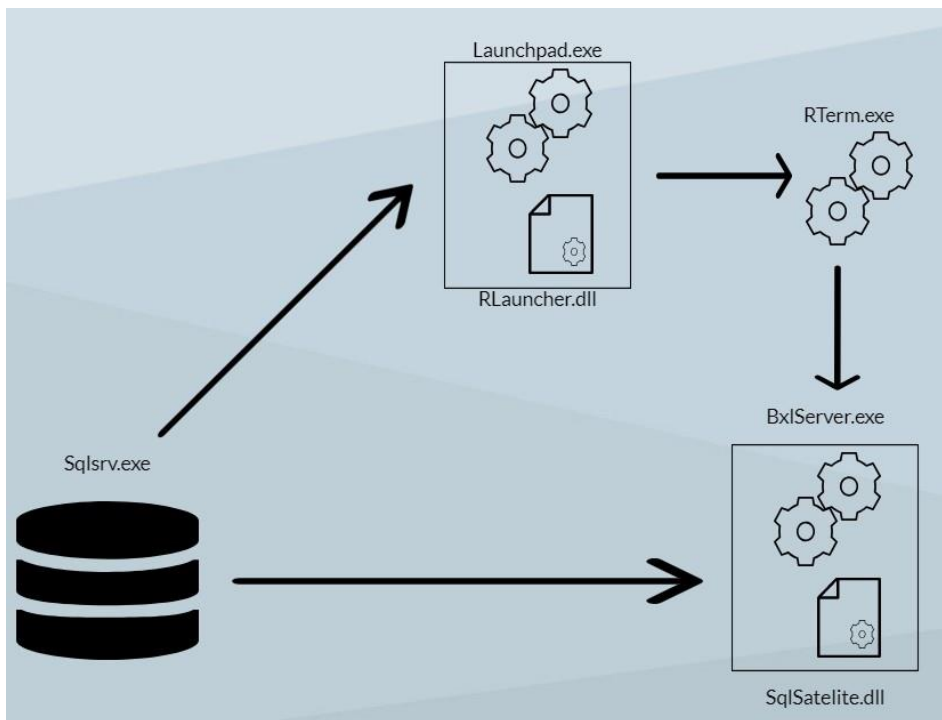


Figura 25 - Esquema do funcionamento interno do R Services

Em seguida para verificar que está tudo configurado de forma correta basta testar um *stored procedure* com código R, deixo em baixo um exemplo básico.

```
EXEC sp_execute_external_script @language = N'R',
    @script = N'OutputDataSet = InputDataSet;',
    @input_data_1 = N'SELECT 1 AS [ISEG]'
WITH RESULT SETS ([[ISEG] int])
GO
```

Results	
	ISEG
1	1

Figura 26 – *Stored procedure* de validação da integração

Caso o resultado seja o apresentado acima, o *R Services* está corretamente configurado.