



Mestrado em Gestão de Sistemas e Tecnologias de Informação

SaaS – SCP (Sports Consulting Platform)

Dissertação para obtenção do Grau de Mestre

Elaborado por André Filipe Batista Xavier

Discente N° 20141910

Orientador Professor Doutor Alexandre Humberto dos Santos Barão

Barcarena

Setembro de 2019

New Atlântica

Mestrado em Gestão de Sistemas e Tecnologias de Informação

SaaS – SCP (Sports Consulting Platform)

Dissertação para obtenção do Grau de Mestre

Elaborado por André Filipe Batista Xavier

Discente Nº 20141910

Orientador Professor Doutor Alexandre Humberto dos Santos Barão

Barcarena

Setembro de 2019

O autor é o único responsável pelas ideias expressas neste documento.

“Tente uma, duas, três vezes e se possível tente a quarta, a quinta e quantas vezes for necessário. Só não desista nas primeiras tentativas, a persistência é amiga da conquista. Se você quer chegar onde a maioria não chega, faça o que a maioria não faz.”

– Bill Gates

Agradecimentos

Aos meus pais, António e Cristina, e irmã Inês, pelo apoio fundamental que me deram ao longo dos últimos cinco anos, três de Licenciatura e dois de Mestrado.

A todos os meus amigos, colegas e professores, quer da Licenciatura em Sistemas e Tecnologias da Informação, quer do Mestrado em Gestão de Sistemas e Tecnologias da Informação da Atlântica, por terem feito de mim uma pessoa melhor e com mais conhecimento dentro da minha área.

Ao orientador deste trabalho, Professor Doutor Alexandre Humberto dos Santos Barão, pelo apoio e por toda a ajuda fornecida ao longo da minha caminhada como estudante, foi fundamental quer para a Licenciatura quer para o Mestrado.

Ao Professor Doutor Joaquim Guerreiro Marques, que apesar de só ter tido o privilégio de ser discente numa cadeira de Licenciatura, foi um suporte muito grande para que esta tese fosse feita porque me foi motivando ao longo do tempo.

Lista de Abreviaturas e Siglas

API – Application Programming Interface
BI – Business Intelligence
CDSS – Collaborative Decision Support Systems
CIS – Sistemas de Comunicações e Informações
CSS – Cascading Style Sheet
DSR – Design Science Research
DSS – Sistemas de Suporte à Decisão
ETL – Extract, Transform and Load
FSM – Framework SaaS Multifuncional
HTML – HyperText Markup Language
IaaS – Infrastructure as a Service
IBM – International Business Machines Corporation
IDE – Ambiente de Desenvolvimento Integrado
IDSS – Sistemas Inteligentes de Suporte à Decisão
GC – Gestão de Conhecimento
KMS – Knowledge Management Systems
MIS – Sistemas de Informações de Gestão
MVC – Model-View-Controller
PHP – Hypertext Preprocessor
ROI - Return on Investment
SaaS – Software as a Service
SDK – Software Development Kit
SI – Sistemas de Informação
SO – Sistema Operativo
PaaS – Platform as a Service
TI – Tecnologias de Informação
UML– Unified Modeling Language

Resumo

O presente documento resulta de um trabalho de pesquisa efetuado no contexto da tese de mestrado do autor, que tem por título: SaaS - Sports Consulting Platform (SCP). Para suportar o desenvolvimento do sistema, foram referidos neste trabalho temas como a Gestão de Conhecimento, Sistemas de Suporte à Decisão, *Cloud Computing*, Aplicações *Cross-Platform* e *Engenharia Web*.

Com base na revisão da literatura, apresenta-se a arquitetura da plataforma SaaS - SCP, respetivos diagramas de classes, casos de uso, diagrama de componentes em UML, requisitos funcionais e não funcionais, bem como tecnologias de desenvolvimento adotadas.

O trabalho tem como objetivo principal definir uma *framework* SaaS multifuncional para auxiliar a tomada de decisão de um apostador desportivo com base em dados estatísticos, tendo uma aplicação (“Pré Live”) específica para a obtenção de probabilidades antes de começar o jogo e, outra Aplicação (“Live”) para determinar probabilidades com base na estatística do próprio jogo.

É ainda apresentado um esquema navegacional que inclui diversos *mockups* para o protótipo a realizar.

Palavras chave

Gestão de Conhecimento, Sistemas de Suporte à Decisão, SaaS, FMS, SCP, Apostas em Portugal.

Abstract

This document is the result of a research work carried out in the context of the author's master's thesis, entitled: SaaS - Sports Consulting Platform (SCP). To support the development of the system, topics such as Knowledge Management, Decision Support Systems, Cloud Computing, Cross-Platform Applications, and Web Engineering were mentioned in this work.

Based on the literature review, the author presents the multifunctional SaaS - SCP framework, class diagrams, use cases, UML components diagrams, functional and non-functional requirements and technologies adopted.

This work is intended to assist a sports gambler's decision making based on statistical data, having a specific application (“Pré Live”) for obtaining probabilities before starting the game and another application (“Live”) for determining probabilities based on the statistics of the game itself.

A prototype navigational scheme is also presented, featuring various application mockups representing the future look of the system.

Keywords

Knowledge Management, Decision Support Systems, SaaS, FMS, SCP, Betting in Portugal.

Índice

1. Introdução.....	14
1.1. Contexto.....	14
1.2. Problema.....	14
1.3. Objetivos de Investigação.....	15
1.4. Método de Investigação.....	15
1.5. Estrutura do Documento.....	16
2. Revisão da Literatura.....	17
2.1. Gestão de Conhecimento.....	17
2.1.1. Sistemas da Gestão de Conhecimento.....	17
2.1.2. Classificação da Gestão de Conhecimento.....	19
2.1.3. Gestão de Conhecimento nas Organizações.....	22
2.1.4. Estrutura da Gestão de Conhecimento da Inovação de Processos.....	23
2.2. Sistemas de Suporte à Decisão.....	25
2.2.1. Estratégia nas organizações para a aplicação dos SDD.....	26
2.2.2. Tipos de SDD.....	27
2.2.3. Processo de Tomada de Decisão.....	29
2.2.4. Técnicas de <i>Data Mining</i> dos SDD.....	31
2.2.5. Sistemas Inteligentes dos SDD.....	32
2.3. <i>Cloud Computing</i>	34
2.3.1. Principais Características da <i>Cloud Computing</i>	34
2.3.2. Modelos de Implantação na <i>Cloud Computing</i>	35
2.3.3. Principais Elementos da <i>Cloud Computing</i>	36
2.3.4. Arquitetura da <i>Cloud Computing Mobile</i>	37
2.3.5. Tipos de <i>Cloud Computing</i>	38
2.4. Aplicações <i>Cross-Platform</i>	40
2.4.1. Principais Objetivos das Aplicações <i>Cross-Platform</i>	40
2.4.2. Processo de Desenvolvimento de Aplicações Móveis.....	41
2.4.3. Estratégias para o Desenvolvimento de Aplicações Móveis.....	41
2.4.4. Ferramentas de Desenvolvimento <i>Cross-Platform</i>	44
2.4.5. Tecnologias para suporte ao desenvolvimento de sistemas híbridos <i>responsive</i>	48
2.5. Engenharia <i>Web</i>	49

3. Arquitetura da <i>Framework</i>	52
3.1. <i>Framework SaaS Multifuncional (FSM)</i>	52
3.2. Pacotes Principais	52
3.3. Componentes FSM	53
3.4. Tecnologia de Desenvolvimento	54
4. Sistema SaaS - SCP	55
4.1. Diagrama de Classes	55
4.2. Casos de Uso.....	57
4.3. Requisitos funcionais e não funcionais.....	58
5. Esquema Navegacional	60
5.1. <i>Mockup</i> da Aplicação.....	60
5.2. Suporte <i>Cross-Platform</i>	65
6. Conclusões.....	66
Bibliografia.....	68
Anexo	70
Apêndice.....	74
Apêndice A - Atividade do Jogo <i>Online</i> em Portugal	74
Apêndice B - Demonstração de termos técnicos sobre as apostas desportivas	80
Apêndice C - Exemplo prático da época 2018/2019 da Liga NOS	85

Índice de Figuras

Figura 1: Classificação da GC. Adaptado de AF Ragab, M., & Arisha, A. (2013)	19
Figura 2: Modelo SECI. Adaptado de AF Ragab, M., & Arisha, A. (2013).....	20
Figura 3: GC nas Organizações. Adaptado de King, W. R. (2009)	22
Figura 4: Modelo de Processos na GC. Adaptado de King, W. R. (2009).....	22
Figura 5: Framework da GC no processo de inovação. Adaptado de Wang, G. F., Tian, X. T., Geng, J. H., & Guo, B. (2013)	23
Figura 6: Aquisição estratégica de conhecimento como elemento-chave na criação de desempenho superior. Adaptado de Alyoubi, B. A. (2015)	26
Figura 7: Processo para a Tomada de Decisão. Adaptado de Felsberger, A., Oberegger, B., & Reiner, G. (2016)	29
Figura 8: Estrutura de um IDSS. Adaptado de Phillips-Wren, G., Mora, M., Forgionne, G. A., & Gupta, J. N. (2009)	33
Figura 9: Modelos de Cloud Computing. Adaptado de Ambavane, S. A., Pawar, A. S., Verma, V. H., & Marathe, P. (2018).	36
Figura 10: Arquitetura da Cloud Computing. Adaptado de Padhy, R. P., Patra, M. R., & Satapathy, S. C. (2011).....	39
Figura 11: Aplicação Web utilizando o PhoneGap. Adaptado de (Palmieri, M., Singh, I., & Cicchetti, A., 2012)	45
Figura 12: Código fonte do PhoneGap. Adaptado de (Palmieri, M., Singh, I., & Cicchetti, A., 2012)	46
Figura 13: Exemplo de uma aplicação desenvolvida utilizando o DragonRad. Adaptado de (Palmieri, M., Singh, I., & Cicchetti, A., 2012).....	48
Figura 14: Pacotes da FSM.....	52
Figura 15: Arquitetura de Componentes FSM	53
Figura 16: Módulos da Framework de Desenvolvimento	54
Figura 17: Principais classes do sistema SaaS - SCP, multiplicidades e respetivos atributos	56
Figura 18: Principais Caso de Uso do Sistema SaaS - SCP	57
Figura 19: Opções iniciais	60
Figura 20: Criação de conta.....	60
Figura 21: Login	61
Figura 22: Escolha da Aplicação	61
Figura 23: Escolha de jogos na Aplicação Pré Live.....	62
Figura 24: Estatística de um jogo	62
Figura 25: Demonstração da Aplicação Live	63
Figura 26: Exemplo probabilístico para um determinado jogo	64
Figura 27: Quadro Resumo da Atividade do Jogo Online (Fonte: SRIJ).....	74
Figura 28: Receita Bruta dos Jogos e Apostas Online, Evolução 1º T 2017 - 2º T 2019 (M €) (Fonte: SRIJ)	75
Figura 29: Receita Bruta: Apostas Desportivas à Cota, Evolução 2º T 2018 - 2º T 2019 (M €) (Fonte: SRIJ)	75

Figura 30: Volume de Apostas Desportivas à Cota, Evolução 2º T 2018 - 2º T 2019 (M €) (Fonte: SRIJ)	76
Figura 31: Apostas Desportivas à Cota por Modalidade (2º T 2019) (Fonte: SRIJ).....	76
Figura 32: Apostas Desportivas à Cota por Competição (2º T 2019) (Fonte: SRIJ).....	77
Figura 33: Jogadores - Evolução do número de novos registos, 2º T 2018 - 2º T 2019 (Fonte: SRIJ)	77
Figura 34: Jogadores - Distribuição por Estrutura Etária (Fonte: SRIJ)	78
Figura 35: Jogadores - Prática de Jogos por Categoria (Fonte: SRIJ).....	78
Figura 36: Operadores de Jogo Online Ilegais (Fonte: SRIJ).....	79
Figura 37: Fórmula da Distribuição de Poisson	80
Figura 38: Probabilidade de cada equipa fazer um número de golos	82
Figura 39: Classificação final da Época 2018/2019 da Liga NOS	86
Figura 40: Classificação dos jogos em casa da Época 2018/2019 da Liga NOS	86
Figura 41: Classificação dos jogos fora de casa da Época 2018/2019 da Liga NOS	86
Figura 42: Probabilidade de haver pelo menos um golo	88
Figura 43: Probabilidade de haver pelo menos dois golos	88
Figura 44: Probabilidade de haver pelo menos três golos	88

Índice de Tabelas

Tabela 1: Elementos da infraestrutura da Gestão de Conhecimento. Adaptado de (Zaied, A. N. H., Hussein, G. S., & Hassan, M. M., 2012)	18
Tabela 2: Elementos do processo na Gestão de Conhecimento. Adaptado de (Zaied, A. N. H., Hussein, G. S., & Hassan, M. M., 2012)	18
Tabela 3: Características essenciais do sistema de Cloud Computing. Adaptado de (Jain, A., & Kumar, R., 2014)	34
Tabela 4: Desafios da Cloud Computing. Adaptado de (Jain, A., & Kumar, R., 2014)	36
Tabela 5: Requisitos necessários para produzir e manter aplicações móveis de alto desempenho. Adaptado de (Xanthopoulos, S., & Xinogalos, S., 2013).....	40
Tabela 6: Descrição dos artefactos no desenvolvimento de aplicações Web. Adaptado de (Bolchini, D., 2000).....	51
Tabela 7: Requisitos funcionais do Sistema SCP	58
Tabela 8: Requisitos não funcionais do Sistema SCP	59
Tabela 9: Definição de Tipos de Aposta da Aplicação Pré Live.....	63

1. Introdução

Neste capítulo, é introduzido o contexto e problema, que visa a construção de um sistema aplicacional direcionado para as apostas desportivas, bem como os principais objetivos e metodologia de investigação.

1.1. Contexto

Com a legalização das apostas em Portugal no dia 29 de abril de 2015, a cultura de apostar tem vindo a crescer a um grande ritmo e, cada vez existem mais casas de apostas a operar em Portugal. Uma forma de ver este crescimento exponencial, é verificar que no início desta época – 2019/2020 – as quatro maiores equipas portuguesas assinaram acordos com casas de apostas para serem como patrocinadoras dos clubes e, o Sporting de Braga, foi a primeira equipa Portuguesa desde que existe legislação em Portugal, a ter como principal patrocinadora nas camisolas de futebol profissional, uma empresa de apostas. Também é cada vez mais visível nos principais canais de televisão, junto às balizas dos estádios, nas conferências de imprensa, nos *websites* dos clubes, etc.

Como o conhecimento os apostadores é cada vez maior, estes procuram fazer as suas apostas o mais suportadas possível, mas existe uma lacuna evidente quer em Portugal quer no resto do mundo de não existirem suficientes ferramentas para uma melhor tomada de decisão neste domínio. Com base nesta lacuna, surgiu uma pergunta: “Qual o motivo para existirem cada vez mais novas casas de apostas no mercado?” Para perceber o porquê desta lacuna, analisaram-se como funcionam as ferramentas das casas de apostas para atribuir *odds* e implementar, dentro dos recursos possíveis, sem ter que investir quantidades de dinheiro significativas, e desenvolver uma aplicação que recomende a probabilidade justa para um jogo antes de começar e durante o jogo, ajudando assim, os apostadores a saberem qual a *odd* mínima que devem apostar num jogo de futebol. Salienta-se que uma *odd* é a probabilidade de algo acontecer num determinado evento, representada entre 1 (100%) e 1000 (0%).

1.2. Problema

No contexto referido, o principal problema consiste em: definir uma *Framework* SaaS Multifuncional, para auxiliar a tomada de decisão de um apostador desportivo, com base em dados estatísticos que suporte uma aplicação “Pré Live” específica para a obtenção de probabilidades antes de começar o jogo, e outra aplicação “Live” para determinar probabilidades com base na estatística do próprio jogo.

Salienta-se que não existe nenhuma solução no mercado português com as características propostas para a aplicação “Live”.

1.3. Objetivos de Investigação

Com base no problema, definem-se os principais objetivos de investigação:

- Efetuar a revisão da literatura com foco nos tópicos: *Gestão de Conhecimento, Sistemas de Suporte à Decisão, Cloud Computing, Aplicações Cross-Platform e Engenharia Web*;
- Definir uma *Framework SaaS Multifuncional* para instanciação de aplicações na *cloud*, seja em contexto de apostas desportivas, seja noutros contextos;
- Definir componentes modulares para suporte diversos modelos de negócio (incluindo as apostas desportivas);
- Propor tecnologias para desenvolvimento de um futuro protótipo aplicacional;
- Definir as relações estáticas entre as principais classes do sistema;
- Especificar casos de uso no contexto específico do problema a resolver;
- Enumerar os principais requisitos funcionais e não-funcionais;
- Propor um esquema navegacional para um futuro protótipo instanciado em paradigma *cross-platform*;
- Demonstrar cenários em que as heurísticas específicas para apostas são exemplificadas.

1.4. Método de Investigação

O método de investigação adotado para a realização deste trabalho foi o *DSR - Design Science Research*. Este método baseia-se em ações concretas, destinadas a resolver uma situação problema imediato com produção tendencial de artefactos tecnológicos. O processo de investigação é constituído por uma fase de observação e outra de intervenção. Existem cinco ciclos no DSR:

- **Diagnóstico:** Corresponde à identificação dos principais problemas que são as causas subjacentes do desejo da organização para a mudança. É holístico, não envolve de redução ou simplificação. Permite desenvolver uma hipótese de trabalho sobre a natureza da organização e a área do Problema;
- **Planeamento da Ação:** Especifica as ações organizacionais que devem melhorar o problema. A descoberta das ações previstas é guiada pelo quadro teórico, o que indica tanto um estado futuro desejado, e as mudanças que permitirão atingir tal estado. O plano estabelece a meta e a abordagem para a mudança;
- **Empreender ações:** Implementa a ação planeada. Os investigadores e profissionais colaboram na intervenção ativa na organização cliente, fazendo com que certas alterações sejam implementadas;
- **Avaliação:** Após a conclusão das ações, os investigadores e profissionais avaliam os resultados. A avaliação inclui determinar se os efeitos teóricos da ação foram realizados, e se estes efeitos resolveram os problemas.

Quando a mudança foi bem-sucedida, a avaliação deve criticamente questionar se a ação empreendida, entre as ações organizacionais rotineiras e não-rotineiras, foi a real causa do sucesso. Quando a mudança não foi bem-sucedida, devem ser estabelecidos alguns quadros, para a próxima iteração do ciclo de action research;

- **Aprendizagem:** A aprendizagem é geralmente é um processo contínuo.

1.5. Estrutura do Documento

No primeiro capítulo, é apresentado o contexto, o problema e principais objetivos de investigação. De igual modo, é apresentado o método de investigação escolhido.

No segundo capítulo, apresenta-se a revisão da literatura por áreas de conhecimento relevantes para o desenvolvimento do trabalho. Em concreto: Gestão de Conhecimento, Sistemas de Suporte à Decisão, *Cloud Computing*, Aplicações *Cross-Platform* e Engenharia *Web*.

No terceiro capítulo apresenta-se a arquitetura da *Framework* proposta, pacotes principais, diagrama de componentes e tecnologias de desenvolvimento.

No quarto capítulo apresenta-se o sistema SaaS - SCP. Em concreto através do diagrama das principais classes do sistema, casos de uso e análise preliminar dos principais requisitos funcionais e não funcionais.

O quinto capítulo é dedicado à apresentação do esquema navegacional principal para suporte à realização do protótipo a desenvolver.

No sexto capítulo apresentam-se as conclusões e trabalho futuro.

Anexa-se, o artigo científico referente à infraestrutura tecnológica que suporta o sistema proposto e principais características funcionais. Deste modo, as ideias centrais e opções de implementação foram validadas, aceites e publicadas pela comunidade científica, em concreto, na *CISTI'2019 - 14th Iberian Conference on Information Systems and Technologies*.

O presente documento inclui ainda três apêndices relevantes:

- 1) Atividade do jogo *online* em Portugal do segundo trimestre de 2019, onde é descrita a análise global, a atividade por categoria de jogos e apostas *online*, qual o número de jogadores registados e monitorização de operadores ilegais de jogo *online* (que continua a ser um problema muito grave no nosso país devido à legislação atual).
- 2) Designação de termos técnicos sobre as apostas desportivas. E.g como como atribuir uma probabilidade a um resultado no futebol, *odds* de valor e golos esperados.
- 3) Um Exemplo prático da época 2018/2019 da Liga NOS que visa demonstrar heurísticas a realizar.

2. Revisão da Literatura

Neste capítulo é apresentada a revisão da literatura e conceitos essenciais para a leitura e compreensão do trabalho.

2.1. Gestão de Conhecimento

A Gestão de Conhecimento (GC) é o planeamento, organização, motivação e controlo de pessoas, processos e sistemas numa organização, tendo como objetivo garantir que os seus ativos relacionados ao conhecimento sejam aprimorados e efetivamente empregados. Os ativos relacionados ao conhecimento incluem conhecimento na forma de documentos impressos, como patentes/manuais, conhecimento armazenado em repositórios eletrónicos, bases de dados de melhores práticas, conhecimento dos colaboradores sobre a melhor maneira de realizarem as suas tarefas, conhecimento mantido por equipas que trabalham com problemas e conhecimento focado nos produtos, processos e relacionamentos da organização, entre muito outros. Os processos da GC envolvem a adquirir, criar, aprimorar, armazenar, transferir, partilhar e utilizar o conhecimento. A função da GC numa organização é operar processos, desenvolver metodologias e sistemas para apoiar e motivar os colaboradores a participar deles. Os objetivos da GC são alavancar e melhorar os ativos de conhecimento da organização, para efetivar melhores práticas de conhecimento, comportamentos organizacionais aprimorados, melhores decisões e desempenho organizacional (Zaied, A. N. H., Hussein, G. S., & Hassan, M. M., 2012).

No contexto do trabalho, é relevante manter presente que o protótipo a realizar se enquadra de igual modo nos requisitos dos sistemas de gestão de conhecimento.

2.1.1. Sistemas da Gestão de Conhecimento

Os Knowledge Management Systems (KMS) envolvem a participação do utilizador na fase de operação. Por exemplo, quando uma base de dados de vendas é projetada, os utilizadores devem decidir sobre o seu conteúdo e estrutura e, na sua fase operacional, funciona automaticamente. Quando um repositório de conhecimento com “lições aprendidas” é criado, cada unidade de conhecimento submetida para inclusão é única e deve ser avaliada quanto à sua relevância e importância. As ideias centrais da GC são o mecanismo para a organização desenvolver o seu conhecimento, estimular a criação de conhecimento dentro da organização, bem como a partilha e a proteção do mesmo. A Tabela 1 resume estes elementos (Zaied, A. N. H., Hussein, G. S., & Hassan, M. M., 2012).

Elementos	Definição
Tecnologia	Sistemas técnicos dentro de uma organização, que determinam como o conhecimento existe por toda a organização e como o conhecimento é acedido pelos colaboradores.
Estrutura	Extensão da disposição estrutural de uma organização em incentivar atividades relacionadas ao conhecimento.
Cultura	Conjunto de valores, normas e crenças partilhadas, principalmente implícitas, que os membros de uma organização possuem.
Recursos Humanos	Descrevem até que ponto os colaboradores se especializam num domínio específico e, demonstram a capacidade de aplicar o conhecimento para interagir com outros colaboradores.

Tabela 1: Elementos da infraestrutura da Gestão de Conhecimento. Adaptado de (Zaied, A. N. H., Hussein, G. S., & Hassan, M. M., 2012)

Os processos da GC são definidos como o grau em que a organização cria, partilha e utiliza recursos de conhecimento além dos limites funcionais. Segundo Zaied, A. N. H., Hussein, G. S., & Hassan, M. M. (2012), as capacidades do processo da GC são: aquisições, conversões, aplicações de armazenamento e proteções, conforme apresentado na Tabela 2.

Elementos	Definição
Aquisição	Abrange as atividades de acessibilidade, recolha e aplicação do conhecimento adquirido.
Conversão	Converte o conhecimento adquirido de fontes externas e internas em formas úteis e aplicáveis, para melhorar a produtividade e as operações comerciais.
Aplicações	É o processo da utilização real do conhecimento. A aplicação do conhecimento permite que as organizações traduzam continuamente o seu conhecimento organizacional em produtos incorporados.
Armazenamento	Tem como objetivo manter o conhecimento dentro da organização e inclui quer recursos físicos como recursos não físicos.
Proteção	Proteger o conhecimento e mantê-lo seguro, sendo apenas acedido pelos colaboradores autorizados.

Tabela 2: Elementos do processo na Gestão de Conhecimento. Adaptado de (Zaied, A. N. H., Hussein, G. S., & Hassan, M. M., 2012)

2.1.2. Classificação da Gestão de Conhecimento

Segundo AF Ragab, M., & Arisha, A. (2013), a taxonomia da GC é classificada em cinco categorias distintas: Ontologia do Conhecimento e GC, Sistemas de Gestão de Conhecimento, Papel da Tecnologia da Informação, Questões de Gestão e Sociais e Medição do Conhecimento. Estas categorias são detalhadas na Figura 1.

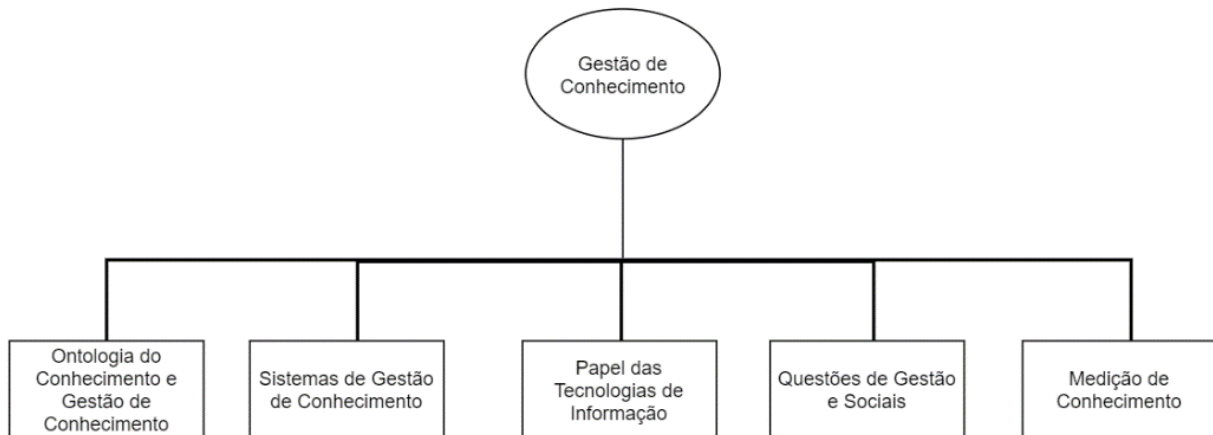


Figura 1: Classificação da GC. Adaptado de AF Ragab, M., & Arisha, A. (2013)

I. Ontologia do Conhecimento e GC

A definição mais comum de conhecimento é a camada superior numa hierarquia de três níveis. Começando pelos "dados" que, quando processados, produzem "informações" e, quando combinadas com experiência e julgamento, e utilizados na tomada de decisões, tornam-se em "conhecimento". Existe então, dois tipos de conhecimento, designados por tácito e explícito. O conhecimento tácito é aquele que pode ser codificado e armazenado em vários formatos - como manuais ou bases de dados - e, portanto, podem ser transferidos sem perdas. O conhecimento explícito, por outro lado, reside no julgamento e nas experiências de um indivíduo e, por isso, não pode ser articulado nem armazenado. Estas características estão detalhadas na Figura 2 (AF Ragab, M., & Arisha, A., 2013).

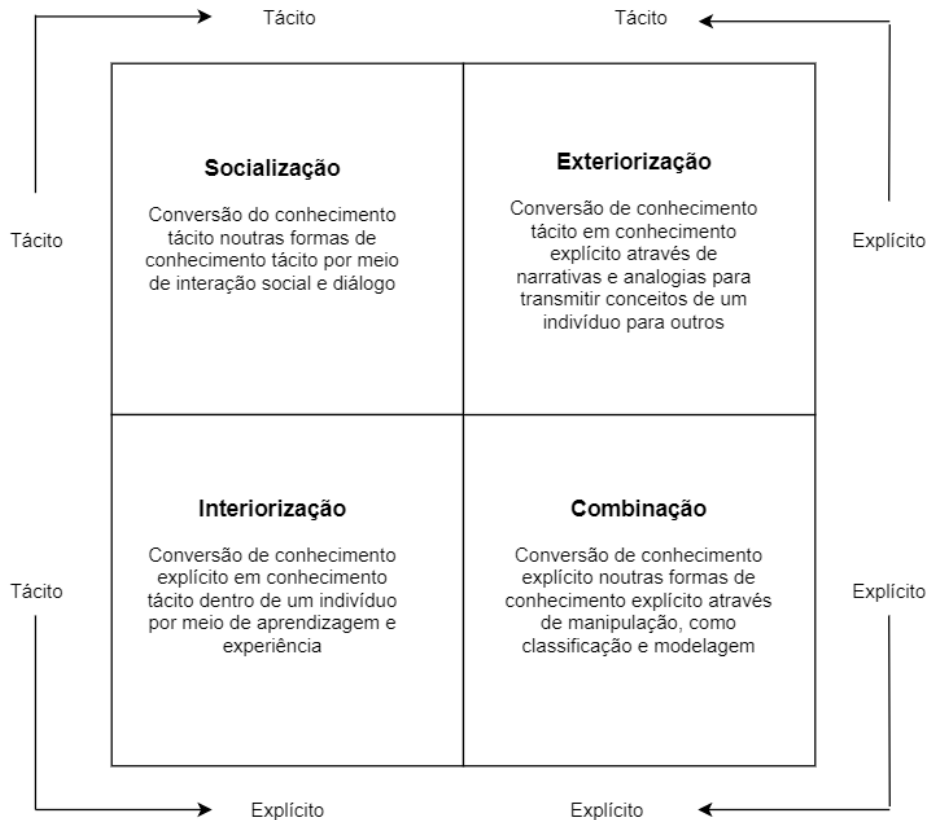


Figura 2: Modelo SECI. Adaptado de AF Ragab, M., & Arisha, A. (2013)

II. Sistemas de Gestão de Conhecimento

Como referido anteriormente, um KMS é um sistema de gestão, técnico, organizacional e estruturado para apoiar a implementação da GC dentro de uma organização. Existem três abordagens para projetar um KMS: codificação, personalização e localizador de pessoas. A abordagem de codificação, tem como objetivo capturar e armazenar conhecimento através de repositórios eletrónicos. Esta abordagem adota uma estratégia de pessoas para documentos, procurando reutilizar investindo na construção de bases de dados robustas. A personalização foca-se na transferência de conhecimento por meio de atividades de interação social presenciais, praticando e contando histórias. Esta abordagem visa partilhar conhecimento e fomentar a inovação e, portanto, é vista como a mais adequada para a disseminação do conhecimento explícito (AF Ragab, M., & Arisha, A., 2013).

III. Papel da Tecnologia da Informação

O papel da tecnologia da informação contribui para o desenvolvimento de novos algoritmos com o objetivo de melhorar o desempenho e a facilidade na utilização das soluções atuais da GC, com recurso a diversas técnicas (e.g. *data mining*, IA, sistemas especialistas, tecnologias de bases de dados, pesquisa técnicas e modelagem). Utilizando estas técnicas, vários tipos de *software* foram produzidos, cada um rotulado como uma solução da GC (AF Ragab, M., & Arisha, A., 2013).

IV. Questões de Gestão e Sociais em GC

Um problema comum é a tendência de os colaboradores impedirem as iniciativas da GC, que decorrem da sua resistência em partilhar o seu conhecimento com outras pessoas da organização. A principal razão para o conhecimento acumular nas organizações é o medo das pessoas de perderem o seu "valor único", que reduziria as suas probabilidades de serem promovidas e aumentaria as de outras pessoas com quem partilharam o seu conhecimento. Essa resistência é amplificada pelo fato de que a acumulação de conhecimento não é proibida, enquanto a partilha de conhecimento geralmente não é reconhecido nem recompensado nas organizações. Quem está disposto a partilhar o seu conhecimento pode ser inibido pela falta de tempo necessário para colocá-lo numa forma adequada para partilhar, desconhecer o conhecimento necessita ser partilhado, medo de publicar algo confidencial e a falta de cultura e/ou estrutura organizacional (AF Ragab, M., & Arisha, A., 2013).

V. Medição do Conhecimento

A medição do conhecimento é subjetiva e difícil. A natureza fluida e intangível do conhecimento faz com que a sua medição seja uma tarefa complexa. As discussões sobre medição do conhecimento são frequentemente associadas ao conceito de Capital Intelectual, geralmente definido como conhecimento, informação, propriedade intelectual e experiência que pode ser utilizada para desenvolver valor. O Capital Intelectual pode ser visto como o "stock" de conhecimento da organização a qualquer momento específico, compreendendo o conhecimento que foi adquirido e formalizado para ser utilizado para criar valor e, assim, obter vantagem competitiva (AF Ragab, M., & Arisha, A., 2013).

Assim, as organizações implementam processos para capturar e disseminar fluxos de conhecimento com o objetivo de acumular Capital Intelectual. Os fatores determinantes da medição de Capital Intelectual são vistos de duas perspetivas organizacionais: interna e externa. De uma perspetiva interna, por vezes a administração pode não conhecer o valor do seu próprio Capital Intelectual, nem onde está dentro da sua organização, apesar de ser a sua principal fonte de vantagem competitiva. De uma perspetiva externa, existe uma visão ampla de que o valor de uma organização só pode ser avaliada se os ativos intangíveis forem levados em consideração, devido às lacunas entre os valores contabilísticos e valor de mercado das organizações, onde a proporção entre as últimas e as anteriores se multiplicou na última década (AF Ragab, M., & Arisha, A., 2013).

2.1.3. Gestão de Conhecimento nas Organizações

Segundo King, W. R. (2009), os processos da GC melhoram diretamente os processos organizacionais, como inovação, tomada de decisão colaborativa, aprendizagem individual e coletivo. Estes processos organizacionais aprimorados produzem resultados intermediários, como melhores decisões, comportamentos organizacionais, produtos, serviços e relacionamentos. Estes, por sua vez, levam a um melhor desempenho dentro das organizações, como demonstra a Figura 3.

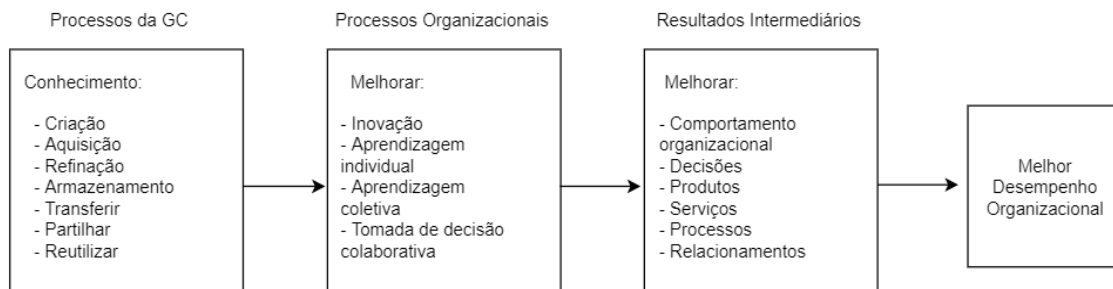


Figura 3: GC nas Organizações. Adaptado de King, W. R. (2009)

A Figura 4 apresenta um Modelo de Ciclo de Processos da GC. Mostra que o início do ciclo da GC envolve a criação ou a aquisição de conhecimento por uma organização. A criação de conhecimento envolve o desenvolvimento de um novo conhecimento ou substituição do conhecimento existente por novos conteúdos. O foco disso geralmente está na criação de conhecimento dentro dos limites da organização, ou em conjunto com parceiros (King, W. R., 2009).

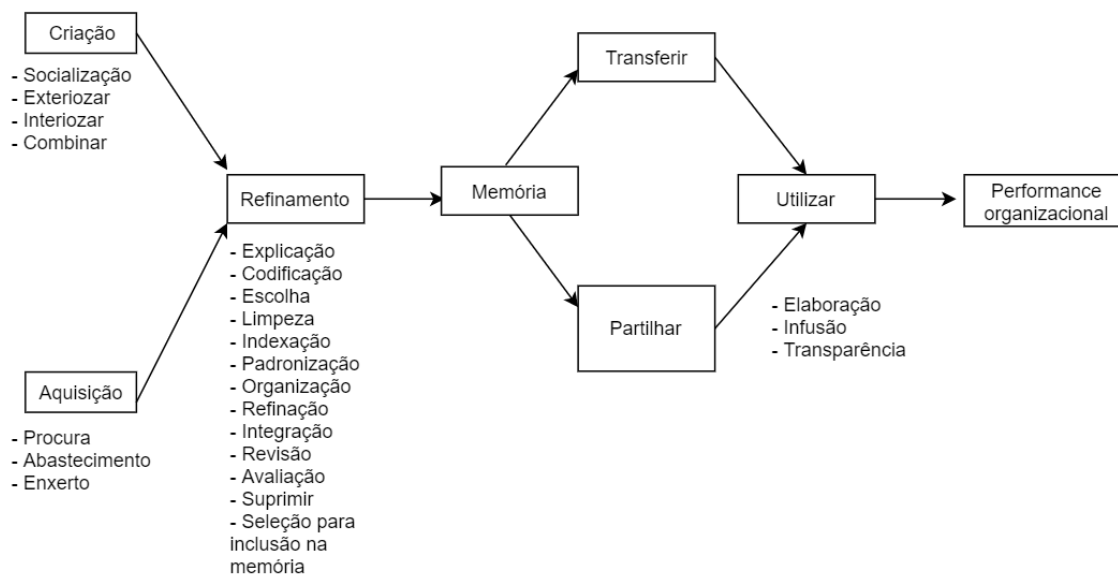


Figura 4: Modelo de Processos na GC. Adaptado de King, W. R. (2009)

2.1.4. Estrutura da Gestão de Conhecimento da Inovação de Processos

A estrutura da gestão de conhecimento da inovação de processos abrange a aquisição de conhecimento, representação de conhecimento, organização do conhecimento, armazenamento de conhecimento, partilha de conhecimento, aplicação de conhecimento, acumulação de conhecimento e outras atividades relevantes. A GC em inovação de processos é um meio da GC, virado especificamente para a inovação de processos, portanto, deve estar em conformidade com as metodologias gerais da GC (Wang, G. F., Tian, X. T., Geng, J. H., & Guo, B., 2013).

À luz das características das atividades de inovação de processos, é proposta uma nova estrutura para a GC em inovação de processos, como exemplificado na Figura 5. A estrutura possui quatro partes principais, em concreto: aquisição de conhecimento, organização do conhecimento, aplicação do conhecimento e acumulação de conhecimento.

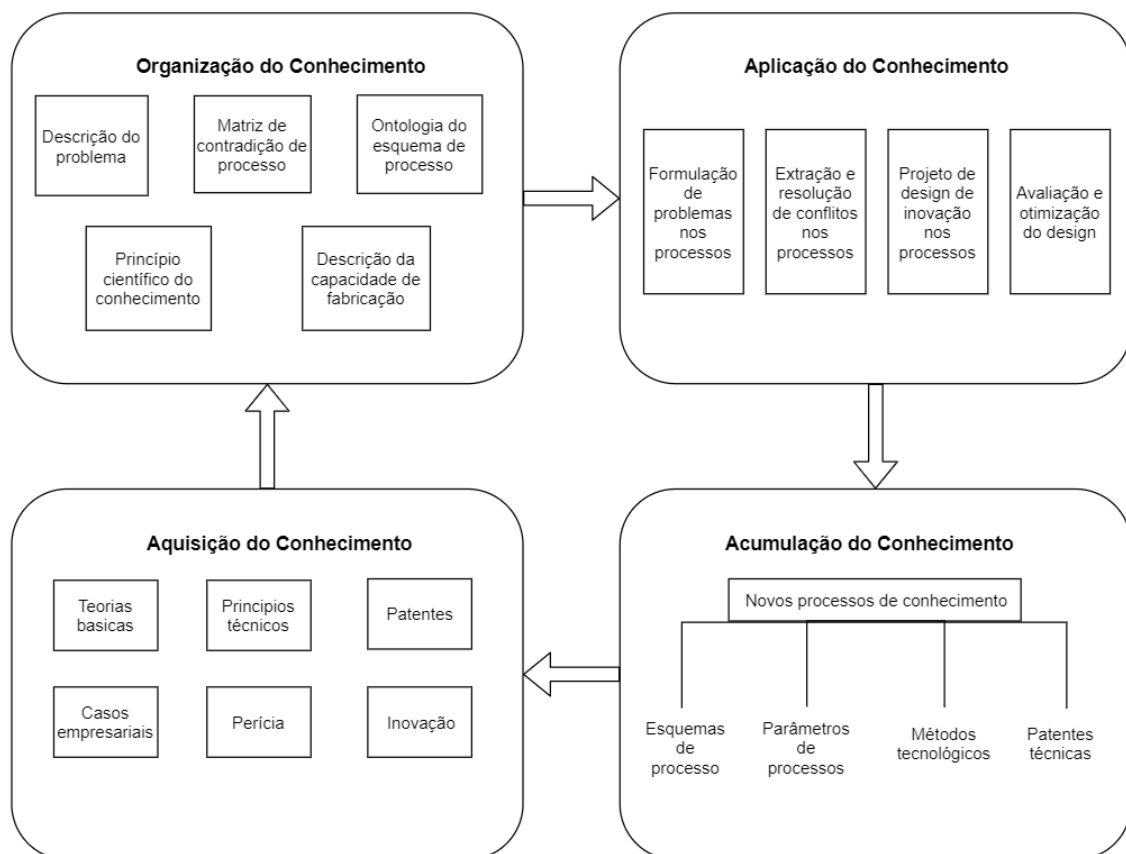


Figura 5: Framework da GC no processo de inovação. Adaptado de Wang, G. F., Tian, X. T., Geng, J. H., & Guo, B. (2013)

A. Aquisição de conhecimento

A inovação de processos pressupõe a síntese do conhecimento multidisciplinar, do conhecimento profissional e do conhecimento das organizações. As seguintes fontes de conhecimento sobre inovação exigem a aplicação completa no design da inovação de processos (Wang, G. F., Tian, X. T., Geng, J. H., & Guo, B., 2013):

- a) **Teorias básicas:** Regras gerais no campo das ciências naturais, que são de importância direta para o processo de inovação, por exemplo, as teorias e os princípios da Física, Química, Geometria, Materiais e Biologia;
- b) **Princípios técnicos:** Métodos de processo e mecanismos de fabricação que podem orientar a fabricação do produto. Design de inovação de processo é uma atividade de investigação, que deve aderir aos princípios técnicos correspondentes para obter uma inovação bem-sucedida;
- c) **Patentes técnicas:** As patentes técnicas como transportadoras de conhecimento apresentam criatividade e aplicabilidade notáveis, que refletem os avanços resultados da investigação, com novos processos e novos métodos;
- d) **Casos organizacionais:** Estratégias e soluções efetivas acumuladas pelas organizações para resolver problemas concretos de fabrico, como abordagens de projetos, projetos inovadores etc. Os casos corporativos incorporam os recursos distintos da organização, que também fornecem referência direta para o projeto;
- e) **Especialização:** A experiência refere-se ao *know-how* e às técnicas de fabrico que alimentam a resolução de conflitos de processos e o design do esquema de inovação de processos com múltiplas linhas de pensamento;
- f) **Teorias e métodos de inovação:** Meios ou metodologias de exploração dos recursos disponíveis na invenção. As teorias e métodos de inovação podem ampliar os pensamentos da inovação de processos.

B. Organização de conhecimento

A organização do conhecimento e do *Data Knowledge of Process Innovation*, está de acordo com os processos de resolução de problemas. Além disso, este método determina a forma e a estrutura do conhecimento adquirido os recursos de conhecimento de inovação, que serão armazenados. Por exemplo, parâmetros de contradição e os princípios inovadores relevantes necessários para o estabelecimento de uma matriz de contradição de processo pode ser derivado da análise ou modelagem de dados de patentes técnicas e casos organizacionais, durante os quais o relacionamento de mapeamento entre processos contradição e princípios inovadores podem ser estabelecidos. Utilizando a rede de ontologia de processos, os princípios inovadores podem ser associados a patentes técnicas ou casos organizacionais que devem ser formalizados, de modo a formar a ontologia do esquema de processos (Wang, G. F., Tian, X. T., Geng, J. H., & Guo, B., 2013).

C. Aplicação de conhecimento

O conhecimento sobre inovação de processos fornece suporte para todos os processos de inovação de processos. O conhecimento necessário na solução de problemas pode ser extraído do módulo de *Data Knowledge of Process Innovation*, para atingir efetivamente a meta do estágio (Wang, G. F., Tian, X. T., Geng, J. H., & Guo, B., 2013).

D. Acumulação de conhecimento

Através da estrutura da GC, o conhecimento formalizado abstraído das fontes de conhecimento da inovação será armazenado na base de dados. Após as atividades de inovação de processo, será transformado em novos esquemas e novo conhecimento sobre os processos, que contêm novos esquemas de processos, novos métodos tecnológicos, etc. Ao adicionar novo conhecimento do processo à base de dados correspondente, o problema subsequente do processo será resolvido com mais facilidade e eficácia (Wang, G. F., Tian, X. T., Geng, J. H., & Guo, B., 2013).

2.2. Sistemas de Suporte à Decisão

Os Sistemas de Suporte à Decisão (SSD) – em inglês DSS – são ferramentas que auxiliam na tomada de uma ou mais decisões numa organização. A importância da GC também é reconhecida devido às suas contribuições na tomada de decisões nas organizações. Os SSD foram sinergizados com os sistemas da GC e, evoluíram do processamento de dados e MIS, para a sua forma atual, como auxílio dos SI, indispensável para a tomada de decisões. A aplicação mais comum da sinergia do SSD e GC pode ser encontrada em GSS. O grupo de suporte do GSS funciona como brainstorming, avaliação de ideias e facilidades de comunicação. A GC tem o potencial de permitir que as organizações obtenham vantagem competitiva por meio de um estudo detalhado dos fatores ambientais. Desta forma, o SSD é automaticamente visto como funções de suporte-chave, porque permite que operadores do conhecimento e quem toma decisões, que sejam bem informados, através do estudo eficaz de variáveis semiestruturadas e mal estruturadas nos fatores ambientais externos (Tripathi, K. P., 2011).

No contexto do trabalho, é relevante manter presente que o protótipo a realizar se identifica como uma ferramenta para suporte à decisão.

Segundo o mesmo autor, as principais características do SSD são:

1. Manipular grandes quantidades de dados, como pesquisas em bases de dados;
2. Obter e processar dados de diferentes fontes, incluindo dados internos e externos armazenados em sistemas e redes de mainframe;
3. Fornecer flexibilidade de relatórios e apresentação para atender às necessidades dos responsáveis pela tomada de decisão;
4. Ter orientação textual e gráfica, como gráficos, linhas de tendência, tabelas, etc.;
5. Realizar análises e comparações complexas e sofisticadas utilizando pacotes de *software* avançados;
6. Suportar a abordagens de otimização, satisfação e heurística, dando a quem toma decisões, uma grande flexibilidade na solução de problemas simples e complexos;
7. Realizar análises "what-if" e de procura de objetivos.

2.2.1. Estratégia nas organizações para a aplicação dos SDD

Segundo Alyoubi, B. A. (2015), a estratégia nas organizações é frequentemente definida como o plano abrangente, unificado e integrado que é desenvolvido para garantir que os objetivos da organização sejam alcançados. É impossível desenvolver o plano sem as informações corretas disponíveis para os planeadores estratégicos. O planeamento estratégico é, portanto, um processo intensivo em informações. Os dados referentes aos fatores internos e externos, relacionados à organização e ao meio ambiente. O processamento destes dados, são, portanto, vitais para a tomada de decisões estratégicas e como tal, a organização deve saber quais dados recolher, algo conhecido como aquisição estratégica de conhecimento, como se ilustra na Figura 6.

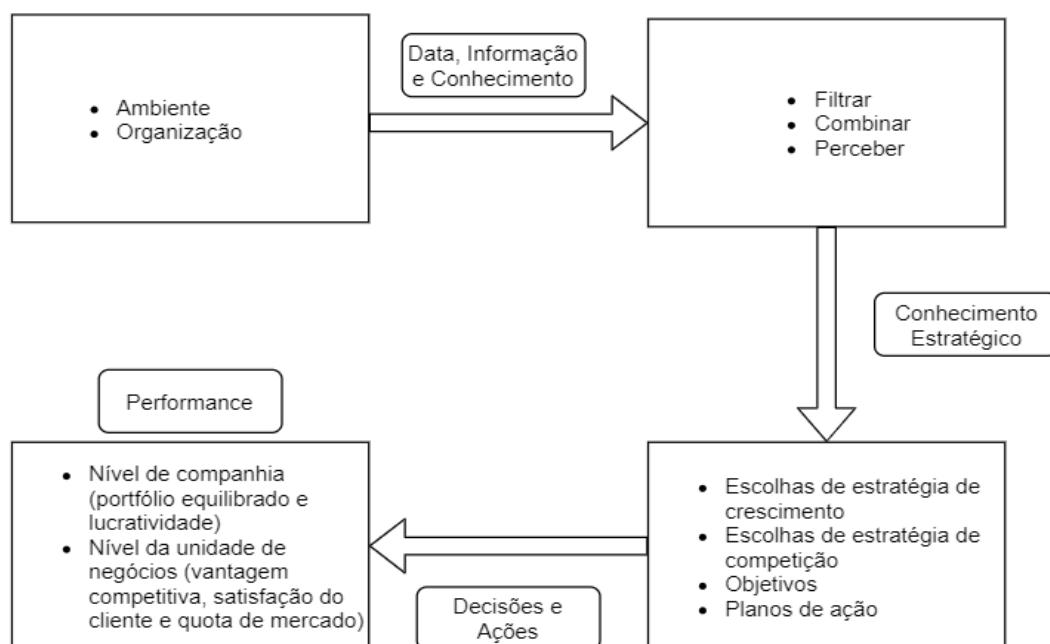


Figura 6: Aquisição estratégica de conhecimento como elemento-chave na criação de desempenho superior.
Adaptado de Alyoubi, B. A. (2015)

2.2.2. Tipos de SDD

1. SDD controlado por modelos

Um SDD orientado a modelos utiliza: modelos de análise, decisão financeira, otimização e simulação algébricas para suporte à decisão. É projetado para a manipulação de parâmetros do modelo pelo utilizador e para dar suporte para tomar uma decisão na análise de uma determinada situação. Não há necessidade de grandes bases de dados para este tipo de SDD, mas para análises específicas, podem ser extraídas. O principal componente na arquitetura é um ou mais modelos quantitativos que fornecem a funcionalidade. Ferramentas analíticas baseadas em modelos algébricos alocam um nível elementar de funcionalidade. Para desenvolver e construir modelos mais complexos de tomada de decisão, foram utilizados modelos de análise de decisão, otimização e programação matemática e técnicas de simulação.

Em geral, os modelos num SDD controlado por modelos representam uma simplificação da realidade. O objetivo de uma análise de decisão é descobrir a alternativa mais favorável sob a situação especificada e os modelos de otimização integrados a um SDD foram desenvolvidos em muitos ambientes. Especialmente na gestão de produção e operações e na gestão da cadeia de fornecimentos, a área dos modelos de otimização tornou-se um campo importante para um SDD (Felsberger, A., Oberegger, B., & Reiner, G., 2016).

2. SDD controlado por dados

Um SDD controlado por dados permite o acesso e a manipulação de dados estruturados, tal como lidar com séries temporais de dados internos e externos da organização, juntando dados em tempo real. Embora sistemas simples de arquivos acedidos por ferramentas de consulta e recuperação facilitem os sistemas de relatórios de gestão como *Data Warehouses*, que permitem a manipulação de dados por ferramentas computadorizadas, oferecem uma funcionalidade mais profunda. Outro exemplo de SDD orientado a dados é o BI, que oferecem o mais alto modo de operação e suporte a decisões. Os sistemas de inteligência de negócios ajudam as organizações a tomar decisões utilizando a tecnologia para relatórios e acesso a dados e práticas analíticas. Em geral, os sistemas de BI ajudam a formular decisões acionando, manipulando e analisando dados ou informações armazenadas em bases de dados históricos. O principal objetivo do BI e do *Analytic Systems* é aumentar a qualidade das informações disponíveis para a tomada de decisão, induzidos pela melhoria do processamento de dados. Neste caso, os principais requisitos de um SDD típico orientado a dados são o acesso a uma grande quantidade de dados e, ao mesmo tempo, alta qualidade dos dados subjacentes (Felsberger, A., Oberegger, B., & Reiner, G., 2016).

3. SDD controlado por comunicação

Normalmente, um SDD orientado à comunicação depende de tecnologias híbridas de rede e comunicação eletrônica, para conectar os responsáveis pela tomada de decisão e desenvolver um ambiente de partilha de recursos e informações, colaboração e comunicação entre um grupo dos responsáveis pela tomada de decisão. Uma grande subcategoria de tomada de decisão, é a tomada de decisão em grupo, posteriormente estendida aos chamados Sistemas de Suporte à Decisão em Grupo. Estes sistemas incluem técnicas de estruturação de problemas, como ferramentas de planeamento e modelagem. Existe o SDD orientado à comunicação, que é o *Collaborative Decision Support Systems* (CDSS), caracterizado como sistemas interativos baseados em computador, nos quais um grupo de responsáveis pela tomada de decisão, trabalha em equipa com o objetivo de encontrar soluções e alternativas para sistemas reconhecidos, novos e mal estruturados. problemas. Utilizando as TI, com ferramentas integradas de modelagem e análise, é possível um ambiente amigável para explorar soluções e tomar decisões (Felsberger, A., Oberegger, B., & Reiner, G., 2016).

4. SDD orientado a documentos

Após longos anos, a gestão de documentos está a tornar-se cada vez mais importante para as organizações. Consequentemente, uma quantidade de documentos, imagens, vídeos e documentos de hipertexto armazenados em algum tipo de sistema são cada vez maiores. Devido a isso, existem cada vez mais bases de dados de documentos grandes e tecnologias da *Internet* que aumentam ou desenvolvem o SDD orientado a documentos. Exemplos de ferramentas de tomada de decisão são os mecanismos de investigação vinculados ao SDD orientado a documentos, como o *Web Crawler*. Como os documentos não são padronizados, a recuperação de informações é utilizada para localizar documentos de um formulário não estruturado utilizando grandes recolhas de dados (Felsberger, A., Oberegger, B., & Reiner, G., 2016).

5. SDD baseado em conhecimento

Os SDD baseados em conhecimento têm a sua origem nos Sistemas Inteligentes de Suporte à Decisão ou, num sentido mais amplo, na IA. São sistemas de raciocínio baseados em computador, com uma distinção de tecnologias como IA, sistemas especializados em gestão, tecnologias *data mining* e mecanismos de comunicação integrados. O SDD inteligente é dividido em dois desenvolvimentos evolutivos. O primeiro tipo deste sistema é sobre sistemas especializados em regras amplamente utilizados para programação em sistemas de produção. Os sistemas especializados são baseados na utilização de heurísticas, que podem ser entendidos como estratégias que levam à solução correta para um problema.

O segundo tipo utiliza redes neuronais e algoritmos genéticos, que são semelhantes aos modelos de programação linear e, controlados por experimentos aleatórios, que são disponibilizados sem a utilização de funções de condicionamento físico. Os sistemas de algoritmos genéticos foram utilizados para problemas combinados e de otimização de parâmetros discretos (Felsberger, A., Oberegger, B., & Reiner, G., 2016).

2.2.3. Processo de Tomada de Decisão

Segundo Felsberger, A., Oberegger, B., & Reiner, G. (2016), o processo é determinado como uma identificação de alternativas como possíveis soluções para um próximo problema mal estruturado. Neste caso, o objetivo não é identificar o maior número possível de alternativas, mas o melhor que se ajusta a metas, objetivos e valores interpretados. O próprio processo de tomada de decisão passa por diversas fases e para tal, é necessário identificar um problema e determinar quem são os responsáveis pela tomada de decisão e, como partes do processo de decisão. Mostra um fluxo de processo de cima para baixo, mas pode sempre retornar uma etapa anterior a partir de qualquer ponto do processo. A Figura 7 ilustra o processo numa variação de *loop* fechado.

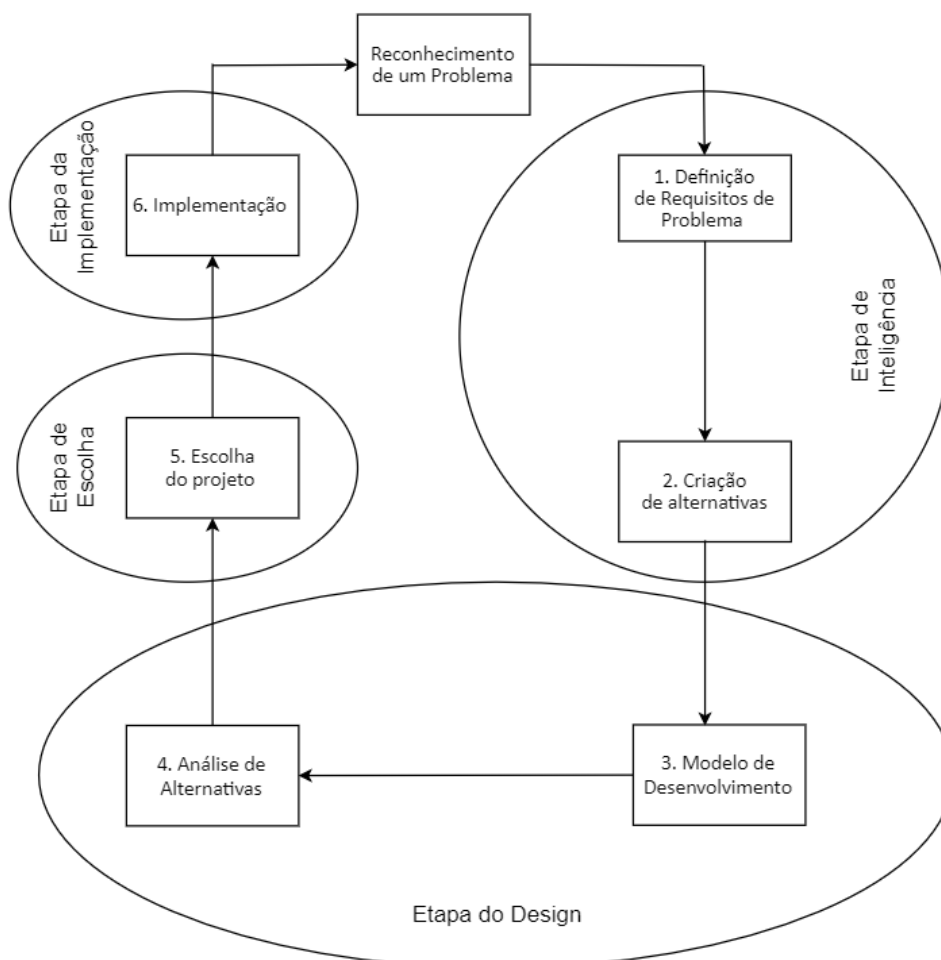


Figura 7: Processo para a Tomada de Decisão. Adaptado de Felsberger, A., Oberegger, B., & Reiner, G. (2016)

Pontos 1 e 2 - Definição de Requisitos de Problema e Criação de alternativas: Depois do problema ser reconhecido, o primeiro passo no processo de tomada de decisão é a definição do problema. O Ponto 1 visa expressar o problema numa declaração que descreve as condições iniciais e desejadas. A segunda parte desta etapa é determinar os requisitos para obter uma lista de exigências e objetivos absolutos.

No Ponto 2, a Criação de alternativas, lida com a criação de possíveis soluções alternativas para incertezas na tomada de decisões. Um aspeto importante é que as alternativas devem atender aos requisitos determinados. Esta fase do processo de tomada de decisão também é chamada de Fase de Inteligência, que consiste em encontrar, identificar e formular o problema ou situação. O resultado da fase de inteligência deve ser uma declaração de decisão.

Pontos 3 e 4 - Desenvolvimento do modelo e análise alternativa: Após a criação de alternativas e seleção de alternativas de ajuste, o próximo passo é o desenvolvimento do modelo. Modelos são desenvolvidos para analisar as várias alternativas e, simultaneamente, critérios devem ser definidos para medir a eficácia da alternativa.

O melhor modelo para comparar alternativas é aquele que atingirá as metas e critérios selecionados o mais rapidamente possível. Depois do modelo estar desenvolvido, o próximo passo é a análise alternativa que conclui a avaliação de alternativas contra critérios, a seleção de uma ferramenta de tomada de decisão, por exemplo. Finalmente, há também a validação de soluções em relação à declaração do problema, que visa garantir que realmente resolva o problema identificado.

Pontos 5 e 6 - Escolha do projeto e implementação da decisão: estas etapas são tomadas de decisão como base da Fase de escolha do modelo. Esta fase é sobre a seleção das alternativas desenvolvidas da fase de design.

O produto final do processo é uma decisão ou um modelo a implementar no ambiente de decisão, implementado na chamada fase de implementação. A solução deve satisfazer o estado desejado, atender aos requisitos e alcançar melhor os objetivos dentro dos valores do processo de decisão Felsberger, A., Oberegger, B., & Reiner, G. (2016).

2.2.4. Técnicas de *Data Mining* dos SDD

Quando existem grandes quantidades de dados, podem ocorrer problemas devido à análise dos dados e à extração de conhecimento necessária. Os dados são analisados através de um processo automatizado, conhecido como descoberta de conhecimento com recurso a técnicas de *Data Mining*.

No contexto do trabalho, é relevante manter presente que o protótipo a realizar se integra numa *Framework* que disponibiliza componentes para *Data Mining*.

Data Mining pode ser definido como um processo de exploração e análise de grandes quantidades de dados com um objetivo específico de descobrir padrões e regras significativamente importantes e, ajuda a encontrar conhecimento de dados não processados. Tem um papel importante em ajudar as organizações a perceber os seus clientes e o seu comportamento, tendo como objetivo encontrar padrões e relações ocultas que possam levar ao aumento da receita. Em relação aos estudos de *data mining*, existem dois tipos. O primeiro tipo é representado pelo teste de hipóteses, que pressupõe expor uma teoria sobre a relação entre ações e os seus resultados. O segundo tipo é representado pela descoberta do conhecimento (Bâra, A., & Lungu, I., 2012).

As técnicas de *Data Mining* residem no cálculo estatístico clássico, na administração da base de dados e na inteligência artificial. Não substituem as técnicas estatísticas tradicionais, mas funcionam como uma extensão das técnicas gráficas e estatísticas. O desenvolvimento de sistemas de suporte à decisão envolve tempo, custos elevados, esforços dos recursos humanos e, o sucesso do sistema pode ser afetado por muitos riscos, como: design do sistema, qualidade dos dados e obsolescência da tecnologia. O objetivo dos sistemas de apoio à decisão é auxiliar os a tomada de decisão sobre os benefícios do investimento, orçamentar fluxos de caixa e planeamento financeiro (Bâra, A., & Lungu, I., 2012).

Segundo o mesmo autor, existem várias séries de estágios de desenvolvimento para sistemas de inteligência de negócios: estudo de viabilidade, planeamento de projeto, análise, design, desenvolvimento e disponibilização para produção:

- i. **Estudo de viabilidade:** Identificar os requisitos e as oportunidades de negócios e propor soluções para melhorar o processo de tomada de decisão. Cada uma das soluções propostas deve ser justificada pelos custos e benefícios implícitos;
- ii. **Planeamento do projeto:** Avaliar as possibilidades de sustentabilidade do projeto, identificando os componentes de infraestrutura existentes e as necessidades futuras. O resultado destas atividades termina com o plano do projeto. Após a sua validação e aprovação, o início efetivo do projeto pode começar;
- iii. **Análise de requisitos de negócios:** Concentra-se no detalhe e na análise prioritária dos requisitos iniciais da equipa da gestão organizacional. Geralmente, os requisitos são identificados com base em entrevistas realizadas e pela equipa do projeto;

- iv. **Projeto do sistema:** De acordo com os requisitos do sistema, os dados necessários serão armazenados tanto num nível detalhado quanto num nível agregado, portanto, abordagens relacionais, orientadas a objetos ou multidimensionais podem ser adotadas. Durante esta subfase, o modelo de dados lógicos é refinado e detalhado e o modelo físico do novo sistema é desenvolvido para satisfazer os requisitos de relatório e análise;
- v. **Construir o sistema:** As tecnologias utilizadas para o desenvolvimento de sistemas de suporte à decisão fazem parte da categoria de tecnologias de inteligência de negócios e consistem em: tecnologias para organização de dados de *data warehouse*, algoritmos de *data mining* e tecnologias *web*;
- vi. **Implementação do sistema:** Representa o estágio em que o sistema está a ser entregue, e são realizadas sessões de esclarecimento sobre o que foi implementado, tal como é fornecido o suporte técnico necessário e os procedimentos de carregamento de dados é executado.

2.2.5. Sistemas Inteligentes dos SDD

Segundo Phillips-Wren, G., Mora, M., Forgionne, G. A., & Gupta, J. N. (2009), os Sistemas Inteligentes, utilizando bases de dados, contém os dados diretamente relevantes para o problema de decisão, incluindo os valores dos estados da natureza, cursos de ação e medidas de desempenho. A base de conhecimento contém conhecimento problemático, como orientação para a seleção de alternativas de decisão ou aconselhamento na interpretação de possíveis resultados. A base do modelo é um repositório para os modelos formais do problema de decisão e as abordagens (algoritmos e metodologias) para o desenvolvimento de resultados a partir dos modelos formais. Os responsáveis pela tomada de decisão utilizam computador e tecnologia da informação para processar as entradas em saídas relevantes para o problema. Este processamento envolve:

- Organização de entradas de problemas;
- Estruturar o modelo de decisão do problema de decisão;
- Utilizar o modelo de decisão para simular políticas e eventos;
- Encontrar a melhor solução para o problema.

Como a Figura 8 ilustra, um IDSS possui uma base de dados, uma base de conhecimento e base de modelos.

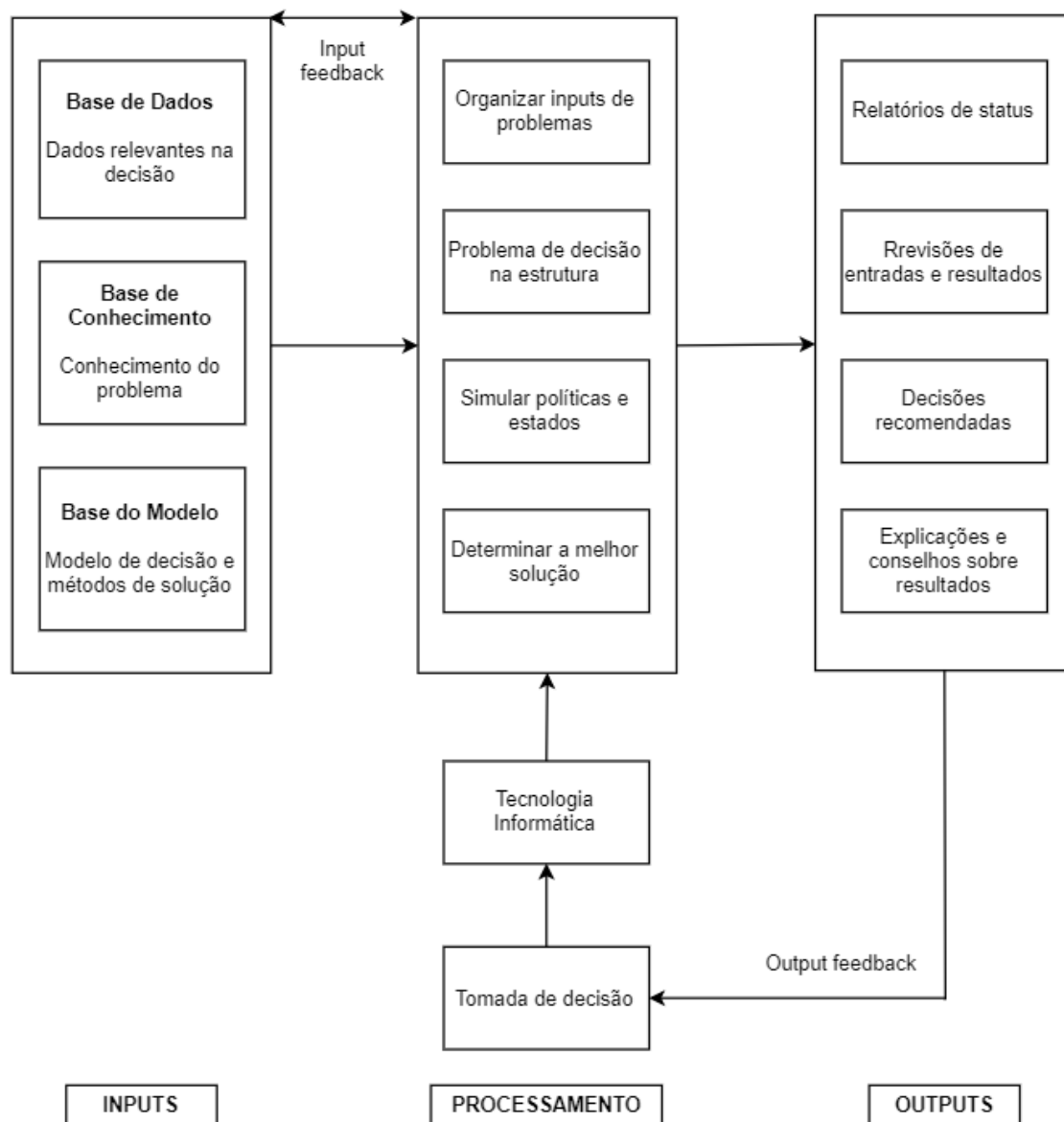


Figura 8: Estrutura de um IDSS. Adaptado de Phillips-Wren, G., Mora, M., Forgionne, G. A., & Gupta, J. N. (2009)

2.3. Cloud Computing

A *Cloud Computing* é definida como uma tecnologia onde utilizador final tem vantagens ao nível do investimento em recursos como hardware e software uma vez que o utilizador pode utilizar o serviço da *Cloud Computing* utilizando a *Internet*, a qualquer hora do dia e em qualquer lugar onde se encontre. Como referido, evita-se assim que várias organizações façam investimentos significativos em software, hardware, infraestruturas de recursos e plataformas (Jain, A., & Kumar, R., 2014).

2.3.1. Principais Características da *Cloud Computing*

Segundo Jain, A., & Kumar, R. (2014), as principais características da *Cloud Computing* têm como elementos os seguintes pontos da Tabela 3:

Elementos	Definição
Elasticidade e escalabilidade	Elasticidade é a capacidade de expandir ou reduzir um recurso de computação em tempo real, com base nos requisitos de computação do utilizador. A plataforma pode ser expandida ou reduzida dinamicamente de acordo com as circunstâncias da aplicação e o número de utilizadores.
Virtualização	Com a virtualização, um recurso físico pode ser criado para se parecer com vários recursos virtuais. A <i>Cloud Computing</i> fornece recursos aos utilizadores por meio da tecnologia de virtualização. Devido a este recurso, o provedor pode executar várias aplicações e sistemas operacionais numa única máquina física partilhando os recursos disponíveis.
Infraestrutura de larga escala	Existe mais de um milhão de servidores na plataforma de <i>Cloud Computing</i> do Google e mais de centenas de milhares de servidores na IBM, Amazon, Microsoft, Yahoo e outras plataformas de serviços na cloud.
Ubíquo	Os serviços fornecidos pela <i>Cloud Computing</i> não são personalizados para uma aplicação específica. Os utilizadores podem escolher diferentes aplicações de acordo com as suas necessidades. Utilizadores diferentes podem executar aplicações diferentes na mesma plataforma.
Preços baseados em serviços públicos	Quando os clientes utilizam uma infraestrutura da <i>cloud</i> que utiliza mais recursos, pagam por isso. O cliente está a pagar apenas o custo, que é reduzido, da infraestrutura.

Tabela 3: Características essenciais do sistema de *Cloud Computing*. Adaptado de (Jain, A., & Kumar, R., 2014)

2.3.2. Modelos de Implantação na *Cloud Computing*

Segundo Ambavane, S. A., Pawar, A. S., Verma, V. H., & Marathe, P. (2018), os quatro modelos de *Cloud Computing* são:

- **Cloud pública:** Um servidor terceirizado cria recursos, como aplicações e outros recursos de computação, para o público em geral utilizando a *Internet*. É oferecido num modelo de pagamento por utilização. O servidor de serviços é responsável pela configuração de hardware, software, aplicações e recursos de rede. Nuvens públicas não significam que os dados do utilizador sejam públicos. Em muitos casos, são necessários mecanismos de controlo de acesso antes que o utilizador possa utilizar os recursos disponibilizados;
- **Cloud privada:** Está disponível exclusivamente para uma única organização. Limita intencionalmente o acesso aos seus recursos para atender os utilizadores que pertencem à mesma organização que possui a cloud. A infraestrutura é gerida e operada apenas para uma organização. O principal objetivo é manter um nível consistente de controlo sobre segurança, privacidade e governança. Com uma *cloud* privada, os recursos de computação são agrupados e geridos internamente. Isso proporciona maior eficiência. Os recursos podem ser aplicados dinamicamente de acordo com a procura. Sendo privada, permite que a organização utilize os procedimentos de fluxo de trabalho e segurança;
- **Cloud híbrida:** São combinações de nuvens públicas e privadas que trabalham juntas. Geralmente as TI terceirizam informações e processos não críticos para a cloud pública, mantendo os serviços e dados críticos dos negócios sob seu controlo. O ambiente trabalha para integrar perfeitamente aplicações externas noutras nuvens públicas e privadas, com os seus processos internos;
- **Cloud da comunidade:** Pode ser privada, comprada por um único utilizador para oferecer suporte a uma comunidade de utilizadores ou uma *cloud* híbrida com os custos distribuídos por alguns utilizadores da cloud. Geralmente é configurada como um ambiente de *sandbox*, onde os utilizadores da comunidade podem testar as suas aplicações ou aceder aos recursos através de um ambiente seguro.

Os quatro modelos de implementação, são representados na Figura 9.

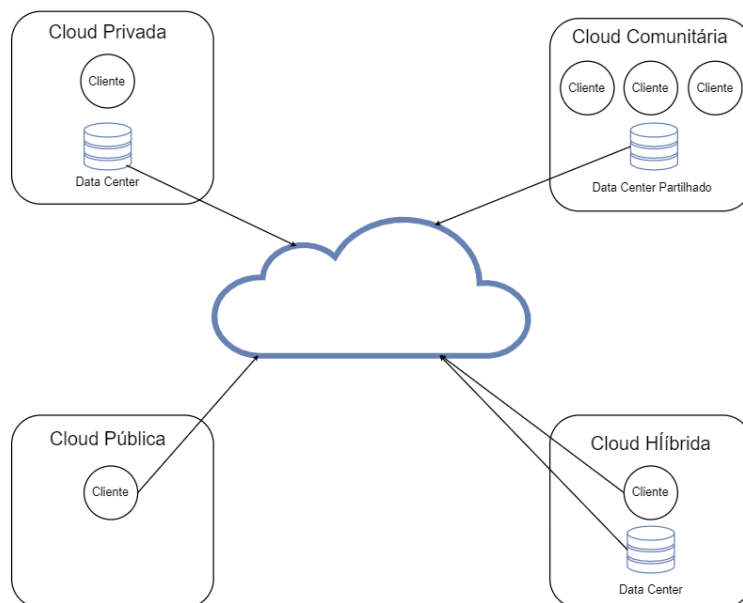


Figura 9: Modelos de Cloud Computing. Adaptado de Ambavane, S. A., Pawar, A. S., Verma, V. H., & Marathe, P. (2018).

2.3.3. Principais Elementos da Cloud Computing

Segundo Jain, A., & Kumar, R. (2014), os principais elementos da *Cloud Computing* são apresentados na Tabela 4:

Elementos	Definição
Segurança	Devido à escalabilidade dinâmica, abstração, transparência de localização, recursos virtualizados partilhados pelos recursos de várias localizações dos modelos de <i>Cloud Computing</i> , é difícil manter a confidencialidade e a integridade dos dados.
Balanceamento de carga	No ambiente em <i>cloud</i> , os servidores são monitorizados continuamente e, quando um deles não responde, um mecanismo de balanceamento de carga é chamado para evitar falhas no sistema.
Gestão de dados armazenados	Há um aumento exponencial nos dados armazenados na rede devido à terceirização de dados. Logo, a gestão de dados armazenada tornou-se um dos principais desafios para a implementação bem-sucedida da <i>Cloud Computing</i> , distribuindo os dados e otimizando o armazenamento de dados, mantendo o acesso rápido.
Provisionamento automatizado de serviços	A elasticidade é o recurso mais importante do ambiente em <i>cloud</i> . Devido a este recurso, os recursos podem ser alocados ou libertados de acordo com a procura. Como é possível utilizar ou libertar os recursos, mantendo o mesmo desempenho dos sistemas tradicionais e utilizando os recursos ideais.
Gestão de energia	A utilização eficiente de energia é um grande desafio. Estima-se que o custo de energia e refrigeração represente 53% do total de despesas operacionais dos <i>Data Centers</i> , portanto, existe a preocupação em diminuir a utilização de energia.

Tabela 4: Desafios da Cloud Computing. Adaptado de (Jain, A., & Kumar, R., 2014)

2.3.4. Arquitetura da *Cloud Computing Mobile*

Nos dias de hoje, os *smartphones* são omnipresentes nas atividades diárias da maioria da população mundial, e isso motivou as organizações a desenvolverem aplicações que podem ser facilmente acedidos por meio de *smartphones*. No entanto, recursos limitados (por exemplo, processador, memória e armazenamento de dados) dos dispositivos móveis apresentam alguns desafios de design para os programadores das aplicações móveis. Segundo Noor, T. H., Zeadally, S., Alfazi, A., & Sheng, Q. Z. (2018), existem três camadas diferentes:

- i. **Camada de utilizador móvel:** Consiste em muitos utilizadores acederem a serviços utilizando os seus dispositivos móveis. Os dispositivos móveis conectam-se à camada de rede móvel utilizando pontos de acesso sem fios ou satélite;
- ii. **Camada de rede móvel:** Consiste em várias operadoras de rede móvel que lidam com solicitações de utilizadores móveis e as informações são entregues através de estações base. As solicitações de utilizadores móveis e as transferências de informações são tratadas por serviços de rede móvel. Neste ponto, as operadoras de rede móvel ajudam a identificar os dados dos assinantes armazenados nas suas bases de dados. Após autenticação e autorização bem-sucedidas, a operadora entrega as solicitações dos utilizadores móveis na cloud. O utilizador móvel pode aceder os serviços correspondentes, conforme fornecidos pelos controladores na cloud;
- iii. **Camada de servidor:** Consiste em vários servidores de serviços de *Cloud Computing* que fornecem todos os tipos de serviços, incluindo IaaS, PaaS e SaaS. Estes serviços são elásticos e, podem ser aumentados ou reduzidos, com base no que os utilizadores necessitam.

A *Cloud Computing* fornece serviços para utilizadores, incluindo aqueles com *smartphones*, podem aceder aos serviços utilizando a *Internet*. De uma perspetiva geral, esta arquitetura demonstra a eficácia da *Cloud Computing* para atender às expectativas dos utilizadores que a utilizam. A principal motivação é complementar o desempenho de computação do dispositivo móvel, utilizando uma arquitetura orientada a serviços, como servidores *web*, aplicações e bases de dados. O design da arquitetura para a entrega dos serviços deve ser capaz de fornecer serviços orientados aos negócios, considerando as necessidades dos consumidores, como garantir a disponibilidade dos serviços, proteger os canais de comunicação e preservar a privacidade dos seus dados pessoais. Por exemplo, a execução aumentada é uma técnica que visa aumentar a capacidade dos dispositivos móveis, superando a questão das limitações de recursos executando algumas tarefas computacionais na *cloud* (Noor, T. H., Zeadally, S., Alfazi, A., & Sheng, Q. Z., 2018).

2.3.5. Tipos de *Cloud Computing*

A. *SaaS*

O SaaS (*Software as a Service*) é um modelo de entrega de *software* que fornece acesso a aplicações utilizando a *Internet* como um serviço baseado na *web*. As aplicações são desenvolvidas para serem acessíveis a vários utilizadores por meio de um navegador *web*.

O conceito SaaS pressupõe que o cliente utilize *software* com recurso à *Internet* pagando um valor pelo serviço que é disponibilizado pelo fornecedor. Nas soluções SaaS, o cliente não necessita de comprar *hardware* adicional e *software*, uma vez que estas aplicações *web* são colocadas em execução na infraestrutura do fornecedor de serviços.

Segundo (Rani, B. K., Rani, B. P., & Babu, A. V., 2015), as principais características do SaaS:

- O *software* é disponibilizado através da *Internet*;
- O *software* é mantido através de servidores;
- A licença do *software* é baseada em assinatura por um determinado período de tempo;
- Não é necessária manutenção no lado do utilizador final e, portanto, as aplicações SaaS são muito económicas;
- O *software* está disponível sob procura e pode ser escalado de acordo com a procura;
- O *software* é atualizado e atualizado automaticamente e também suporta multitenancy.

GoogleApps, Oracle on Demand, Salesforce.com e SQL Azure, são alguns exemplos de SaaS.

B. *PaaS*

As soluções PaaS (*Platform as a Service*) constituem o *middleware*, sobre o qual as aplicações são desenvolvidas com o objetivo de fornecerem uma plataforma de desenvolvimento e implantação para a execução de aplicações na *cloud*.

Segundo (Rani, B. K., Rani, B. P., & Babu, A. V., 2015), as principais características do PaaS:

- As interfaces de segurança, escalabilidade e serviços *web* integradas são fornecidas pelo PaaS;
- Ferramentas internas para definir regras de negócios e definir processos de fluxo de trabalho e aprovação são fornecidas pelo PaaS;

- A integração de aplicações com outras aplicações na mesma plataforma é facilitada;
- O PaaS fornece interfaces de serviços *web* que permitem conectar as aplicações que se encontram fora da plataforma.

Force.com, Google AppEngine, Plataforma Windows Azure e GoGrid Cloudcenter são alguns dos exemplos de PaaS.

C. IaaS:

As soluções IaaS (Infrastructure as a Service) são o segmento de mercado mais popular e desenvolvido da *Cloud Computing*. As soluções de IaaS trazem todos os benefícios da virtualização de hardware.

Segundo (Rani, B. K., Rani, B. P., & Babu, A. V., 2015), as principais características do IaaS:

- O IaaS fornece às máquinas virtuais SO's pré-instalados;
- Os recursos estão disponíveis sob procura;
- IaaS permite armazenar cópias de dados em diferentes locais;
- Os recursos da podem ser facilmente escalados.

Amazon ECC, Eucalyptus, GoGrid, Flexiscale, Linode, RackSpace Cloud e Terremark são exemplos de IaaS.

Com base nas características dos tipos de *Cloud Computing*, a Figura 10 ilustra a arquitetura da mesma.

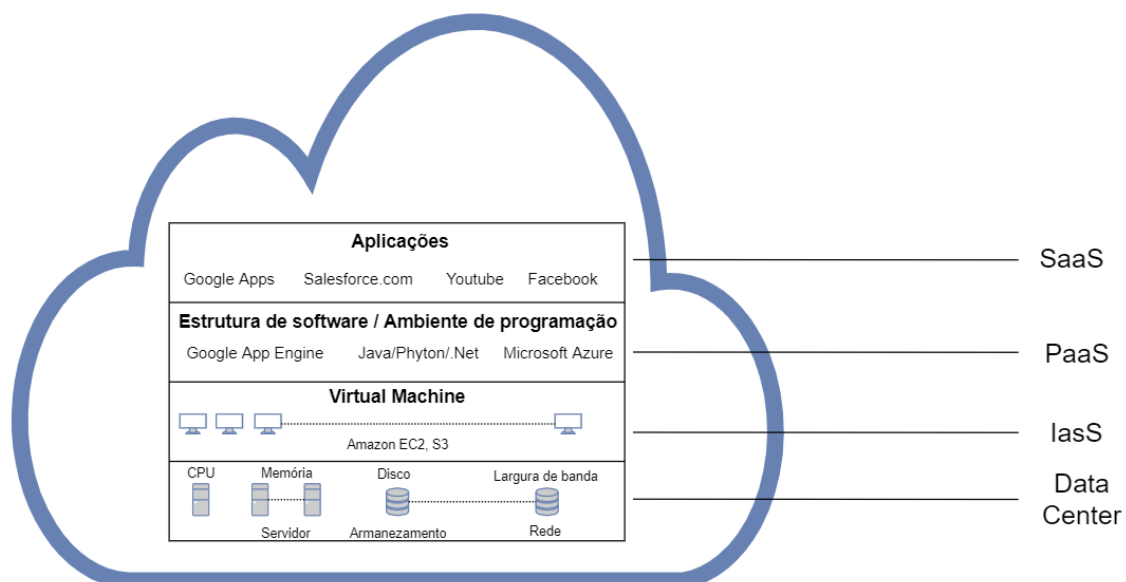


Figura 10: Arquitetura da Cloud Computing. Adaptado de Padhy, R. P., Patra, M. R., & Satapathy, S. C. (2011)

2.4. Aplicações *Cross-Platform*

Nos dias de hoje, existe um forte crescimento de novas ferramentas e aplicações de desenvolvimento de *software* para dispositivos móveis inteligentes. O rápido crescimento do mercado móvel motivou a implementação de ambientes de desenvolvimento de *software* multiplataforma que, tornaram o desenvolvimento mais fácil e eficiente. As principais categorias de aplicações produzidas por ambientes de *software* são: aplicações nativas, híbridas e *web* (Xanthopoulos, S., & Xinogalos, S., 2013).

2.4.1. Principais Objetivos das Aplicações *Cross-Platform*

Segundo Xanthopoulos, S., & Xinogalos, S. (2013), apresentam-se na Tabela 5 os principais objetivos das Aplicações *Cross-Platform*.

Elementos	Definição
Escalabilidade e manutenção das aplicações	Oferecer a possibilidade de manter e melhorar a aplicação. Por exemplo, se novas opções necessitarem de ser implementadas numa aplicação, as modificações deverão estar sempre no nível da <i>cross-platform</i> e, posteriormente, implementadas em plataformas diferentes.
Acesso aos recursos dos dispositivos	A aplicação desenvolvida utilizando ferramentas <i>cross-platform</i> , deve ter todo o acesso aos recursos do dispositivo, sem ser necessária uma camada intermediária.
Consumo de recursos	É necessário que os programadores levem em consideração o consumo de recursos, incluindo processador e utilização da memória, mas a <i>cross-platform</i> deve oferecer otimização automática na implantação em cada plataforma.
Segurança	A questão da segurança é um dos critérios mais importantes no desenvolvimento móvel. Parte deste problema pode ser resolvida pelo programador. O programador deve saber quais são os dados que ficam armazenados no dispositivo, porque os dispositivos podem ser roubados ou cair em mãos não autorizadas.
Ambiente de desenvolvimento	O ambiente de desenvolvimento é importante nas aplicações <i>cross-platform</i> , mas, o mais importante, é ter vários simuladores com sistemas diferentes (<i>Android</i> , <i>iOS</i> , <i>Windows Phone</i> etc.) para obter uma boa visibilidade e noção da aparência da aplicação antes que seja implantada.

Tabela 5: Requisitos necessários para produzir e manter aplicações móveis de alto desempenho. Adaptado de (Xanthopoulos, S., & Xinogalos, S., 2013)

2.4.2. Processo de Desenvolvimento de Aplicações Móveis

Segundo Popa, M. (2013), o processo de desenvolvimento de aplicações móveis deve levar em consideração a ampla variedade de dispositivos móveis com diferentes SO's móveis ou versões dos mesmos. Cada versão do SO possui recursos específicos, como hardware, segurança, tamanhos e resoluções de ecrã, modelos e velocidades de conectividade etc. Os programadores de aplicações móveis devem estar preparados para:

- Codificar a aplicação para diferentes sistemas/versões de operativos móveis;
- Implementar a aplicação móvel para uma grande variedade de resoluções do ecrã;
- Considerar os recursos de hardware específicos do dispositivo que podem oferecer funcionalidades especiais à aplicação em comparação com os outros.

Segundo Popa, M. (2013), estas etapas visam testes e garantia de qualidade, suporte, implantação e distribuição:

- **Forte procura:** Uma massa crítica de dispositivos de hardware pode fornecer uma procura suficiente de aplicações móveis, sendo este um problema crítico para fornecer um retorno significativo do investimento;
- **Uma base sólida:** Existem algumas aplicações, serviços e plataformas móveis como ponto de partida no desenvolvimento futuro de aplicações móveis;
- **Gestão dos dispositivos móveis:** É utilizado para entrega de serviços e políticas de segurança;
- **Arquitetura Orientada a Serviços:** Uma camada de serviço abstrata é utilizada pela aplicação móvel para aceder à lógica e os dados do negócio;
- **Segurança adequada:** Os problemas acima são considerados antes de iniciar o processo de desenvolvimento móvel para integração de dispositivos móveis na arquitetura corporativa.

2.4.3. Estratégias para o Desenvolvimento de Aplicações Móveis

A. Aplicações nativas

São aplicações dependentes da plataforma, sendo desenvolvidos com *SDK's* e ferramentas fornecidas pelo proprietário da plataforma. A estratégia restringe o número de dispositivos móveis executando a aplicação, mas possui uma grande vantagem em relação à otimização da utilização que a plataforma oferece aos programadores. Existem aplicações móveis para as quais, a estratégia nativa é a única opção devido à sua especificidade.

Segundo More, K. A., & Chandran, M. P. (2016), as aplicações nativas têm vantagens como:

- **Altamente gráfico:** Oferecem os melhores gráficos e animações. Se uma organização necessita de aplicações altamente gráficas, como um jogo, as aplicações nativas são a melhor opção;
- **Distribuição de loja das aplicações:** São distribuídas por meio das lojas da sua própria plataforma. Embora muitas vezes não seja um requisito para aplicações de negócios, a distribuição nestas lojas é essencial para as organizações que exigem uma ampla distribuição do consumidor ou que, necessitam vender as mesmas;
- **Integração de dispositivos:** Oferecem acesso total ao hardware do dispositivo, como o sensor de *GPS*, contatos, câmara, microfone, giroscópio, acelerómetro, etc. Estes recursos são essenciais para qualquer aplicação que exija dados do dispositivo, como localização geográfica ou posições/movimentos dos dispositivos.

Segundo More, K. A., & Chandran, M. P. (2016), as aplicações nativas têm desvantagens como:

- **Nenhuma portabilidade:** Como cada aplicação só é executada numa única plataforma, as organizações que criam aplicações fazem uma escolha de plataforma, tendo várias possíveis;
- **Instabilidade da plataforma:** Uma plataforma popular nos dias de hoje, pode desaparecer em poucos anos, ou até podem surgir novas plataformas que passem a dominar o mercado;
- **Custo de desenvolvimento:** Embora o custo de desenvolvimento varie dependendo da complexidade da aplicação, é facilmente a abordagem mais dispendiosa e demorada;
- **Tempo de desenvolvimento:** Quando se pretende desenvolver aplicações para mais de uma plataforma, os requisitos de tempo aumentam;
- **Custo de manutenção:** Embora todas as aplicações necessitem de atualizações e manutenção regulares, estas exigem a manutenção mais recente disponível, quando comparados com as outras duas opções de aplicações móveis;
- **Controlo limitado:** Quando colocados numa loja de aplicações, são totalmente controladas pelo proprietário da mesma.

B. Aplicações híbridas

São aplicações *web* envolvidas num ambiente nativo, que atenuam as desvantagens das aplicações *web*, fornecendo acesso aos recursos da plataforma móvel por meio do ambiente nativo, tendo acesso entre a linguagem de programação específica *web* e o código nativo que fornece acesso aos recursos do dispositivo. A estratégia híbrida combina as vantagens de estratégias nativas e *web*, sendo a melhor estratégia para o desenvolvimento de aplicações para várias plataformas.

Segundo More, K. A., & Chandran, M. P. (2016), as aplicações híbridas têm vantagens como:

- **Sem custo nativo:** Para fins comerciais, que preferem a aparência e a funcionalidade das aplicações nativas, sem os altos custos de desenvolvimento, as aplicações híbridas são a melhor escolha. Estes atributos nativos, tornam uma aplicação híbrida praticamente indistinguível das nativas. Exemplos como o *LinkedIn*, o *Foursquare* e o *Twitter*, representam aplicações híbridas;
- **Integração de dispositivos:** As aplicações *web* para dispositivos móveis oferecem integração de dispositivos quase total, com algumas exceções. É aí que entram as aplicações híbridas, que oferecem acesso total ao dispositivo;
- **Custos baixos para desenvolvimento *cross-platform*:** Embora não seja tão simples quanto a abordagem para aplicações *web*, a criação de uma aplicação híbrida é consideravelmente mais barata do que a criação de uma aplicação nativa.

Segundo More, K. A., & Chandran, M. P. (2016), as aplicações híbridas têm desvantagens como:

- **Gráficos limitados:** Apesar da sua aparência nativa, as aplicações híbridas oferecem as mesmas habilidades gráficas das aplicações *web*. Como tal, as aplicações altamente gráficas são mais adequadas para a abordagem nativa;
- **Requer familiaridade com a *framework* móvel:** Transformar aplicações *web* para dispositivos móveis em aplicações híbridas exige familiaridade com estruturas móveis.

C. Aplicações *web*

São aplicações desenvolvidas utilizando tecnologias padrão como: HTML, JavaScript e CSS. Segundo More, K. A., & Chandran, M. P. (2016), as vantagens das aplicações *web* são:

- Grande disseminação de tecnologias *web*;
- Custos reduzidos;
- Atualização única da base de código a qualquer momento;
- Disponibilidade permanente através da *Internet*;
- Evitar as lojas de aplicações com regras específicas para publicação de aplicações.

Segundo More, K. A., & Chandran, M. P. (2016), as desvantagens das aplicações *web* são:

- Não poder utilizar os recursos do dispositivo;
- A utilização *offline* é restrita.

2.4.4. Ferramentas de Desenvolvimento *Cross-Platform*

Existem várias ferramentas de desenvolvimento para as aplicações *cross-platform*. Analisam-se as características de duas ferramentas específicas, em concreto, *PhoneGap* e *DragonRad*, bem como outras tecnologias para produção de pacotes de *software cross-platform responsive*.

2.4.4.1. *PhoneGap*

O *PhoneGap* é uma ferramenta de desenvolvimento móvel de código aberto desenvolvida pela Adobe System Inc., sob licença Apache 2.0. O *PhoneGap* permite que os programadores e organizações desenvolvam aplicações gratuitas, comerciais e de código aberto, além de oferecer a possibilidade de utilizar qualquer combinação de licenças. O ambiente de desenvolvimento permite a criação de aplicações para SO's *Android*, *Bada*, *BlackBerry*, *iOS*, *Symbian*, *webOS* e *Windows Phone*. O *PhoneGap* é uma solução útil para desenvolver aplicações móveis utilizando linguagens de programação *web*, como, *HTML5*, *CSS3* e *JavaScript*, e a funcionalidade dos *SDK's*, permite utilizar linguagens menos conhecidas, como *Objective-C* ou outras linguagens. Como cada *software* de código aberto válido, é composto por componentes e extensões. As aplicações *PhoneGap* são híbridas, o que significa que não são puramente nativas ou baseadas na *web*. Além disso, o *PhoneGap* também oferece a possibilidade de estender a ferramenta desenvolvendo *plug-ins* próprios (Palmieri, M., Singh, I., & Cicchetti, A., 2012).

Adotando uma abordagem de *cross-platform*, a criação e a manutenção de aplicações podem ser aprimoradas, porque os programadores escrevem apenas um único código-fonte para qualquer SO móvel. O *PhoneGap* não fornece um IDE para desenvolver aplicações, mas é possível utilizar por exemplo, o IDE *Eclipse* para *Android*, *XCode* para *iOS* etc. Esta abordagem não permite que os programadores tenham um ambiente de desenvolvimento centralizado, portanto, o esforço necessário para compilar o código-fonte e produzir o produto final é alto. Graças à utilização de diferentes IDE's para o desenvolvimento das aplicações, o *PhoneGap* pode ser executado em diferentes SO's, como *Mac*, *Linux* e *Microsoft Windows*. Infelizmente, algumas vezes existem algumas exceções, porque nem todos os IDE's são compatíveis com todos os SO's. A arquitetura do *PhoneGap* é composta principalmente por três camadas: aplicação, *PhoneGap* e SO e API nativa (Palmieri, M., Singh, I., & Cicchetti, A., 2012).

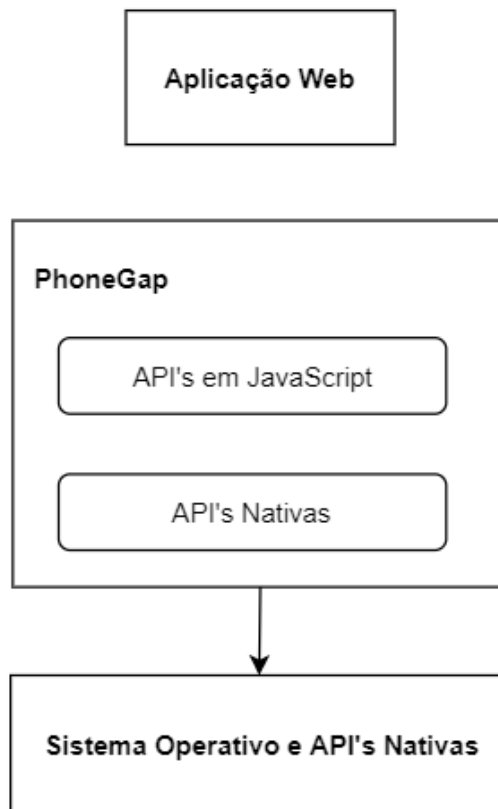


Figura 11: Aplicação Web utilizando o PhoneGap. Adaptado de (Palmieri, M., Singh, I., & Cicchetti, A., 2012)

A Figura 11 ilustra a camada superior, que representa o código fonte da aplicação e a camada central, é composta por JavaScript e API's nativas. Esta camada, é responsável pela interface entre a aplicação *web* e as camadas *PhoneGap*. Além disso, também trata da interface entre as API's JavaScript utilizadas pela aplicação com as API's nativas pelos SOs móveis. A funcionalidade desta camada é manter o relacionamento entre as API's JavaScript e as API's nativas de cada SO móvel. O *PhoneGap* fornece API's JavaScript para os programadores, que permitem o acesso a funcionalidades do dispositivo, como acelerómetro, Bluetooth, calendário, câmara, bússola, contatos, *GPS*, menu, NFC, etc.

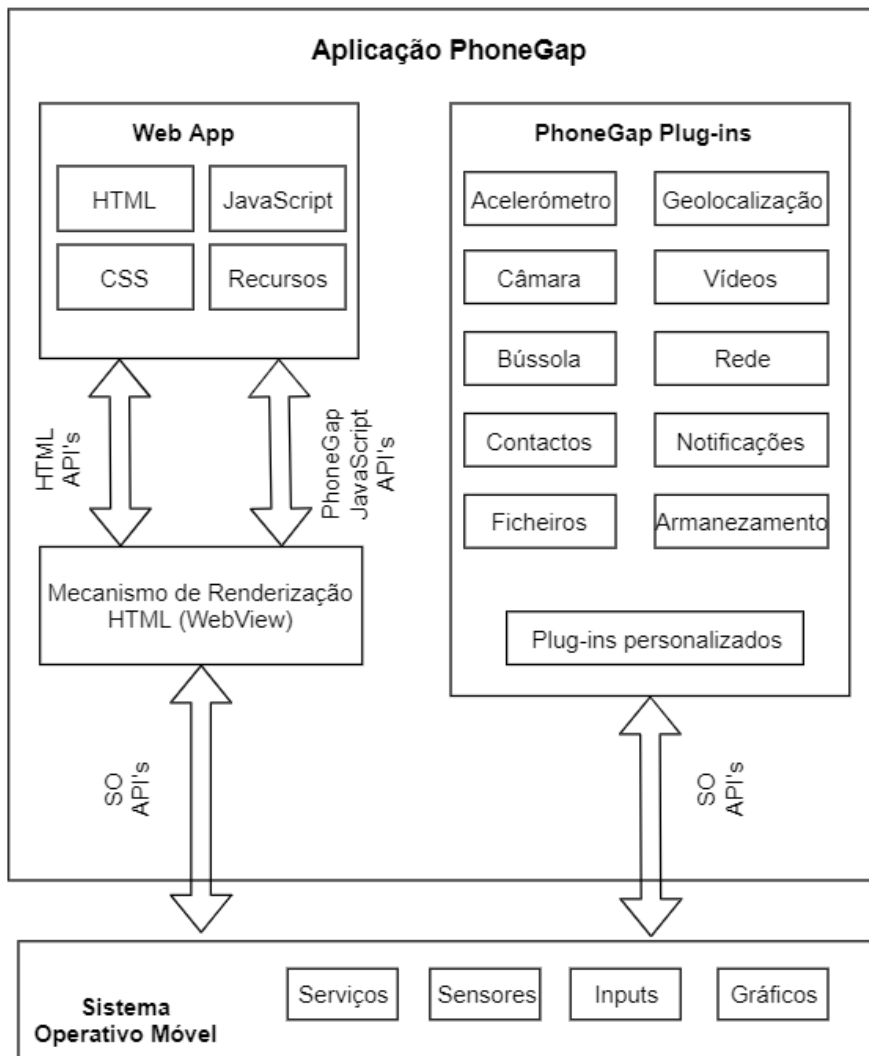


Figura 12: Código fonte do PhoneGap. Adaptado de (Palmieri, M., Singh, I., & Cicchetti, A., 2012)

Na Figura 12 apresenta-se a arquitetura mais detalhada, fornecida pela IBM. A figura representa todos os componentes sobre a aplicação *web*, o mecanismo de renderização HTML, a API e as camadas do SO. Além disso, diversas interfaces são detalhadas, como a interface entre as camadas da API do *PhoneGap* e da API nativa.

2.4.4.2. *DragonRad*

O *DragonRad 5.0* é uma plataforma de desenvolvimento de aplicações móveis da Seregon Solutions Inc., distribuída sob uma licença comercial. Permite que os programadores projetem, façam a gestão e implantem aplicações móveis em *iOS*, *Android*, *BlackBerry* e *Windows Mobile*. A ferramenta concentra-se em aplicações corporativas móveis orientadas a base de dados, com fácil e amplo leque de suporte a bases de dados. Fornece o ambiente de D&D, que ajuda os programadores a economizarem tempo e a desenvolverem lógicas. (Palmieri, M., Singh, I., & Cicchetti, A., 2012).

É necessário configurar o servidor e a base de dados com base nas necessidades dos programadores, mas contém todos os pré-requisitos de servidor e de bases de dados, como o Tomcat, MySQL, etc. As possibilidades de extensão em termos de adição de *plug-ins* e outro suporte à *framework* são bastante limitadas, mas facilita a integração e sincronização do sistema da base de dados com funções nativas dos sistemas operativos móveis acima definidos, como Contatos, Calendário, Localização geográfica, Menu e Armazenamento (Palmieri, M., Singh, I., & Cicchetti, A., 2012).

Segundo (Palmieri, M., Singh, I., & Cicchetti, A., 2012), a arquitetura do *DragonRad* é composta por três componentes principais:

1) *DragonRad Designer*

É um ambiente visual de D&D ou GUI para os programadores projetarem, desenvolverem e instalarem aplicações móveis. Os recursos de D&D não estão apenas a ajudar os programadores a projetarem aplicações, mas também a reduzir os esforços de manutenção e codificação.

2) *DragonRad Host*

Pode ser executado através de servidores Linux ou Windows, preenchendo a lacuna entre a base de dados de aplicações corporativas e móveis. Mantém a comunicação com o dispositivo móvel, que também inclui a consulta de transação durante a indisponibilidade da rede. Desempenha o papel de estabelecer uma conexão sem problemas com o acesso à base de dados e sincronização. Os recursos mais relevantes são:

- Receber consultas no dispositivo;
- Executar consultas num destino específico;
- Enviar dados para o dispositivo com base em solicitações;
- Enviar atualizações dos dispositivos das bases de dados;
- Compactar e verificar de dados de pacotes de dados.

3) Cliente *DragonRad*

Este componente comporta-se como uma aplicação nativa no dispositivo, que auxilia a executar e interpretar o código da aplicação. Também possui o recurso de personalizar aplicações como ícone de alteração, nome, projeto *DragonRad* e *link* para instalação do *host DragonRad*, conforme ilustra a Figura 13.

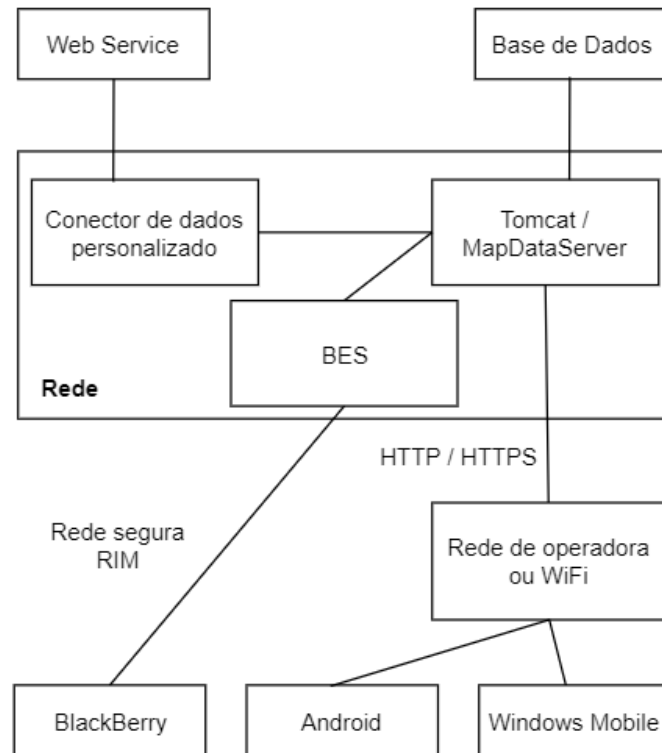


Figura 13: Exemplo de uma aplicação desenvolvida utilizando o *DragonRad*. Adaptado de (Palmieri, M., Singh, I., & Cicchetti, A., 2012)

2.4.5. Tecnologias para suporte ao desenvolvimento de sistemas híbridos *responsive*

Existem vários tipos de tecnologias como ASP, Perl, Python ou Java. Tal como as anteriores, o conjunto de tecnologias HTML5/CSS3/PHP/MySQL é frequentemente utilizado em contexto web uma vez que permite a realização de aplicações que podem ser executadas em *PC Desktop*, *Smartphones*, *Tablets* e outros dispositivos.

O HTML5 (estrutura) e CSS3 (apresentação) são utilizadas para criar *websites*, tendo como principais benefícios a reprodução de vídeo e áudio, armazenamento em *cache offline*, código limpo, compatibilidade com os principais navegadores e otimização móvel.

O PHP para aceder a bases de dados MySQL funciona em qualquer plataforma independentemente do sistema operativo. O código é aberto, tratando-se de uma linguagem onde são feitas melhorias constantemente.

Um dos desafios de desenvolvimento das aplicações *web* consiste em garantir que as mesmas sejam visualizadas de forma correta e adaptadas ao dispositivo do cliente. Assim, o termo *responsive* significa que na perspectiva de *web design* as peças de *software* permitem que os *layouts* das aplicações se adaptem à resolução do dispositivo cliente, e.g. *smartphone*, *tablet* ou *desktop*. Neste domínio, a tecnologia HTML5/CSS3 permite que a visualização de conteúdos de aplicações *web* seja *responsive*. O W3C especifica os standards HTML e CSS.

Laravel é uma *framework* que utiliza o padrão MVC (*Model-View-Controller*) em PHP, e que permite o desenvolvimento e reutilização modular de código. E.g. Laravel é composta por mais de vinte pacotes de bibliotecas e *frameworks* diferentes que são geridas por um gestor de dependências, o *composer*. Possui ainda ferramentas para teste, definição de mecanismos de autenticação, construção de *queries*, compatibilidade com várias bases de dados e migração de dados.

2.5. Engenharia Web

O processo de desenvolvimento de um *website* é de uma conceção muito vasta, e pretende considerar uma sucessão de atividades, normalmente agrupadas em fases e tarefas, que são efetuadas de forma sistemática e uniformizada, realizadas por intervenientes com responsabilidades bem definidas.

Para existir um *website*, são necessárias tecnologias *Client-Side* e tecnologias *Server-Side*. Nas primeiras tecnologias, a aplicação executa diretamente no computador do utilizador e dá a resposta a alguma interação que é feita no *website* (E.g. HTML, CSS3, JavaScript, etc.). Em relação ao *Server-Side*, sempre que o utilizador carregue uma página, o navegador vai executar o comando diretamente no servidor. (E.g. PHP, ASP, ASP.NET, etc.) (Kappel, G., 2006).

Segundo Pressman (2005), neste campo de ação, classifica a construção de um *website* em três fases:

- **Conceção:** onde se pretende reconhecer “o que o *website* deve conter”, nomeadamente a informação a processar, as funcionalidades a implementar, as restrições existentes;
- **Implementação:** onde o objetivo é identificar “como fazer o *website*” e construí-lo na realidade, definindo e construindo as estruturas de dados, os testes a realizar, a concretização do alojamento e do domínio. No final desta fase, deverá ser disponibilizado o *website* completamente funcional;
- **Manutenção:** inclui todas as alterações posteriores à aceitação do *website* pelo cliente como correção de erros, introdução de melhorias e/ou novas funcionalidades e atualizações.

Segundo Pressman (2005), as características que podem ser encontradas na grande maioria das aplicações web são:

- **Rede Intensiva:** aplicações *web* utilizam recursos de rede por natureza, que devem atender às necessidades dos utilizadores. Pode estar disponível na *Internet*, *Intranet* ou *Extranet*;
- **Focado no Conteúdo:** em muitos casos, a função primária da aplicação *web* é utilizar hiperligações para apresentar textos, gráficos, e vídeo para os utilizadores;
- **Evolução Contínua:** ao contrário das aplicações convencionais, que evoluem através de uma série de versões planeadas e lançadas em determinados intervalos de tempo, as aplicações web evoluem continuamente.

Pressman (2005) detalha também as principais diferenças entre desenvolver uma aplicação *web* em comparação a desenvolver um *software* tradicional:

- **Imediatismo:** o tempo que um *website* completo precisa ficar pronto pode ser apenas alguns dias ou semanas. Os programadores devem, portanto, utilizar métodos de planeamento, análise, projeto, implementação e testes, que estejam adaptados para estes cronogramas comprimidos necessários no desenvolvimento para a *web*;
- **Segurança:** as aplicações *web* estão disponíveis via rede, como tal, é difícil ou até mesmo impossível limitar a população de utilizadores que irão aceder à aplicação. Para se conseguir proteger o conteúdo e fornecer métodos seguros de transmissão de dados, é preciso implementar medidas rígidas de segurança na aplicação e na infraestrutura do mesmo;
- **Estética:** um aspeto relevante das aplicações *web* é o seu aspeto. Quando uma aplicação é projetada para vender produtos ou ideias, a estética pode ser tão ou mais importante para o sucesso, quanto o projeto técnico.

Quando se começa a desenvolver um artefacto para a construção de uma aplicação *web*, Bolchini, D. (2000), demonstra cada um dos passos a seguir para que o artefacto seja construído da forma mais correta possível, que estão descritos na Tabela 6:

Artefacto	Descrição
Requisitos	Determinar os requisitos funcionais e não funcionais da aplicação;
Briefing	Processo interativo com o cliente no qual se procura obter uma definição de contexto e diretrizes para o projeto do ponto de vista de design. Deve definir: o contexto da aplicação, definição de cores, padrões gráficos, aplicação de logotipos e identidade visual;
Storyboard	Modelo sequencial da parte visual do sistema com esboço visual e indicação das principais funcionalidades e informações;
Modelo Conceptual	Permite uma visão de alto nível da aplicação a partir das suas entidades principais e do relacionamento entre elas;
Modelo E-R (ou classes)	Permite uma visão das entidades (ou classes) do projeto e dos relacionamentos entre elas. Serve para determinar a arquitetura da informação, bem como a estrutura de dados e atributos. É diferente do modelo conceptual porque está mais focado em dados e por representar todas as entidades, não apenas as principais;
Diagrama de Navegação	Mostra como os utilizadores irão navegar através do <i>website</i> por meio de uma representação em árvore das páginas e <i>links</i> entre elas;
Desenho das interfaces	Pode ser feito inicialmente em papel ou em formato digital. Visa o esquema de cores e a disposição dos elementos de navegação;
Diagrama de composição	Modelo no qual é organizado o conteúdo das páginas (imagens, informação textual, dados, formulários, <i>links</i> , etc.);
Modelo de personalização	Descreve como as páginas são apresentadas para os diferentes utilizadores, de acordo com a personalização definida por estes;
Elementos de HTML	Elementos de interface gráfica para a construção das páginas HTML, como menus, imagens de fundo, ícones, etc.;
Templates	Padrões reaproveitáveis para a página de aplicação, codificados em HTML;
Protótipo não funcional	Sequência de páginas da aplicação e navegação com dados estáticos;
Protótipo funcional	Implementação das principais funcionalidades da aplicação (dados dinâmicos, interatividade e transações).

Tabela 6: Descrição dos artefactos no desenvolvimento de aplicações Web. Adaptado de (Bolchini, D., 2000)

3. Arquitetura da *Framework*

3.1. *Framework SaaS Multifuncional (FSM)*

A *Framework SaaS Multifuncional* tem por objetivo permitir a instanciação na *cloud* das aplicações multiplataforma SaaS para suportar diferentes necessidades de clientes em função de diferentes contextos e áreas de negócio.

Num cenário onde a utilização de dispositivos móveis prolifera, as aplicações multiplataforma são uma solução para que as mesmas sejam executadas e visualizadas corretamente independentemente do sistema operativo do cliente.

Assim, o objetivo é escrever o código-fonte da FSM, de modo as aplicações possam ser executadas em diferentes sistemas operativos e dispositivos do cliente.

Dispondo de uma infraestrutura comum, as heurísticas de áreas de negócio específicas são desenvolvidas incrementalmente através de peças de *software* modulares (*plugins*).

3.2. Pacotes Principais

A arquitetura FSM baseia-se em dois pacotes principais:

- *SAAS Service*;
- *SAAS Client*.

A Figura 14 ilustra a relação entre estes pacotes.

O pacote *SAAS Client* utiliza o pacote *SAAS Services* para executar um conjunto de serviços que garantem funcionalidades como a configuração dos plugins necessários, gestão de utilizadores do sistema, personalização de logótipos do cliente, esquema de cores da aplicação e configuração de funcionalidades *Web 2.0*.

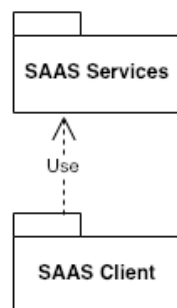


Figura 14: Pacotes da FSM

3.3. Componentes FSM

A visão geral da arquitetura de componentes FSM é descrita na Figura 15.

Responsive representa uma aplicação *web responsive* que utiliza o sistema *SAAS Services*. Este sistema inclui uma biblioteca e uma base de dados. Os algoritmos de acesso à base de dados (*SAAS Database*) são implementados no componente *Library* do sistema *SAAS Services*.

O sistema *SAAS Services* disponibiliza ainda dois tipos de interfaces:

- (1) interfaces para suportar aplicações *web (Responsive)*, e
- (2) interfaces para suportar integração externa de dados (*ETL Engine*).

Como ilustrado na Figura 15, o componente *SAAS Plugins* oferece uma interface para disponibilizar o acesso aos *plugins KM (Knowledge Management)*, *DSS (Decision Support System)* e *EM (Event Management)*. Estes *plugins* são responsáveis pela implementação de heurísticas específicas para aplicações SaaS em contexto de Sistemas de Gestão de Conhecimento, Sistemas de Suporte à Decisão, e Sistemas de Gestão de Eventos respetivamente.

Salienta-se que o *plugin DSS* corresponde ao componente SCP, podendo ser instanciados de igual modo no contexto do problema os *plugins KM* (para manutenção de informação relacionada) e *EM* (para gestão de eventos, e.g. calendário de jogos/apostas).

O componente *ETL* visa garantir que as aplicações SaaS disponham de ferramentas *ETL (Extract, Transform and Load)*. O objetivo destas ferramentas de *software* é extrair dados de diversos sistemas, efetuar a sua transformação de acordo com regras de negócio e por fim carregar os mesmos para o sistema de destino.

Deste modo, o sistema é resiliente a diferentes aplicações.

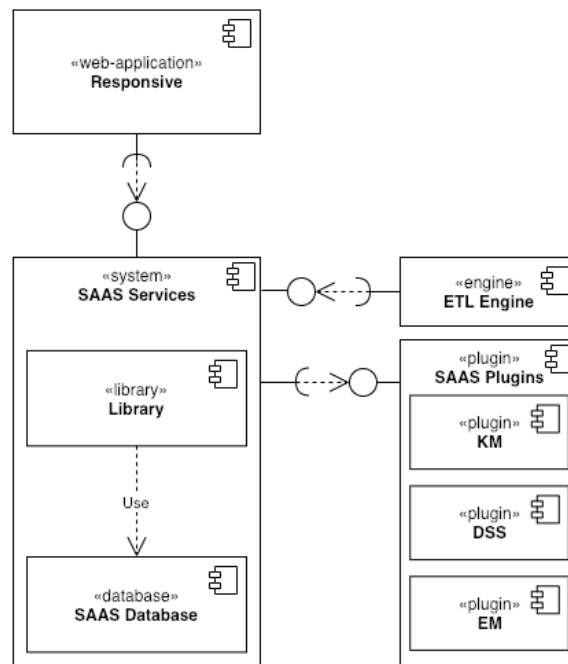


Figura 15: Arquitetura de Componentes FSM

3.4. Tecnologia de Desenvolvimento

A FSM é desenvolvida com tecnologia Laravel que foi introduzida na secção 2.4.5.

Laravel é uma *framework* que utiliza o padrão MVC (*Model-View-Controller*) em PHP, e que permite o desenvolvimento e reutilização modular de código. E.g. Laravel é composta por mais de vinte pacotes de bibliotecas e *frameworks* diferentes que são geridas por um gestor de dependências, o *composer*. Possui ainda ferramentas para teste, definição de mecanismos de autenticação, construção de *queries*, compatibilidade com várias bases de dados e migração de dados.

A Figura 16 ilustra as relações dos principais módulos da *framework* de desenvolvimento Laravel.

O módulo *Client* representa um *Web Browser* ou um *cliente API*. O módulo *Controller* suporta as principais operações lógicas do sistema (e.g. gestão de encaminhamentos, controladores e recursos). O módulo *Model* suporta as ligações e respetivos mapeamentos de atributos da base de dados. O módulo *Data Source* representa a base de dados que pode ser do tipo PostgreSQL, MongoDB ou MySQL. O módulo *View* utiliza *templates* PHP e é o responsável pelo tratamento visual dos dados, através do *Blade*, um motor de processamento de modelos HTML. Finalmente, na Figura 16 as setas sólidas representam *pedidos*, enquanto que as setas tracejadas representam *respostas*.

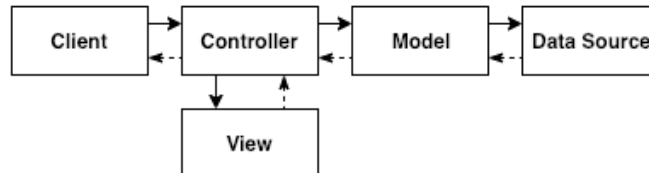


Figura 16: Módulos da Framework de Desenvolvimento

4. Sistema SaaS - SCP

O *Sistema SaaS – SCP* destina-se a ser aplicado em contexto de desenvolvimento da aplicação integrada na FSM. I.e., no momento da instanciação da plataforma para *Online Betting* com o objetivo de concretizar as seguintes funcionalidades comuns a sistemas de suporte à decisão:

- Lidar com grandes quantidades de dados, e suportar as respetivas pesquisas a bases de dados;
- Obter e processar dados de diferentes fontes, incluindo dados internos e externos;
- Ter flexibilidade ao nível da produção de relatórios para atender às necessidades do decisor/cliente;
- Dispor de gráficos, linhas de tendência e vistas tabulares;
- Realizar análises e comparações;
- Suportar abordagens heurísticas, em função das necessidades do decisor.

Para concretizar estes objetivos, propõe-se usar a FSM enquanto suporte para a plataforma SaaS com o objetivo de garantir as funcionalidades genéricas:

- Gestão de utilizadores;
- Aconselhamento com base nas heurísticas desenvolvidas para o modelo de negócio;
- Gestão de pagamentos *online*;
- Análise estatística de resultados;
- *Mashup* de mecanismos de visualização de dados.

4.1. Diagrama de Classes

A Figura 17 apresenta o diagrama das principais classes do sistema que representa as relações de multiplicidade entre as mesmas e respetivos atributos. Esta abordagem visa definir o futuro mapeamento para a criação do modelo de dados da base de dados.

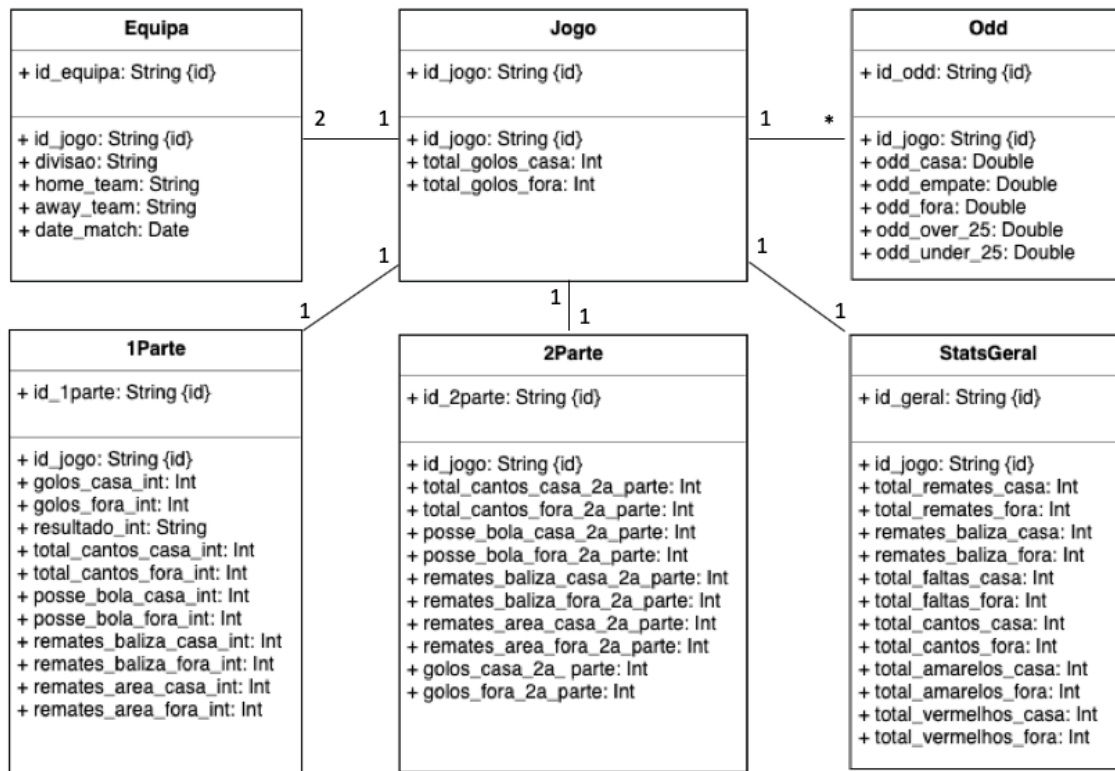


Figura 17: Principais classes do sistema SaaS - SCP, multiplicidades e respetivos atributos

Com base nos atributos incluídos no diagrama de classes, a base de dados deverá conter as seguintes tabelas e atributos:

- **Jogo:** id_jogo, total_golos_casa e total_golos_fora;
- **Equipa:** id_equipa, id_jogo, divisão, home_team, away_team e date_match;
- **Odd:** id_jogo, odd_casa, odd_empate, odd_fora, odd_over_25 e odd_under_25;
- **1Parte:** id_jogo, golos_casa_int, golos_fora_int, resultado_int, total_cantos_casa, total_cantos_fora, posse_bola_casa_int, posse_bola_fora_int, remates_baliza_casa_int, remates_baliza_fora_int, remates_area_casa_int e remates_area_fora_int;
- **2Parte:** id_jogo, total_cantos_casa_2a_parte, total_cantos_fora_2a_parte, posse_bola_casa_2a_parte, posse_bola_fora_2a_parte, remates_baliza_casa_2^a_parte, remates_baliza_fora_2a_parte, remates_area_casa_2a_parte, remates_area_fora_2a_parte, golos_casa_2a_parte e golos_fora_2a_parte;
- **StatsGeral:** id_jogo, total_remates_casa, total_remates_fora, remates_baliza_casa, remates_baliza_fora, total_faltas_casa, total_faltas_fora, total_cantos_casa, total_cantos_fora, total_amarelos_casa, total_amarelos_fora, total_vermelhos_casa total_vermelhos_fora.

4.2. Casos de Uso

Um caso de uso define uma sequência de ações executadas pelo sistema que dão origem um resultado de valor observável para um ator em particular.

A Figura 18 apresenta os principais casos de uso do sistema SaaS - SCP.

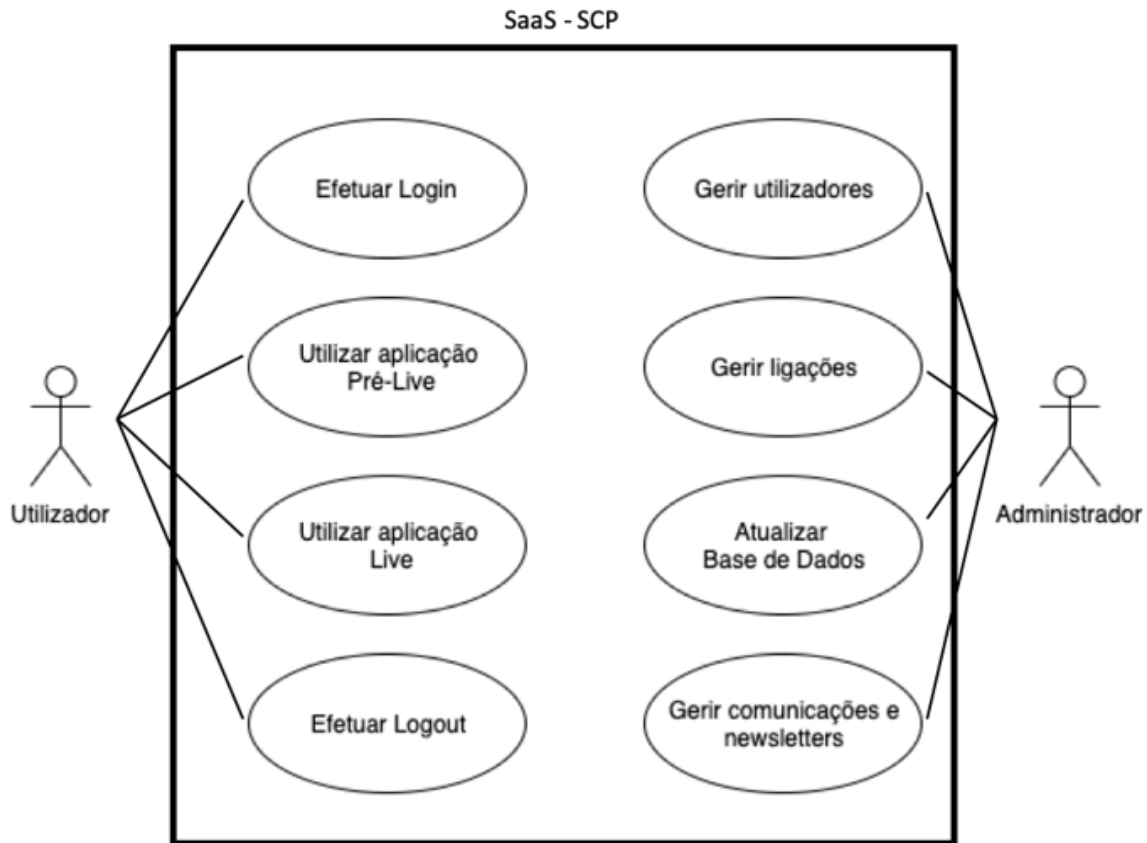


Figura 18: Principais Caso de Uso do Sistema SaaS - SCP

Ao nível do ator *utilizador*, resumem-se os principais casos de uso:

- **Efetuar Login:** Tendo o registo feito, o utilizador deve introduzir o seu *username* e *password*;
- **Utilizar Aplicação Pré-Live:** Se o objetivo do utilizador for obter as *odds* justas para um determinado jogo antes de começar, deve escolher esta aplicação;
- **Utilizar Aplicação Live:** Esta aplicação deve ser utilizada para obter *odds* com base no comportamento do jogo (dados estatísticos);
- **Efetuar Logout:** Se o utilizador pretender, pode terminar a sua sessão.

Ao nível do ator *administrador*, resumem-se os principais casos de uso:

- **Gerir utilizadores:** sendo administrador, tem à sua disposição as ferramentas de ativação e/ou bloqueio de utilizadores;
- **Gerir ligações:** sendo um sistema de gestão de conhecimento, o administrador pode disponibilizar ligações para nós de informação complementares (e.g. informações de atualidade desportiva);
- **Atualizar base de dados:** semanalmente, são introduzidos os jogos realizados. Este processo pode ser manual ou automático com recurso a serviços de terceiros (e.g. consumo de *web services*);
- **Gerir comunicações e newsletters:** como técnica de valor acrescentado para os utilizadores, é enviada todas as semanas uma estratégia/sugestão de apostas com base em pesquisas feitas na base de dados e após aplicação das respetivas heurísticas de análise.

4.3. Requisitos funcionais e não funcionais

Os requisitos funcionais descrevem as operações e atividades que o sistema deve ser capaz de realizar para satisfazer as necessidades dos utilizadores. A Tabela 7 apresenta os principais requisitos funcionais do sistema SaaS – SCP.

Nº	Requisitos funcionais
RF1	O sistema deve permitir fazer o registo e login de novos utilizadores.
RF2	O sistema deve permitir que os utilizadores alterem a sua <i>password</i> e quaisquer outras configurações pessoais.
RF3	O sistema deve permitir obter e processar dados de diferentes fontes, incluindo dados internos e externos.
RF4	O sistema deve ter flexibilidade ao nível da produção de relatórios para atender às necessidades do decisor/cliente.
RF5	O sistema deve dispor de gráficos, linhas de tendência e vistas tabulares.
RF6	O sistema deve realizar análises e comparações.
RF7	O sistema deve permitir pagamentos <i>online</i> .
RF8	O sistema deve efetuar análises estatísticas de resultados.
RF9	O sistema deve auxiliar os administradores a realizarem a gestão de utilizadores e ligações.
RF10	O sistema deve determinar <i>odds</i> para o confronto entre duas equipas.
RF11	O sistema deve permitir fazer as pesquisas à base de dados que o utilizador pretenda.

Tabela 7: Requisitos funcionais do Sistema SCP

A Tabela 8 apresenta os 3 principais requisitos não funcionais do sistema SaaS – SCP.

Nº	Requisitos não funcionais
RNF1	O sistema deve funcionar em vários dispositivos, <i>browsers</i> e sistemas operativos.
RNF2	O sistema deve estar sempre disponível, sendo apenas necessário o acesso à <i>Internet</i> .
RNF3	O sistema deve possuir um tempo de resposta inferior a 5 segundos.

Tabela 8: Requisitos não funcionais do Sistema SCP

5. Esquema Navegacional

Neste capítulo identificam-se os principais aspetos de navegação a implementar no futuro protótipo aplicacional em contexto *cross-platform*.

5.1. Mockup da Aplicação

Descreve-se de seguida uma sequência lógica de *mockups* da aplicação.

Quando um utilizador entra no sistema, tem duas opções: fazer *login*, ou criar a sua conta, conforme ilustra a Figura 19.

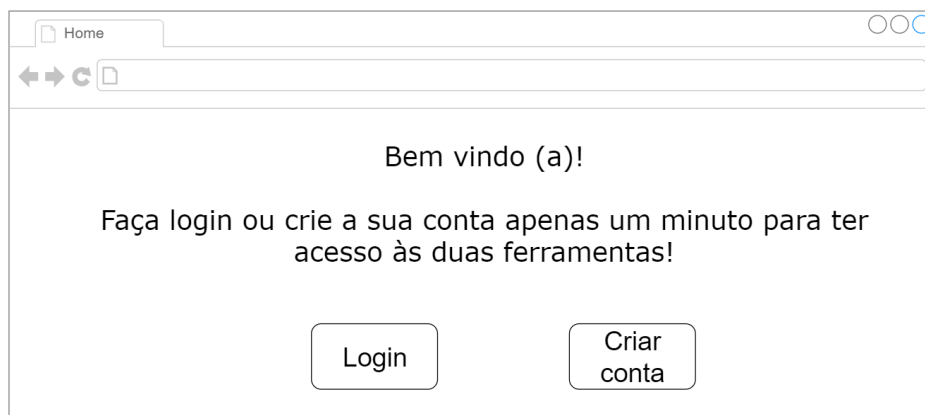


Figura 19: Opções iniciais

Caso o utilizador ainda não tenha uma conta no sistema, é necessário preencher os campos ilustrados na Figura 20.

A imagem mostra uma janela de navegador com o endereço 'Registo' na barra de endereços. O conteúdo principal da página apresenta o texto 'Introduza os seguintes dados para criar a sua conta'. Abaixo deste texto, há seis campos de entrada de texto empilhados verticalmente: 'Primeiro nome', 'Último nome', 'Username', 'Password', 'Email' e 'Data de Nascimento'. Abaixo dos campos, há um botão 'Registrar'.

Figura 20: Criação de conta

Depois do utilizador já ter conta no sistema tem de realizar o *login* através de *username* e *password* conforme ilustrado na figura 21.

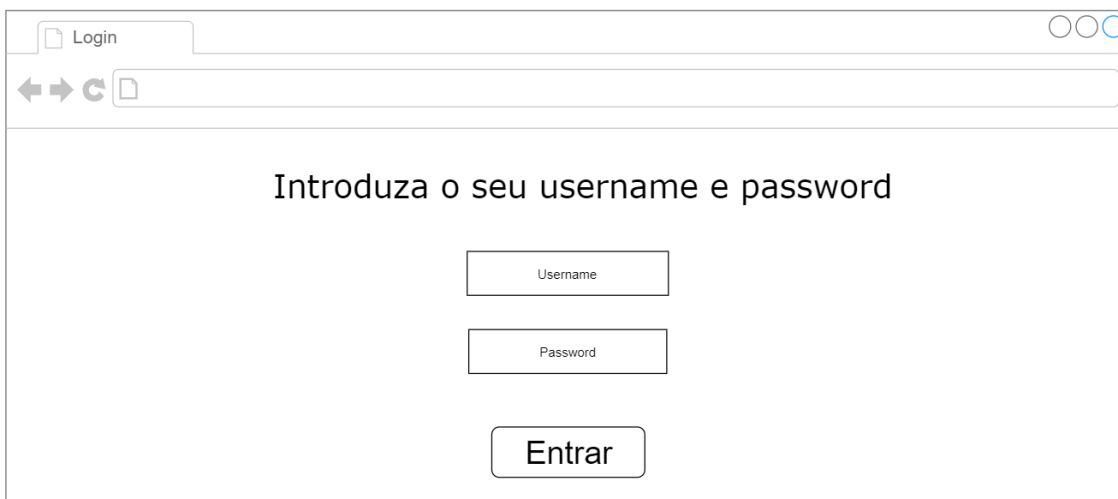
A screenshot of a web browser window showing a login page. The browser's address bar is empty, and the page title is 'Login'. The main content area has a heading 'Introduza o seu username e password'. Below the heading are two text input fields: the first is labeled 'Username' and the second is labeled 'Password'. Below these fields is a button labeled 'Entrar'.

Figura 21: Login

Quando o utilizador já tem o *login* feito, pode escolher qual das duas aplicações quer utilizar: *Pré Live* ou *Live* conforme ilustrado na Figura 22.

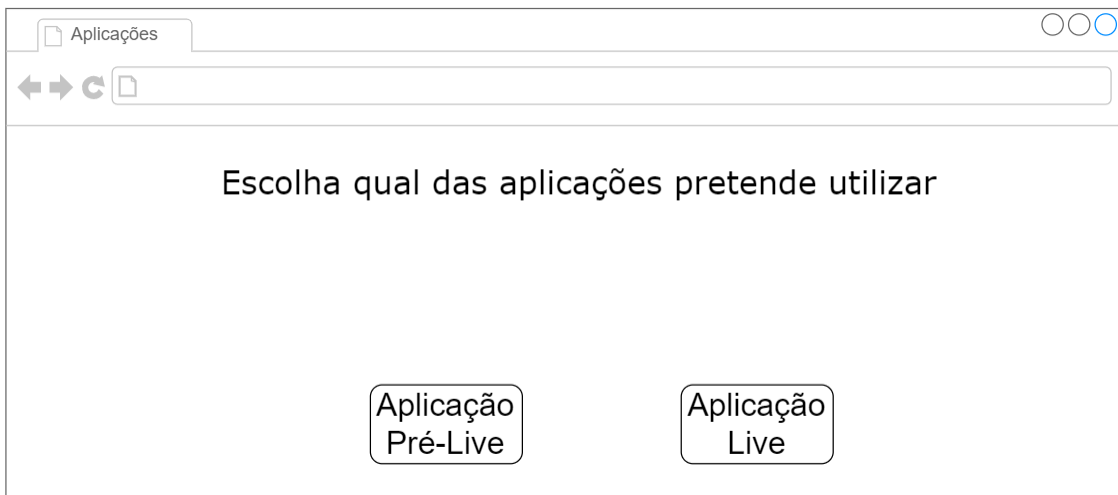
A screenshot of a web browser window showing an application selection page. The browser's address bar is empty, and the page title is 'Aplicações'. The main content area has a heading 'Escolha qual das aplicações pretende utilizar'. Below the heading are two buttons: the first is labeled 'Aplicação Pré-Live' and the second is labeled 'Aplicação Live'.

Figura 22: Escolha da Aplicação

No caso de selecionar a Aplicação Pré Live, só tem de escolher a equipa que joga em casa, a equipa que joga fora de casa e selecionar o botão “Avançar” conforme ilustrado na Figura 23.

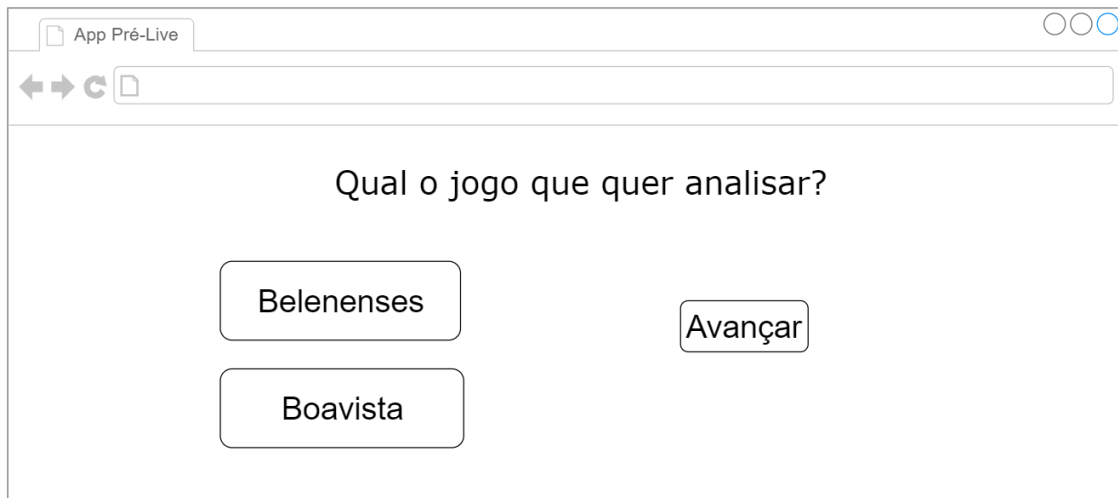


Figura 23: Escolha de jogos na Aplicação Pré Live

Na fase seguinte obtêm-se de imediato uma tabela de análise estatística conforme ilustrado na Figura 24.

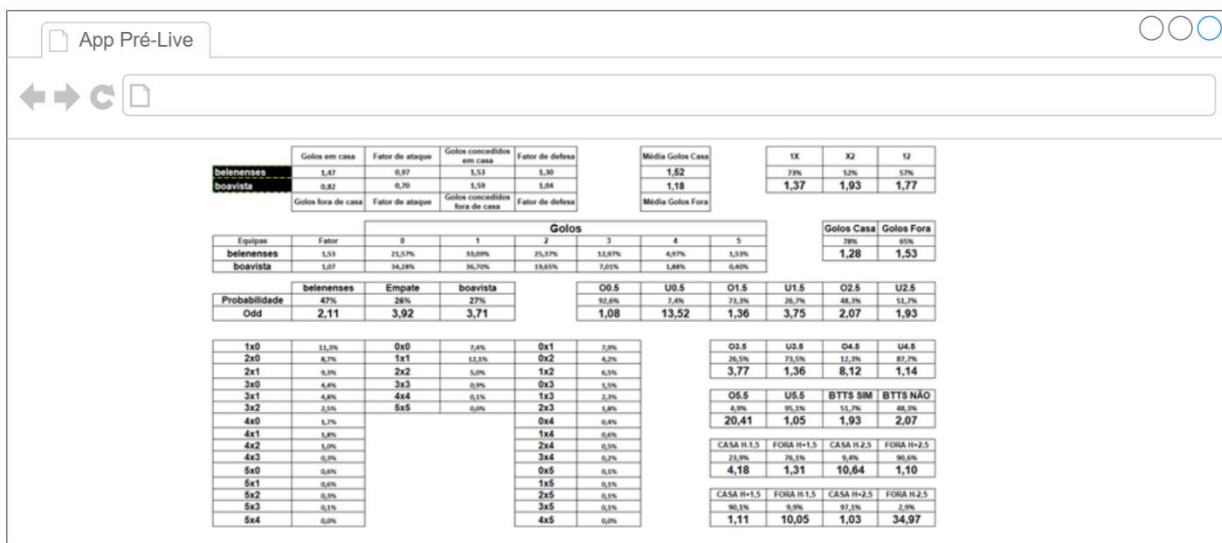


Figura 24: Estatística de um jogo

É possível atribuir probabilidades a alguns tipos de apostas, como: 1x2, dupla hipótese, golos casa e golos fora, *over* e *under* desde o 0.5 até ao 4.5, ambas marcam, *handicap*'s (quer positivos quer negativos) e resultado correto. A Tabela 9 apresenta as definições dos possíveis tipos de aposta que o sistema oferece.

Tipos de Aposta	Definição
1x2 (ou <i>matchodds</i>)	Mercado que determina quem vai ser vencedor do encontro. Se é a equipa da casa (1), fora (2), ou se vai terminar empatado (x).
Dupla hipótese	Mercado que determina dois resultados de três possíveis. 1x para casa ou empate, x2 para empate ou fora e 12 para casa ou fora.
Golos Casa/Fora	Mercado que determina qual a probabilidade da equipa de casa e a equipa de fora para fazer pelo menos um golo.
Over/Under	Mercado que determina quantos golos existem no encontro. No caso do 0.5, o O0.5 representa que vai haver pelo menos um golo de uma das equipas e o U0.5 que não irá haver nenhum.
Ambas marcam	Mercado que determina se as duas equipas marcam (sim), ou se apenas uma ou nenhuma marca (não).
Handicaps Europeus	Mercado que determina qual a diferença de golos entre as equipas. No caso do mercado “casa h-1.5”, a aposta para ser vencedora tem que ter pelo menos dois golos de vantagem.
Resultado correto	Mercado que determina o número de golos que cada equipa marca (exemplo: se fica 0-0, 1-0...).

Tabela 9: Definição de Tipos de Aposta da Aplicação Pré Live

No caso de escolher a Aplicação *Live*, o utilizador tem à sua disposição uma janela para fazer pesquisas para resposta estatística com base no desenvolvimento do jogo (Figura 25). O sistema de pesquisa será traduzido/mapeado automaticamente para SQL.

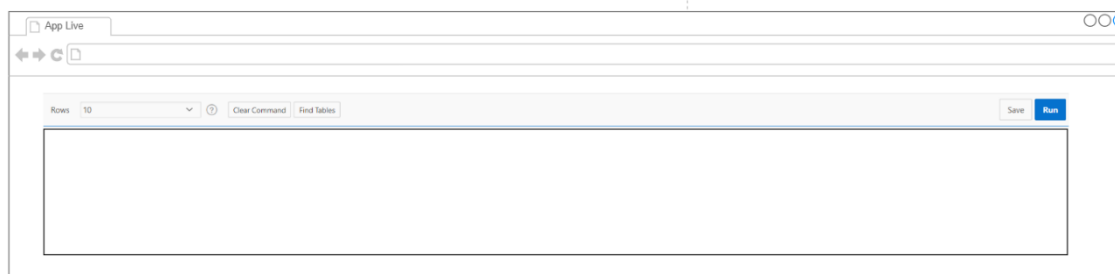


Figura 25: Demonstração da Aplicação Live

Para cada jogo o utilizador tem acesso a diversas estatísticas diferentes, sendo que pode utilizar as estatísticas gerais do tempo regulamentar, ou então apenas obtendo as informações da primeira parte ou segunda.

Também estão associadas a cada jogo *odds* que servem de referência para o decorrer do jogo, existindo as *odds* para a equipa que joga em casa, equipa que joga fora, empate, se vai existir 3 golos ou mais (*odd_over_25*) e 2 golos ou menos (*odd_under_25*).

Para exemplificar heurísticas a realizar nesta fase, considere-se um cenário com mais de 16 mil jogos, estando distribuídos por:

- Top 5 ligas europeias: Espanha, Inglaterra, Alemanha, Itália e França), desde a época 2014/2015 até a época 2018/2019;
- Restantes ligas europeias: 1ª e 2ª divisão de Portugal, 1ª divisão da Bélgica, 1ª divisão da Escócia, 1ª divisão da Grécia, 1ª divisão da Holanda, 1ª divisão da Turquia, 2ª divisão de Espanha, 2ª divisão da Alemanha, 2ª divisão de Itália, 2ª divisão da França), tem as épocas 2017/2018 e 2018/2019.

Então, utilizando a estatística anteriormente demonstrada, testam-se quatro exemplos, utilizando as estatísticas gerais, em cada um, determinando qual a probabilidade sendo uma equipa super favorita ($odd < 1.40$), favorita (entre 1.40 e 1.80), sem favorita (> 1.80) e geral (sem utilizar as *odds* para vencer a partida):

- Qual a probabilidade de um jogo se encontrar 0-0 ao intervalo e sair pelo menos um golo na segunda parte;
- Qual a probabilidade de um jogo se encontrar 1-1 ao intervalo e sair pelo menos um golo na segunda parte;
- Qual a probabilidade de um jogo se encontrar 0-0 ao intervalo, existirem pelo menos quatro remates à baliza e sair pelo menos um golo na segunda parte;
- Qual a probabilidade de um jogo se encontrar 1-1 ao intervalo, sabendo que em *pré live*, a *odd* de saírem pelo menos três golos era de 1.60 (62.5%);

As probabilidades destes exemplos estão representadas na Figura 26.

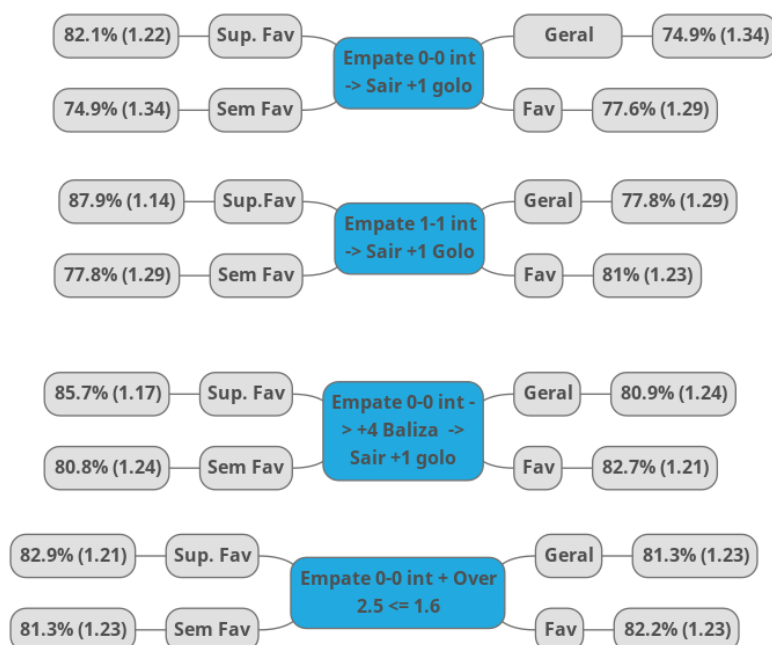


Figura 26: Exemplo probabilístico para um determinado jogo

- ✓ Se o jogo tiver empatado ao intervalo 0-0, a probabilidade de sair um golo no jogo é de 74.9%, mas, se existir um super favorito, essa probabilidade aumenta para 82.1% (+7.2% no caso de uma das equipas ter uma *odd* de vitoria ≤ 1.40);
- ✓ Se chegar empatado ao intervalo 1-1, a probabilidade geral é de 77.8% de sair mais um golo, mas no caso de existir um super favorito passa para 87.9% (+10.1% no caso de uma das equipas ter uma *odd* de vitoria ≤ 1.40);
- ✓ Utilizando agora um comportamento estatístico (existirem 4 remates à baliza na primeira parte), probabilidade de sair um golo no jogo é de 80.9%, essa probabilidade aumenta para 85.7% (+4.8% no caso de uma das equipas ter uma *odd* de vitoria ≤ 1.40);
- ✓ No caso de o jogo chegar empatado ao intervalo 0-0, sabendo que a *odd* de existirem pelo menos 3 golos no jogo é de 81.3%, essa probabilidade aumenta para 82.9% (+1.6% no caso de uma das equipas ter uma *odd* de vitoria de ≤ 1.40).

Considere-se por exemplo a execução SQL com o objetivo de determinar, em todas as ligas disponíveis na base de dados, qual a probabilidade de um jogo ter as duas equipas a fazerem pelo menos golo:

```
select * from total  
  
where (Total_Golos_Casa >=1 and Total_Golos_Fora >=1)
```

Se o resultado desta pesquisa for 8900 jogos, então, a probabilidade de as duas equipas fazerem golo é de:

- Probabilidade de acontecer: 55.6%
- Odd correspondente: 1.80 (esta odd é obtida através da fórmula: $1/\text{probabilidade}$)

5.2. Suporte *Cross-Platform*

O esquema navegacional deve ser realizado com tecnologia *responsive* conforme referido na secção 2.4.5. visto que os dispositivos móveis são cada vez mais utilizados nos dias de hoje.

6. Conclusões

No contexto de apostas em Portugal, identificou-se o problema: “*Definir uma framework SaaS para auxiliar a tomada de decisão de um apostador desportivo com base em dados estatísticos que suporte uma aplicação Pré Live específica para a obtenção de probabilidades antes de começar o jogo, e outra aplicação Live para determinar probabilidades com base na estatística do próprio jogo*”.

Com base na definição do problema, foram definidos os principais objetivos de investigação:

- Efetuar a revisão da literatura com foco nos tópicos: Gestão de Conhecimento, Sistemas de Suporte à Decisão, *Cloud Computing*, Aplicações *Cross-Platform* e Engenharia *Web*. Neste domínio encontra-se a revisão da literatura no Capítulo 2;
- Definir uma *Framework SaaS Multifuncional* para instanciação de aplicações na *cloud*, seja em contexto de apostas desportivas, seja noutros contextos. Relativamente a este objetivo, foi proposta a arquitetura da *Framework* no Capítulo 3. Esta arquitetura foi validada, aceite e publicada pela comunidade científica, tendo sido descritos os principais pacotes do sistema, componentes FSM, e tecnologias de desenvolvimento.
- Definir componentes modulares para suporte a diversos modelos de negócio (incluindo as apostas desportivas). A arquitetura proposta no Capítulo 3 demonstrou a aplicabilidade de um sistema de *plugins* em função do modelo de negócio.
- Propor tecnologias para desenvolvimento de um futuro protótipo aplicacional. Relativamente a este objetivo de investigação, foi proposta no Capítulo 3 a tecnologia de desenvolvimento tendo por base a necessidade de desenvolver um sistema *cross-platform responsive*.
- Definir as relações estáticas entre as principais classes do sistema. Este objetivo de investigação foi concretizado e apresentado no Capítulo 4.
- Especificar casos de uso no contexto específico do problema a resolver. Este objetivo de investigação foi concretizado e apresentado no Capítulo 4.
- Enumerar os principais requisitos funcionais e não-funcionais. Este objetivo de investigação foi concretizado e apresentado no Capítulo 4.
- Propor um esquema navegacional para um futuro protótipo instanciado em paradigma *cross-platform responsive*. Este objetivo de investigação foi concretizado e apresentado no Capítulo 5.
- Demonstrar cenários em que as heurísticas específicas para apostas são exemplificadas. Esta demonstração foi abordada no Capítulo 5 e reforçada através da inclusão de três apêndices no presente documento: 1) Atividade do jogo *online* em Portugal do segundo trimestre de 2019, onde é descrita a análise global, a atividade por categoria de jogos e apostas *online*, qual o número de jogadores registados e monitorização de operadores ilegais de jogo *online*; 2) Designação de termos técnicos sobre as apostas desportivas. E.g atribuir uma probabilidade a um jogo de futebol, *odds* de valor e golos esperados num encontro; 3) Um Exemplo

prático da época 2018/2019 da Liga NOS que visa demonstrar heurísticas a realizar.

Relativamente ao método de investigação adotado (DSR), conclui-se:

- **Fase de Diagnóstico:** identificaram-se os principais problemas e desenvolveram-se hipóteses de trabalho;
- **Fase de Planeamento da Ação:** foram identificados aspetos teóricos que permitem estabelecer metas para a concretização do futuro protótipo;
- **Fase de Empreender Ações:** ao nível dos objetivos de investigação foram empreendidas as ações necessárias, tendo existido *feedback* de investigadores da comunidade científica. No futuro, será necessário realizar o protótipo para dar sequência a esta fase;
- **Fase de Avaliação:** no plano teórico os aspetos centrais do trabalho foram avaliados pela comunidade científica. Após o desenvolvimento do protótipo será necessário visitar esta fase;
- **Fase de Aprendizagem:** Este processo foi contínuo. Todavia, destaca-se que a principal aprendizagem foi a forma de definir uma arquitetura de engenharia de *software* compatível com a instanciação de diferentes modelos de negócio em contexto SaaS. Outras aprendizagens resultaram da análise holística de um sistema de apostas, classes necessárias e respetivos atributos, definição de atores e respetivos papéis, e conceção do sistema de navegação a realizar. A Aplicação *Pré Live* está baseada na estatística e na aplicação da Distribuição de Poisson, onde o utilizador final coloca o nome da equipa da casa e equipa que joga fora e obtém todas as *odds* justas para esse jogo. Se algum tipo de aposta tiver uma *odd* superior à disponibilizada pela casa de apostas, essa aposta é uma aposta de valor e no longo prazo vai trazer um ROI positivo, porque o utilizador está a apostar em algo que tem uma maior probabilidade do que a casa de apostas paga. A Aplicação *Live* tem como foco o comportamento do jogo (estatísticas em tempo real), fazendo pesquisas constantes para verificar se há apostas de valor. Pressupondo que quem aposta em live está a ver o jogo, pode entender se as estatísticas correspondem à realidade do jogo ou se são enganadoras, trazendo assim uma vantagem enorme em comparação às casas de apostas, que apenas tratam a estatística pura e as *odds* só variam devido a dois fatores: volume de dinheiro e tempo de jogo. Assim, com base nestas duas aplicações, quem as utilizar, tem um suporte estatístico para tomar a sua decisão de fazer uma aposta a uma determinada *odd*, e não utilizar termos como “eu acho que vai ganhar” ou “é impossível esta equipa não fazer pelo menos um golo”, porque infelizmente a grande maioria dos apostadores pensam desta forma sem fazerem um estudo sobre a *odd* apresentada, muito menos se essa *odd* tem valor no longo prazo.

Com base na definição do problema e nos objetivos de investigação definidos, considera-se que os mesmos foram alcançados. O trabalho futuro consiste realizar um protótipo aplicacional com base na arquitetura definida e heurísticas estatísticas/probabilísticas demonstradas neste documento.

Bibliografia

AF Ragab, M., & Arisha, A. (2013). Knowledge management and measurement: a critical review. *Journal of knowledge management*, 17(6), 873-901.

Alyoubi, B. A. (2015). Decision support system and knowledge-based strategic management. *Procedia Computer Science*, 65, 278-284.

Ambavane, S. A., Pawar, A. S., Verma, V. H., & Marathe, P. (2018). *Cloud Computing and SaaS (ERP) Implementation*.

Bâra, A., & Lungu, I. (2012). Improving decision support systems with data mining techniques.

Bolchini, D. (2000). *Web Design Patterns: improving quality and performance in Web Application design*. Universit della Svizzera Italiana–USI, Communication Sciences–Thesis. Lugano.

Felsberger, A., Oberegger, B., & Reiner, G. (2016, October). A Review of Decision Support Systems for Manufacturing Systems. In *SAMI@ iKNOW*.

H. Ferreira, A. Xavier, M. Romão and A. Barão, "Multifunctional SaaS Framework," 2019 14th Iberian Conference on Information Systems and Technologies (CISTI), Coimbra, Portugal, 2019, pp. 1-5.

Jain, A., & Kumar, R. (2014). A taxonomy of cloud computing. *International journal of scientific and research publications*, 4(7), 1-5.

Kappel, G., Pröll, B., Reich, S., & Retschitzegger, W. (2006). *Web engineering*. New York: Wiley.

King, W. R. (2009). Knowledge management and organizational learning. In *Knowledge management and organizational learning* (pp. 3-13). Springer, Boston, MA.

More, K. A., & Chandran, M. P. (2016). Native vs hybrid apps. *Proceeding of International Journal of Current Trends in Engineering & Research*, 563-572.

Noor, T. H., Zeadally, S., Alfazi, A., & Sheng, Q. Z. (2018). Mobile cloud computing: Challenges and future research directions. *Journal of Network and Computer Applications*, 115, 70-85.

Rani, B. K., Rani, B. P., & Babu, A. V. (2015). Cloud computing and inter-clouds–types, topologies and research issues. *Procedia Computer Science*, 50, 24-29.

Padhy, R. P., Patra, M. R., & Satapathy, S. C. (2011). Cloud computing: security issues and research challenges. *International Journal of Computer Science and Information Technology & Security (IJCSITS)*, 1(2), 136-146.

Palmieri, M., Singh, I., & Cicchetti, A. (2012, October). Comparison of cross-platform mobile development tools. In *2012 16th International Conference on Intelligence in Next Generation Networks* (pp. 179-186). IEEE

Phillips-Wren, G., Mora, M., Forgionne, G. A., & Gupta, J. N. (2009). An integrative evaluation framework for intelligent decision support systems. *European Journal of Operational Research*, 195(3), 642-652.

Popa, M. (2013). Considerations regarding the cross-platform mobile application development process. *Academy of Economic Studies. Economy Informatics*, 13(1), 40.

Pressman, R. S. (2005). *Software engineering: a practitioner's approach*. Palgrave Macmillan.

SRIJ - Serviço de Regulação Inspeção de Jogos (2019). Dados relativos à atividade de jogo online no ano de 2019. Retrieved from <https://www.srij.turismodeportugal.pt/pt/publicacoes-e-estatisticas/estatisticas/>

Tripathi, K. P. (2011). Decision support system is a tool for making better decisions in the organization. *Indian Journal of Computer Science and Engineering (IJCSE)*, 2(1), 112-117.

Wang, G. F., Tian, X. T., Geng, J. H., & Guo, B. (2013). A process innovation knowledge management framework and its application. In *Advanced Materials Research (Vol. 655, pp. 2299-2306)*. Trans Tech Publications.

Xanthopoulos, S., & Xinogalos, S. (2013, September). A comparative analysis of cross-platform development approaches for mobile applications. In *Proceedings of the 6th Balkan Conference in Informatics (pp. 213-220)*. ACM.

Zaied, A. N. H., Hussein, G. S., & Hassan, M. M. (2012). The role of knowledge management in enhancing organizational performance. *International Journal of Information Engineering and Electronic Business*, 4(5), 27.

Anexo

Framework SaaS Multifuncional

Multifunctional SaaS Framework

Hugo Ferreira, André Xavier, Marta Romão,
Alexandre Barão

Atlântica - School of Management Sciences,
Health, IT & Engineering

Fábrica da Pólvora de Barcarena, Barcarena, Portugal

20111471@academia.uatlantica.pt,

20141910@academia.uatlantica.pt,

201728783@academia.uatlantica.pt, abarao@uatlantica.pt

Resumo — A Framework SaaS Multifuncional (FSM) é um projeto de investigação cujo objetivo principal é permitir a instanciação de aplicações multiplataforma SaaS para suportar necessidades de clientes em função de diferentes contextos e áreas de negócio. Este artigo apresenta a arquitetura de pacotes FSM e respetivas relações entre os principais componentes de *software*. A FSM é desenvolvida com recurso à tecnologia Laravel. São ainda apresentadas três áreas para cenários de instanciação, de modo a permitir a validação de plataformas SaaS desenvolvidas com base na FSM: Sistemas de Gestão de Conhecimento; Sistemas de Suporte à Decisão; e Sistemas de Gestão de Eventos.

Palavras Chave – SaaS; FSM; Laravel; Sistemas de Gestão de Conhecimento; Sistemas de Suporte à Decisão; Sistemas de Gestão de Eventos.

Abstract — The Multifunctional SaaS Framework (from portuguese FSM) is a research project in which the main goal is to develop multi-platform SaaS applications in the cloud to support customer needs based on different contexts and business areas. This paper presents the FSM packages architecture and relationships among the main software components. The FSM is developed using Laravel technology. Moreover, three instantiation and validation scenarios for SaaS platforms based on the FSM are presented: Knowledge Management Systems; Decision Support Systems; and Event Management Systems.

Keywords - SaaS; FSM; Laravel; Knowledge Management Systems; Decision Support Systems; Event Management Systems.

I. INTRODUÇÃO

O conceito *cloud computing* pressupõe a utilização de modelos capazes de suportar processamento e armazenamento remoto [1]. Este conceito baseia-se em cinco princípios básicos [2]:

- Partilha de recursos;
- Escalabilidade;
- Elasticidade;
- Pagamento por recursos utilizados;

- Autoaprovisionamento de recursos.

Ao nível da *partilha de recursos*, estes podem ser reservados especificamente por cada cliente sendo também possível atribuir os mesmos de uma forma dinâmica, conseguindo desta forma uma utilização mais eficiente do *hardware*, permitindo assim mais clientes por servidor físico.

A *escalabilidade* pressupõe que uma solução baseada na nuvem liberta o cliente de preocupações ao nível dos recursos. E.g. a alocação de recursos remotos é garantida pelo fornecedor de *software* e serviços ao qual cabe a responsabilidade de garantir a disponibilidade de recursos contratada e se necessário, adicionar novos recursos para aumentar a escala do negócio de uma forma massiva.

A *elasticidade* permite que as necessidades do serviço/cliente possam ser dinâmicas. I.e., as mesmas podem aumentar rapidamente sendo necessário disponibilizar novos recursos apenas no momento em que são necessários, ou ser reduzidas se as mesmas deixarem de ser necessárias, baixando assim custos de operação.

O *pagamento por recursos utilizados*, significa que os clientes pagam apenas os recursos que estão a consumir no momento, não existindo necessidade de investimentos adicionais para suportar eventuais picos de utilização. Ou seja, esta solução permite que o cliente tenha inicialmente um custo mais baixo para consumo de menos recursos, e posteriormente um custo mais elevado se a necessidade de recursos também aumentar.

O *autoaprovisionamento de recursos* permite que o cliente adicione e faça a gestão de recursos do seu sistema sem a intervenção do fornecedor de serviços.

Apesar das vantagens dos cinco princípios básicos referidos, existem ameaças em relação aos serviços que são disponibilizados e expostos na nuvem. Em concreto, por ordem de gravidade [3]:

- Divulgação de dados;
- Perda de dados;

- Furto de contas;
- APIs inseguras;
- Negação de serviço;
- Colaboradores mal-intencionados;
- Abuso dos serviços na nuvem;
- Problemas relacionados com tecnologia partilhada.

O conceito SaaS (*Software as a Service*), pressupõe que o cliente utilize *software* com recurso à Internet pagando um valor pelo serviço que é disponibilizado pelo fornecedor. Nas soluções SaaS, o cliente não necessita de comprar *hardware* adicional e *software*, uma vez que estas aplicações *web* são colocadas em execução na infraestrutura do fornecedor de serviços [4].

Um dos desafios de desenvolvimento das aplicações *web* consiste em garantir que as mesmas sejam visualizadas de forma correta e adaptadas ao dispositivo do cliente. Assim, o termo *responsive* significa que na perspectiva de *web design* as peças de *software* permitem que os *layouts* das aplicações se adaptem à resolução do dispositivo cliente, e.g. *smartphone*, *tablet* ou *desktop*. Neste domínio, a tecnologia HTML5/CSS3 permite que a visualização de conteúdos de aplicações *web* seja *responsive*. O W3C especifica os standards HTML e CSS [5].

Por outro lado, é importante garantir que as aplicações SaaS disponham de ferramentas ETL (*Extract, Transform and Load*). O objetivo destas ferramentas de *software* é extrair dados de diversos sistemas, efetuar a sua transformação de acordo com regras de negócio e por fim carregar os mesmos para o sistema de destino [6].

Na Secção II é apresentado o conceito da Framework SaaS Multifuncional (FSM). A arquitetura FSM é descrita na Secção III. A tecnologia de desenvolvimento adotada é apresentada na Secção IV. Os cenários de validação são descritos na Secção V. Finalmente, na Secção VI, são apresentadas as conclusões preliminares de acordo com o principal desafio de investigação.

II. A FRAMEWORK SAAS MULTIFUNCIONAL

A Framework SaaS Multifuncional é um projeto em curso no qual o objetivo principal de investigação é permitir a instanciação na nuvem de *aplicações multiplataforma* SaaS para suportar diferentes necessidades de clientes em função de diferentes contextos e áreas de negócio.

Num cenário em que a utilização de dispositivos móveis prolifera, as *aplicações multiplataforma* são uma solução para que as mesmas sejam executadas e visualizadas corretamente independentemente do sistema operativo do cliente.

Assim, o objetivo é escrever o código-fonte da FSM, de modo as aplicações possam ser executadas em diferentes sistemas operativos e dispositivos do cliente.

Dispondo de uma infraestrutura comum, as heurísticas de áreas de negócio específicas são desenvolvidas incrementalmente através de peças de *software* modulares (*plugins*).

III. ARQUITETURA FSM

A arquitetura FSM baseia-se em dois pacotes principais:

- *SAAS Services*;
- *SAAS Client*.

A Fig. 1 ilustra a relação entre estes pacotes.

O pacote *SAAS Client* utiliza o pacote *SAAS Services* para executar um conjunto de serviços que garantem funcionalidades como a configuração dos *plugins* necessários, gestão de utilizadores do sistema, personalização de logótipos do cliente, esquema de cores da aplicação e configuração de funcionalidades Web 2.0.

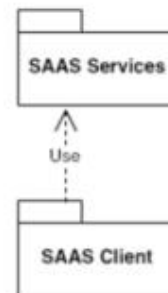


Figure 1. Pacotes da FSM

A Fig. 2 descreve a visão geral da arquitetura de componentes FSM. *Responsive* representa uma aplicação *web responsive* que utiliza o sistema *SAAS Services*. Este sistema inclui uma biblioteca e uma base de dados. Os algoritmos de acesso à base de dados (*SAAS Database*) são implementados no componente *Library* do sistema *SAAS Services*.

O sistema *SAAS Services* disponibiliza ainda dois tipos de interfaces: (1) interfaces para suportar aplicações *web (Responsive)*, e (2) interfaces para suportar integração externa de dados (*ETL Engine*).

Como ilustrado na Fig. 2, o componente *SAAS Plugins* oferece uma interface para disponibilizar o acesso aos *plugins* KM (*Knowledge Management*), DSS (*Decision Support System*) e EM (*Event Management*). Estes *plugins* são responsáveis pela implementação de heurísticas específicas para aplicações SaaS em contexto de Sistemas de Gestão de Conhecimento, Sistemas de Suporte à Decisão, e Sistemas de Gestão de Eventos respetivamente.

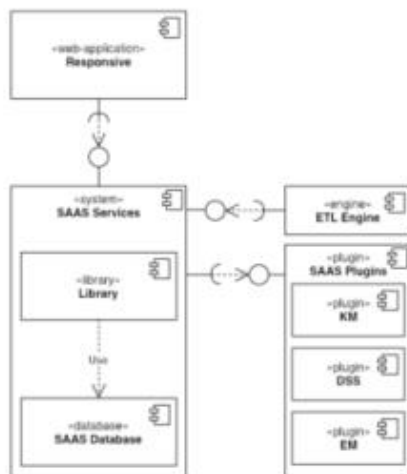


Figure 2. Arquitetura de componentes FSM

IV. TECNOLOGIA DE DESENVOLVIMENTO

A FSM é desenvolvida com tecnologia Laravel [7]. Laravel é uma *framework* que utiliza o padrão MVC (*Model-View-Controller*) em PHP, e que permite o desenvolvimento e reutilização modular de código. E.g. Laravel é composta por mais de vinte pacotes de bibliotecas e *frameworks* diferentes que são geridas por um gestor de dependências, o *composer*. Possui ainda ferramentas para teste, definição de mecanismos de autenticação, construção de *queries*, compatibilidade com várias bases de dados e migração de dados [8].

A Fig.3 (adaptada) ilustra as relações dos principais módulos da *framework* Laravel [8]. O módulo *Client* representa um *Web Browser* ou um *cliente API*. O módulo *Controller* suporta as principais operações lógicas do sistema (e.g. gestão de encaminhamentos, controladores e recursos). O módulo *Model* suporta as ligações e respetivos mapeamentos de atributos da base de dados. O módulo *Data Source* representa a base de dados que pode ser do tipo PostgreSQL, MongoDB ou MySQL. O módulo *View* utiliza *templates* PHP e é o responsável pelo tratamento visual dos dados, através do *Blade*, um motor de processamento de modelos HTML. Finalmente, na Fig.3 as setas sólidas representam *pedidos*, enquanto que as setas tracejadas representam *respostas*.

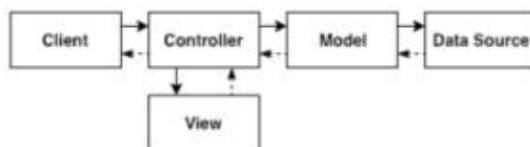


Figure 3. Módulos da Framework Laravel (Adaptada)

V. CENÁRIOS DE VALIDAÇÃO

Para validar a instanciação de plataformas SaaS com base na FSM, estão em curso três projetos de investigação que permitem a validação de três cenários: (A) Sistemas de Gestão de Conhecimento; (B) Sistemas de Suporte à Decisão; e (C) Sistemas de Gestão de Eventos. A utilização de recursos comuns é *per se* um benefício desta solução. Descrevem-se de seguida os principais desafios de investigação dos três cenários referidos.

A. Sistemas de Gestão de Conhecimento

Os Sistemas de Gestão de Conhecimento (SGC) são aplicações dos Sistemas de Comunicação e Informação (SCI), baseados na computação da organização para dar suporte aos vários processos de Gestão de Conhecimento (GC). Normalmente estes sistemas não são tecnologicamente distintos dos SCI, mas envolvem bases de dados, como repositórios para lições aprendidas, diretórios e redes para permitir que os participantes da organização estejam em contato com especialistas reconhecidos em várias áreas [9].

Com base na metodologia de investigação *Action-Research* [10], o principal objetivo é investigar e definir os principais fatores de resistência existentes na mudança em IT. Em concreto, compreender melhor a resistência à mudança por parte dos *key users/end users*. Para tal, será utilizada a FSM para instanciação da plataforma SaaS que suportará as seguintes atividades de investigação:

- Enquadrar fatores que influenciam a resistência à mudança;
- Definir práticas aplicadas na implementação de mudanças;
- Entrevistar responsáveis de áreas de IT para analisar quais os fatores que levam a resistência à mudança quando aprovam a implementação de mudanças ao nível de infraestruturas ou *software*;
- Questionar técnicos de infraestrutura e desenvolvimento sobre as barreiras com as quais se deparam na implementação de mudanças;
- Analisar questionários e apurar junto dos *key users/end users* o que consideram ser barreiras nos processos de mudanças em IT;
- Divulgar um conjunto de boas práticas a ter em consideração para diminuir a resistência à mudança em IT;
- Testar alternativas com base nos resultados obtidos;
- Partilhar conclusões.

B. Sistemas de Suporte à Decisão

O objetivo principal de um Sistema de Suporte à Decisão é apoiar a tomada de decisão, em ambientes complexos e como tal sujeitos a fenómenos emergentes [11]. Com base na

metodologia de investigação *Design Science Research* [12], o objetivo é recorrer à FSM para efetuar a instanciação de uma plataforma *Online Betting* que concretize as seguintes funcionalidades que são comuns a sistemas de suporte à decisão [13]:

- Lidar com grandes quantidades de dados, e suportar as respetivas pesquisas a bases de dados;
- Obter e processar dados de diferentes fontes, incluindo dados internos e externos;
- Ter flexibilidade ao nível da produção de relatórios para atender às necessidades do decisor/cliente;
- Dispor de gráficos, linhas de tendência e vistas tabulares;
- Realizar análises e comparações;
- Suportar abordagens heurísticas, em função das necessidades do decisor.

Para concretizar estes objetivos de investigação, será utilizada a FSM para desenvolvimento da plataforma SaaS com o objetivo de desenvolver as principais funcionalidades para:

- Gestão de utilizadores;
- Aconselhamento com base nas heurísticas desenvolvidas para o modelo de negócio;
- Gestão de pagamentos *online*;
- Análise estatística de resultados;
- *Mashup* de mecanismos de visualização de dados.

C. Sistemas de Gestão de Eventos

Os Sistemas de Gestão de Eventos devem estar preparados para gerir diferentes tipos de ocorrências sociais. E.g. reuniões familiares ou corporativas, conferências, exposições, eventos políticos, eventos desportivos, entre outros [14].

Com base na metodologia de investigação *Design Science Research* [12], o objetivo é instanciar uma plataforma SaaS para integrar aspetos essenciais do funcionamento de empresas de *catering*, especialmente dedicadas à área de eventos sociais. Para tal, será utilizada a FSM para desenvolvimento da referida plataforma com o objetivo de suportar as seguintes tarefas:

- Gestão de eventos;
- Gestão de clientes/fornecedores;
- Gestão de *stocks e faturação*;
- Gestão documental;
- *Mashup* de mecanismos *Social Media*.

VI. CONCLUSÕES

O objetivo principal de investigação consiste no desenvolvimento de uma *framework* que permita a instanciação

na nuvem de *aplicações multiplataforma SaaS* para suportar diferentes necessidades de clientes em função de diferentes contextos e áreas de negócio. Com base na arquitetura de pacotes e componentes FSM, e com recurso à tecnologia Laravel, a *Framework SaaS Multifuncional* reúne características técnicas que permitem o desenvolvimento modular de peças de *software* para responder ao principal desafio de investigação. Para validar a FSM com base em cenários reais, estão em curso três projetos de investigação com o objetivo de instanciar e testar plataformas SaaS nas áreas de Sistemas de Gestão de Conhecimento, Sistemas de Suporte à Decisão e Sistemas de Gestão de Eventos.

Dada a modularidade dos componentes da arquitetura proposta e com base na tecnologia adotada, considera-se que a FSM reúne condições para provar a sua aplicabilidade em cenários reais. Assim, como contribuição expectável, pretende-se disponibilizar no futuro os componentes de *software SAAS Services* com o objetivo de permitir à comunidade académica o desenvolvimento e validação de novos *plugins*.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] NIST. (2013). The NIST Definition of Cloud Computing. Consultado em 4 de fevereiro de 2019: <http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf>.
- [2] Mather, T., Kumaraswamy, S., & Latif S. (2009). *Cloud Security and Privacy*, 1ª Edição. California: O'Reilly Media Inc.
- [3] Cloud Security Alliance. (2013). The Notorious Nine – Cloud Computing Threats in 2013. Consultado em 4 de fevereiro de 2019: https://downloads.cloudsecurityalliance.org/initiatives/top_threats/The_Notorious_Nine_Cloud_Computing_Top_Threats_in_2013.pdf.
- [4] Xin, M., & Levina, N. (2008). *Software-as-a-Service Model: Elaborating Client-Side Adoption Factors*. Paris, France: Proceedings of the 29th International Conference on Information Systems, R. Boland, M. Limayem, B. Pentland, (eds).
- [5] W3C – HTML and CSS. Consultado em 4 de fevereiro de 2019: <https://www.w3.org/standards/webdesign/htmlcss>.
- [6] What Is ETL? | SAS. Consultado em 4 de fevereiro de 2019: https://www.sas.com/en_us/insights/data-management/what-is-etl.html.
- [7] Laravel - The PHP Framework For Web Artisans. Consultado em 4 de fevereiro de 2019: <https://laravel.com/>.
- [8] Saunier, R. (2014). *Getting Started with Laravel 4*. Birmingham, UK: Packt Publishing.
- [9] Khan, S. R. (2018). Knowledge Creation And Knowledge Management. *Journal of Management Science, Operations & Strategies* (ISSN 2456-9305), 2(1), 27-31.
- [10] E. Trist (1976) *Engaging with Large-scale Systems*. In *Experimenting with Organizational Life: The Action Research Approach*. New York, pp. 43-75.
- [11] Mohamad, R., et al. (2010). Mohamad, R., Hamdan, A. R., Othman, Z. A., & Noor, N. M. M. Decision support systems (dss) in construction tendering processes. *arXiv preprint arXiv:1004.3260*.
- [12] Alan R. Hevner, Salvatore T. March, Jinsoo Park, and Sudha Ram. 2004. Design science in information systems research. *MIS Q.* 28, 1 (March 2004), 75-105.
- [13] Tripathi, K. P. (2011). Decision support system is a tool for making better decisions in the organization. *Indian Journal of Computer Science and Engineering (IJCSSE)*, 2(1), 112-117.
- [14] Raj, R. and Musgrave, J. (2009). The economics of sustainable events. In Raj, Razaq & Musgrave, James (Ed.). *Event Management and Sustainability* (pp. 56-65). Oxfordshire: Leeds Metropolitan University.

Apêndice

Apêndice A - Atividade do Jogo *Online* em Portugal

Em apêndice está apresentada a informação recolhida através do website do Serviço de Regulação e Inspeção de Jogos (SRIJ), onde é possível ter acesso aos dados relativos à atividade de jogo *online* no ano de 2019, mais especificamente do 2º trimestre de 2019.

Análise Global

Em 30 de junho de 2019, conforme ilustrado na Figura 29, 11 entidades estavam autorizadas a exercer a atividade de exploração de jogos e apostas *online* em Portugal, mais 3 face ao mesmo período de 2018. No seu conjunto, aquelas entidades são detentoras de 18 licenças (8 licenças para exploração de apostas desportivas à cota e 10 licenças para exploração de jogos de fortuna ou azar). Durante o 2º trimestre de 2019, a atividade de jogos e apostas *online* gerou cerca de 48,3 milhões de euros de receita bruta¹, valor superior em 11 milhões de euros (29,5%) comparativamente ao período homólogo de 2018.

O valor total do Imposto Especial de Jogo *Online* (IEJO) ascendeu, no 2º trimestre de 2019, a 21 milhões de euros, superior em cerca de 4 milhões de euros (23,9%) ao apurado para o mesmo período de 2018.

Atividade de Jogo <i>Online</i> - Resumo			
	Valor trimestral		Variação
	2T 2018	2T 2019	
	(Nº)		
Entidades Exploradoras ¹	8	11	3
Licenças atribuídas ²	13	18	5
Apostas Desportivas à Cota	6	8	2
Jogos de Fortuna ou Azar	7	10	3
	(M €)		(%)
Receita Bruta	37,3	48,3	29,5%
Apostas Desportivas à Cota	20,5	23,0	12,4%
Jogos de Fortuna ou Azar	16,8	25,3	50,3%
IEJO	16,9	21,0	23,9%
	(Milhares)		(%)
Novos Registos de Jogadores	103,9	102,2	-1,6%
Jogadores com Prática de Jogo	255,2	297,0	16,4%
Jogadores Autoexcluídos	24,6	38,6	57,2%

Figura 27: Quadro Resumo da Atividade do Jogo *Online* (Fonte: SRIJ)

Análise da atividade por categoria de jogos e apostas *online*

No 2º trimestre de 2019, conforme ilustrado na Figura 30, a receita bruta criada pela atividade de jogos e apostas *online* ascendeu a 48,3 milhões de euros, valor mais alto desde o 1º trimestre de 2017.

A receita bruta resultado das apostas em jogos de fortuna ou azar representou 52,3% do total da receita bruta dos jogos e apostas *online*, enquanto a receita bruta criada pelas apostas desportivas à cota foi 47,7% (correspondendo a 25,3 e a 23,0 milhões de euros, respetivamente).

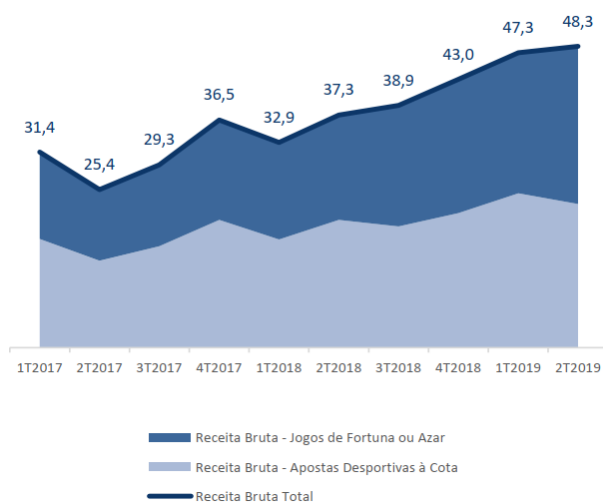


Figura 28: Receita Bruta dos Jogos e Apostas Online, Evolução 1º T 2017 - 2º T 2019 (M €) (Fonte: SRIJ)

No 2º trimestre de 2019, conforme ilustrado na Figura 31, a receita bruta das entidades exploradoras de apostas desportivas à cota *online* atingiu o valor de 23,0 milhões de euros, registando-se um aumento de 2,5 milhões de euros face a igual período de 2018.

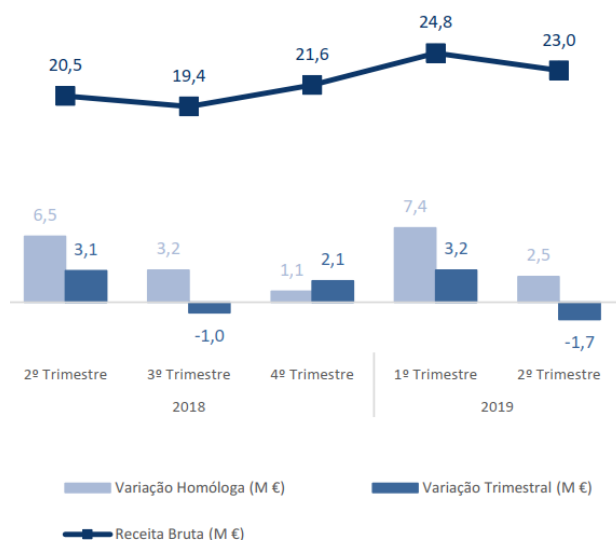


Figura 29: Receita Bruta: Apostas Desportivas à Cota, Evolução 2º T 2018 - 2º T 2019 (M €) (Fonte: SRIJ)

Conforme ilustrado na Figura 32, o valor em apostas desportivas à cota, no período em referência, de aproximadamente 112 milhões de euros, superior em 22,3 milhões face ao registado no mesmo período de 2018.

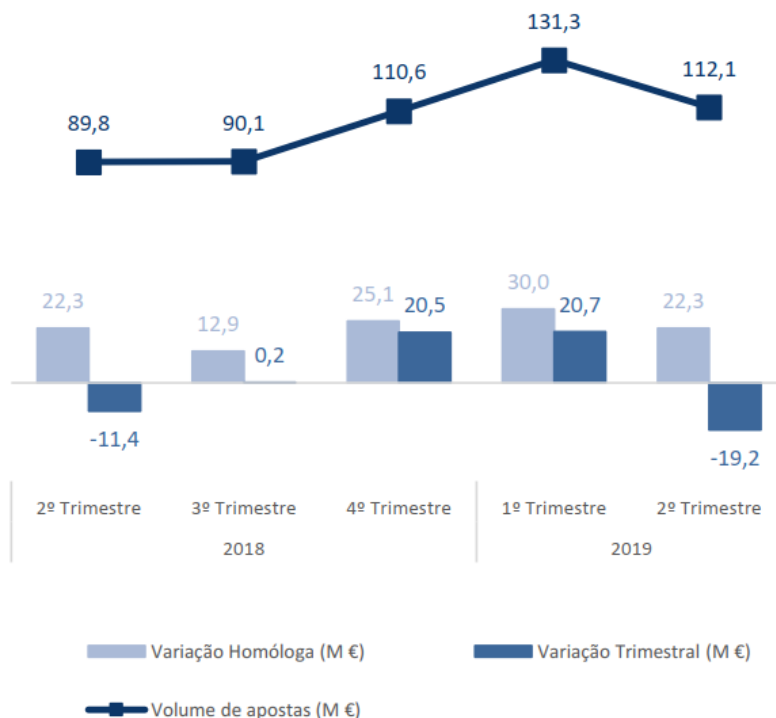


Figura 30: Volume de Apostas Desportivas à Cota, Evolução 2º T 2018 - 2º T 2019 (M €) (Fonte: SRIJ)

A Figura 33 demonstra que o Futebol continuou a ser a modalidade desportiva onde se regista o maior volume de apostas, representando 71,41% do total de apostas desportivas efetuadas. As apostas nas modalidades de Ténis e Basquetebol, representaram cerca de 24% do total das apostas desportivas (16,38% e 7,5%, respetivamente).

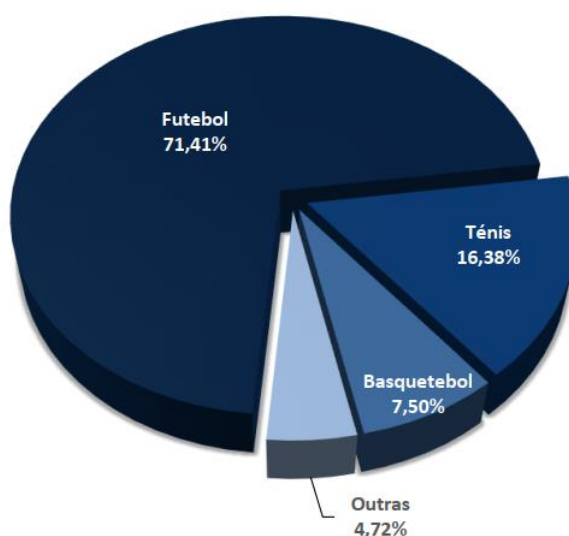


Figura 31: Apostas Desportivas à Cota por Modalidade (2º T 2019) (Fonte: SRIJ)

Ao nível das competições desportivas, conforme ilustrado na Figura 34, a Primeira Liga portuguesa representou quase 10% do volume de apostas efetuadas na modalidade de Futebol, seguida da La Liga espanhola, da Premier League inglesa, da Serie A italiana e da UEFA Champions League, que, no seu conjunto correspondem, aproximadamente, a cerca de 21% do total. No Ténis, a segunda modalidade com maior volume de apostas no segundo trimestre de 2019, a competição Roland Garros foi responsável pela maior percentagem das respetivas apostas (i.e., 19,4%). A competição norte-americana NBA representou mais de 45% do volume total de apostas na modalidade Basquetebol.

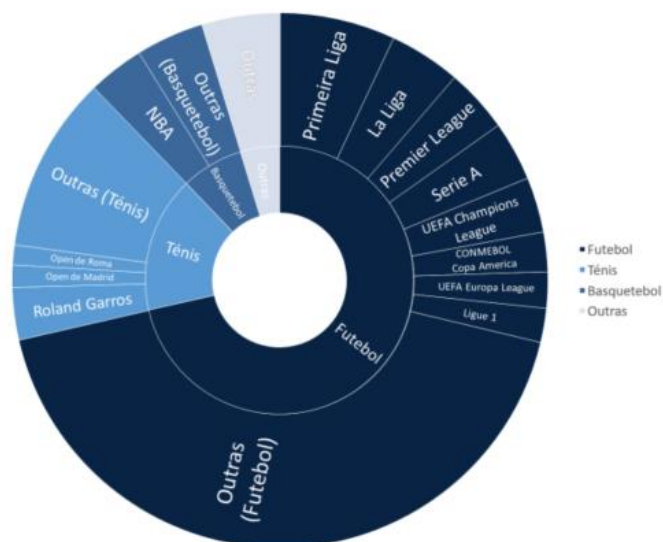


Figura 32: Apostas Desportivas à Cota por Competição (2º T 2019) (Fonte: SRIJ)

Jogadores registados

No 2º trimestre de 2019, e no conjunto das 11 entidades exploradoras, apuraram-se 102,2 mil novos registos de jogadores, registando-se um decréscimo de 1,6% face ao registado em igual período do ano anterior (103,9 mil novos registos de jogadores). A Figura 35 ilustra estes aspetos.

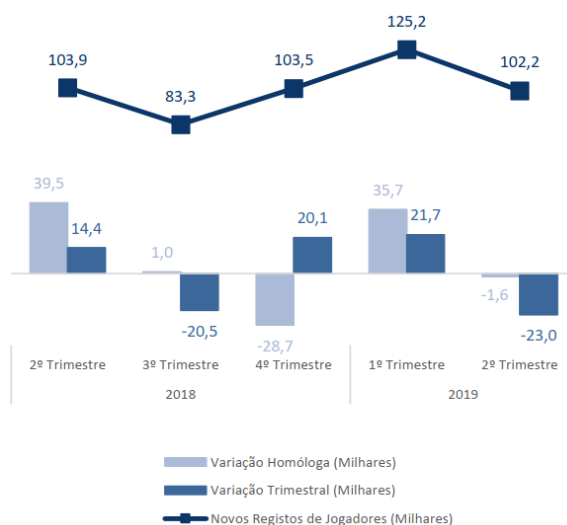


Figura 33: Jogadores - Evolução do número de novos registos, 2º T 2018 - 2º T 2019 (Fonte: SRIJ)

Conforme ilustrado na Figura 36, em 30 de junho de 2019, os jogadores com idades compreendidas entre os 25 e 44 anos representavam 62,1% do total de jogadores registados. Na mesma data, os indivíduos entre os 18 e os 24 anos representavam 24,4% do total de jogadores registados.

Relativamente aos novos registos ocorridos no trimestre em análise, quase 65% dos mesmos reportaram-se a jogadores com idade inferior a 35 anos.

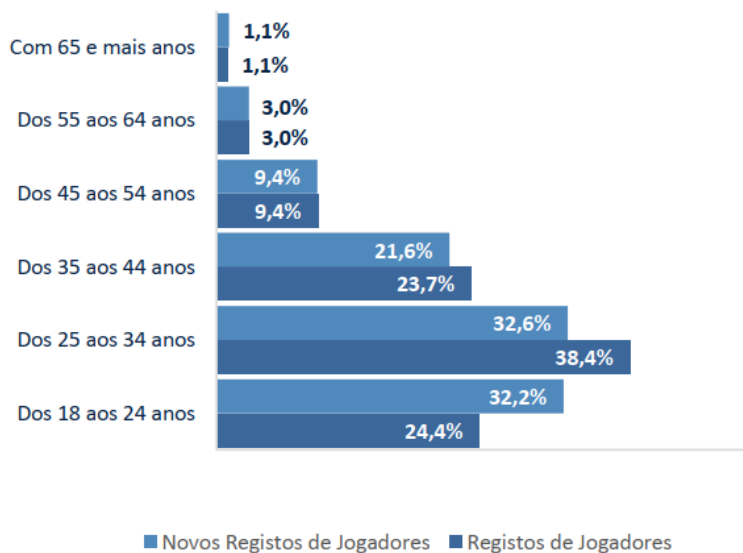


Figura 34: Jogadores - Distribuição por Estrutura Etária (Fonte: SRIJ)

Durante o 2º trimestre de 2019, observou-se a prática de jogo, ou seja, a realização de pelo menos uma aposta em jogos de fortuna ou azar ou em apostas desportivas à cota *online*, em cerca de 297 mil jogadores.

Finalmente, a Figura 37 demonstra que do número total de jogadores com prática de jogo, 49,1% efetuaram as suas apostas exclusivamente em apostas desportivas à cota, 35,5% jogaram somente em jogos de fortuna ou azar e 15,4% dos jogadores tiveram prática de jogo em apostas desportivas à cota e em jogos de fortuna ou azar.

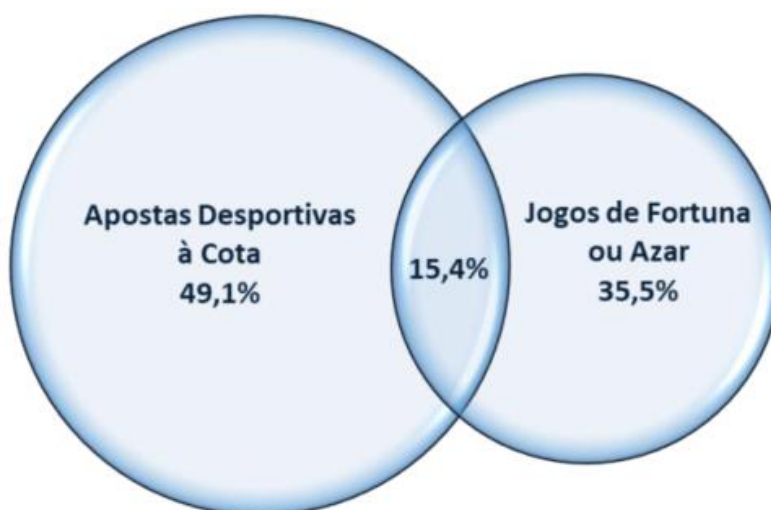


Figura 35: Jogadores - Prática de Jogos por Categoria (Fonte: SRIJ)

Monitorização de operadores ilegais de jogo *online*

O Serviço de Regulação e Inspeção de Jogos (SRIJ), no âmbito dos poderes que lhe estão legalmente conferidos, desenvolve a sua atuação no controlo e monitorização da atividade dos operadores ilegais que disponibilizam jogos e apostas *online* no mercado nacional.

Neste contexto, o SRIJ tem notificado as entidades identificadas como operadores ilegais de jogo *online* para que estas cessem essa atividade e removam da *Internet* o serviço de jogos e apostas *online*.

Sempre que essas entidades, no prazo legalmente fixado, não procedam em conformidade com o que lhes foi determinado, o SRIJ notifica os prestadores intermediários de serviços em rede (ISP's) para que estes interrompam o acesso aos respetivos websites, impedindo a disponibilização e utilização dessa oferta em Portugal.

Desde a entrada em vigor do RJO, em 29 de junho de 2015, e até 30 de junho de 2019, foram notificados para encerrarem a sua atividade em Portugal 408 operadores ilegais (21 durante o 2º trimestre de 2019) e procedeu-se à notificação aos ISP's para o bloqueio de 324 sítios na *Internet* de operadores ilegais (22 no trimestre em análise) que, não obstante terem sido notificados pelo SRIJ para cessarem a atividade, continuaram a disponibilizar em Portugal jogos e apostas *online*.

No total foram efetuadas 13 participações junto do Ministério Público para efeitos de instauração dos correspondentes processos-crime conforme ilustrado na Figura 38.



Figura 36: Operadores de Jogo Online Ilegais (Fonte: SRIJ)

Apêndice B - Demonstração de termos técnicos sobre as apostas desportivas

Como atribuir uma probabilidade a um resultado no futebol

A Distribuição de Poisson, associada aos dados históricos, fornece um método simples e fiável para calcular o resultado mais provável num jogo de futebol. É um conceito matemático para converter as médias numa probabilidade para resultados variáveis de uma distribuição. A probabilidade de que existam exatamente k ocorrências (k sendo um inteiro não negativo, $k = 0, 1, 2, \dots$), onde:

$$f(k; \lambda) = \frac{e^{-\lambda} \lambda^k}{k!}$$

Figura 37: Fórmula da Distribuição de Poisson

- e é base do logaritmo natural ($e = 2.71828\dots$);
- $k!$ é o fatorial de k ;
- λ é um número real, igual ao número esperado de ocorrências que ocorrem num dado intervalo de tempo. Por exemplo, se o evento ocorre a uma média de 4 minutos, e o número de eventos que ocorre num intervalo de 10 minutos:
 $\lambda = 10/4 = 2.5$.

Antes de utilizar a Distribuição de Poisson, para calcular o resultado mais provável de um jogo, é necessário calcular o número médio de golos que é provável que cada equipa marque nesse jogo. Para tal, é necessário calcular a capacidade ofensiva e a capacidade defensiva de cada equipa e compará-las.

A seleção de um intervalo de dados representativos é fundamental quando se calcula a capacidade ofensiva e a capacidade defensiva: por um lado, se o intervalo for demasiado grande, os dados não serão relevantes para a capacidade atual das equipas, por outro, se for demasiado curto, pode permitir que valores atípicos enviesem os dados.

Utilizando por exemplo os 38 jogos jogados por cada equipa numa época completa da Premier League Inglesa, dará uma amostra de tamanho suficiente para aplicar a Distribuição de Poisson.

Como calcular a capacidade ofensiva

O primeiro passo para calcular a capacidade ofensiva com base nos resultados da última época é determinar a média de golos marcados por cada equipa, nos jogos em casa e nos jogos fora. Para tal, utilizar o número total de golos marcados na última época e dividir pelo número de jogos disputados:

- ❖ Total de golos marcados em casa/número de jogos (na época)
- ❖ Total de golos marcados fora/número de jogos (na época)

Na época de 2015/16 da Premier League inglesa, houve 567 golos/380 jogos em casa e 459/380 fora, o que equivale a uma média de 1,492 golos por jogo em casa e 1,207 fora.

- ❖ Número médio de golos marcados em casa: 1,492
- ❖ Média de golos marcados fora: 1,207

A proporção entre a média da equipa e a média da liga é o que constitui a capacidade ofensiva.

Como calcular a capacidade defensiva

Também é necessária a média de golos sofridos por uma equipa média. É simplesmente o inverso dos valores anteriores, uma vez que o número de golos marcados pela equipa da casa é igual ao número de golos sofridos pela equipa visitante:

- ❖ Média de golos sofridos em casa: 1,207
- ❖ Média de golos sofridos fora: 1,492

A proporção entre a média da equipa e a média da liga é o que constitui a capacidade defensiva.

Como exemplo, serão utilizados os números acima para calcular a capacidade ofensiva e a capacidade defensiva tanto da Equipa A como da Equipa B.

Resultados de um exemplo prático

Calcular a capacidade ofensiva da Equipa A:

1. Utilizar o número de golos marcados em casa na última época pela equipa da casa 35 e dividir pelo número de jogos em casa 19 = 1,842. $(35/19)$
2. Dividir este valor pela média de golos marcados por jogo pelas equipas da casa (1,842/1,492) para obter uma capacidade ofensiva de 1,235. $(35/19) / (567/380) = 1,235$

Capacidade ofensiva da Equipa B:

Para calcular o número de golos que a Equipa B poderá marcar, basta utilizar as fórmulas acima, mas substituindo o número médio de golos marcados em casa pelo número médio de golos marcados em jogos fora.

- $(24/19) / (459/380) = 1,046$

Capacidade defensiva da Equipa A:

- $(15/19) / (459/380) = 0,653$

Da mesma forma que se prevê o número de golos que a Equipa A irá marcar, é possível calcular o número provável de golos que a Equipa B poderá marcar (para tal, basta multiplicar a capacidade ofensiva do Everton pela capacidade defensiva a Equipa A e pelo número médio de golos marcados em jogos fora na Premier League):

- $1,046 \times 0,653 \times 1,207 = 0,824$

Calcular a capacidade defensiva da Equipa B:

1. Utilizar o número de golos sofridos fora na última época pela equipa visitante 25 e dividir pelo número de jogos fora (25/19): 1,315.
2. Dividir este valor pela média de golos sofridos por jogo pela equipa visitante na época (1,315/1,492) para obter a capacidade defensiva de 0,881. $(25/19) / (567/380) = 0,881$.

Então, o número provável de golos que a Equipa A poderá marcar é (para tal, basta multiplicar a capacidade ofensiva da Equipa A pela capacidade defensiva da Equipa B e pelo número médio de golos marcados em casa na Premier League): $1,235 \times 0,881 \times 1,492 = 1,623$

Introduzindo as diferentes ocorrências do evento – no nosso caso, resultados de golos de 0 a 5 – e as ocorrências esperadas que são a possibilidade de cada equipa vir a marcar – a taxa média de sucesso da Equipa A é de 1,623, e a da Equipa B é de 0,824.

0	1	2	3	4	5
19,73%	32,02%	25,99%	14,06%	5,07%	1,85%
43,86%	36,14%	14,89%	4,09%	0,84%	0,14%

Figura 38: Probabilidade de cada equipa fazer um número de golos

Este exemplo mostra que há 19,73% de hipóteses da Equipa A não marcar nenhum golo, mas que há 32,02% de hipóteses de marcar um único golo e 25,99% de hipóteses de marcar dois golos. Por outro lado, a Equipa B tem 43,86% de hipóteses de não marcar nenhum golo, 36,14% de hipóteses de marcar um golo e 14,89% de hipóteses de marcar dois golos.

Uma vez que ambos os resultados são independentes (matematicamente falando), o resultado esperado é de 1-0 – ao emparelhar os resultados mais prováveis para cada equipa. Ao multiplicar estas duas probabilidades, obterá a probabilidade de um resultado de 1-0 – $(0,3202 \times 0,4386) = 0,1404$ ou 14,04%.

Converter as hipóteses estimadas em probabilidades

O exemplo acima exemplificou que há 11,53% de hipóteses de ocorrer um empate a 1-1 ($0,3202 \times 0,3614$) quando se aplica a fórmula da Distribuição de Poisson.

Assim que sejam calculadas as hipóteses de cada resultado, basta convertê-las em probabilidades e compará-las com as probabilidades de uma casa de apostas a fim de encontrar apostas de potencial valor. Para tal, bastaria calcular a probabilidade de todas as combinações de empates possíveis e somá-las. Assim, obteria a probabilidade de se registar um empate, independentemente do resultado.

Naturalmente, existe na realidade um número infinito de possibilidades de empate (por exemplo, ambas as equipas poderiam marcar 10 golos cada), mas as hipóteses de um empate com um resultado acima de 5-5 são tão reduzidas que se pode ignorar neste modelo.

Utilizando o exemplo da Equipa A vs. Equipa B, e combinando todas as possibilidades de empate, dá uma probabilidade de 0,2472 ou 24,72% - tal daria uma *odd* real de 4,05 ($1/0,2472$).

Aplicação da Distribuição de Poisson em Excel

A sintaxe da função DIST.POISSON (dist.poisson: dist.poisson(x,média,cumulativo) tem os seguintes argumentos:

- X: O número de eventos para os quais você deseja a probabilidade (deve ser ≥ 0). Este valor deve ser um inteiro e se um decimal for fornecido, ele será truncado para um inteiro pelo Excel. Argumento necessário;
- Média: O valor numérico esperado. Este argumento é necessário;
- Cumulativo: Um valor lógico que determina a forma da distribuição de probabilidade fornecida. Se cumulativo for VERDADEIRO, DIST.POISSON retornará à probabilidade Poisson de que o número de eventos aleatórios estará entre zero e x inclusive; se FALSO, retornará à função massa da probabilidade Poisson de que o número de eventos será equivalente a x.

O cumulativo pode ser:

1. VERDADEIRO - a função de distribuição cumulativa;
2. FALSO - a função de massa de probabilidade.

Aposta de valor (Expected value, ou +EV)

A fim de encontrar valor, um apostador necessita de determinar a sua própria probabilidade de um evento ocorrer. Em termos simples, se um apostador conseguir identificar corretamente eventos nos quais a possibilidade de o evento ocorrer é superior àquela sugerida pelas probabilidades da casa de apostas, então irá ganhar a longo prazo.

Uma aposta oferece valor quando: a probabilidade do resultado é maior a possibilidade sugerida pelas probabilidades disponibilizadas pelas casas de apostas.

Exemplo de uma aposta de valor: Exemplo de uma moeda ao ar. Ignorando a possibilidade de a moeda cair de lado, onde as hipóteses de sair cara é de 50%. Isto sugere uma probabilidade justa de 2,00. Assim que a casa de apostas adiciona a sua margem, as probabilidades poderiam ser (dependendo da casa de apostas) de cerca de 1,70, sugerindo uma hipótese de 58,8% de sair cara.

Esta aposta oferece pouco valor, uma vez que a probabilidade sobre uma hipótese de 50%, como se houvesse uma hipótese de 58,8% de sair cara. Agora, imaginando que existe alguma incerteza quanto à hipótese percentual de cara. A hipótese de cara é de 50%, mas o mercado é incerto. A casa de apostas está agora a oferecer probabilidades de 2,2 em como sai cara.

Agora, é oferecida a oportunidade de fazer uma aposta com probabilidades de 50%, como se o evento tivesse apenas uma hipótese de 45,5% de ocorrer. Esta é uma aposta de valor.

Assim, com o nosso exemplo da moeda ao ar, o cálculo de valor esperado de uma parada de 10 € é: $0,5 \times 12 \text{ €} - 0,5 \times 10 \text{ €} = +1 \text{ €}$

Para cada aposta de 10 €, o retorno esperado é de 11 € (um lucro de 1 €).

Por cada moeda atirada, os ganhos são de 11 € por cada 10 € apostados. Isto tornar-se-ia, logicamente, muito lucrativo a longo prazo.

Existem três fatores muito importantes para ter sucesso no mundo das apostas:

✓ Fazendo apostas de valor:

Uma boa forma de encontrar apostas de valor é olhar para as diferenças nas probabilidades das casas de apostas. Nos casos em que as casas de apostas discordam quanto à hipótese de um evento ocorrer pode ser que exista valor.

Desenvolver um modelo superior ou tirar partido da assimetria das informações pode também ser uma boa forma de encontrar valor. Se um modelo ou informações internas derem ao apostador uma noção mais clara da real probabilidade do que tem a casa de apostas, então é possível ter lucros.

✓ Pensar sempre a longo prazo:

Quando se aposta no valor, é importante apostar em termos de longo prazo. Mesmo com a vantagem clara no atirar a moeda ao ar com probabilidades de 2,2, continua a haver uma hipótese de 50% de que o apostador perca. É aqui que uma estratégia relacionada com a gestão de banca se torna importante, já que os fundos podem ser dizimados ao oferecer paradas demasiado altas, mesmo em apostas que oferecem valor, porque não significa que uma aposta tenha valor que vai ocorrer.

✓ A importância das probabilidades:

O valor tem tudo a ver com encontrar as melhores probabilidades. Se os apostadores aceitam probabilidades inferiores noutros locais, então uma estratégia de apostas de valor tornar-se-á menos rentável ou poderá até dar origem a perdas ao longo do tempo.

Golos esperados

Os golos esperados são uma métrica que avalia a possibilidade de um remate se tornar num golo. Proporciona uma boa forma de avaliar a qualidade dos remates, uma vez que um remate com um valor de golo esperado xG de 0,4 deverá ser marcado em 40% das vezes. Um xG de 1 é o valor mais elevado que um remate único pode ter, deixando implícito que o jogador tem uma hipótese de 100% de marcar. O número de fatores levados em conta ao calcular o xG de um remate depende do modelo utilizado.

Os golos esperados podem ser benéficos, porque aumentam o tamanho da amostra utilizada ao analisar o futebol. O futebol é um jogo de baixa marcação e os golos são um evento raro. Por isso, os dados sobre golos puros podem, por vezes, ser enganadores.

Existem inúmeros exemplos em cada época em que a equipa que criou mais hipóteses acaba por perder o jogo. Os dados básicos sobre os golos têm dificuldades em refletir este aspeto e podem não ser representativos do jogo real enquanto resultado. Os golos esperados levam as hipóteses em consideração ao calcular o número de golos que, em média, são marcados de cada posição.

Antes dos golos esperados, métricas como o total de remates ou os remates no alvo eram utilizadas para tentar analisar os jogos. Tal como os golos, estas estatísticas podem ser enganadoras. O total de remates conta uma tentativa da linha de meio campo como sendo igual a um remate feito de dentro da pequena área. Para tal, na subsecção seguinte são demonstradas formas de aproveitar estas estatísticas da forma mais correta, através do “Exemplo prático da época 2018/2019 da Liga NOS”, demonstrado no Apêndice C.

Apêndice C - Exemplo prático da época 2018/2019 da Liga NOS

- Golos marcados e sofridos;
- Cantos a favor e contra;
- Remates efetuados e permitidos;
- Remates à baliza efetuados e permitidos;
- Quantos remates são necessários para fazer um golo;
- Quantos remates à baliza são necessários para fazer um golo;
- Quantos remates são necessários para sofrer um golo;
- Quantos remates à baliza são necessários para sofrer um golo;
- Cartões amarelos e vermelhos;
- Faltas cometidas e faltas sofridas.

Então, é possível retirar informações, com base nos golos esperados demonstrados anteriormente, para fazer apostas do género:

Exemplo 1: Jogo Benfica vs Nacional – O Benfica em casa necessita de fazer 2.1 remates à baliza para fazer um golo, e o Nacional fora necessita de 2.0 remates à baliza para sofrer um golo, então, supondo que o Benfica já fez 4 remates à baliza, os xG para o Benfica neste momento são de 1.9 golos, ou seja, o Benfica tem como xG pelo menos 2 golos. Então, uma aposta no Benfica fazer pelo menos 2 golos (+1.5 team goals), é uma aposta em ter em atenção, dependendo da *odd* disponibilizada pela casa de apostas.

Exemplo 2: Jogo Setúbal vs Portimonense – São as duas equipas que mais dificuldades têm de fazer golos em casa e fora, respetivamente. Então, com base nesta informação, a probabilidade de o jogo ter poucos golos já é bastante elevada, se o jogo ao longo do tempo demonstrar que não há remates perigosos, apostar que não vão haver muitos golos no jogo é uma aposta em ter em atenção, dependendo da *odd* disponibilizada pela casa de apostas.

Nas Figuras 44, 45 e 46, é possível obter mais certezas na realização de uma aposta para saírem golos (ou não). São demonstrados os seguintes dados:

- No Over 0.5, qual a probabilidade de num jogo envolvendo uma determinada equipa, de sair pelo menos um golo;
- No Over 1.5, qual a probabilidade de num jogo envolvendo uma determinada equipa, de saírem pelo menos dois golos;
- No Over 2.5, qual a probabilidade de num jogo envolvendo uma determinada equipa, de saírem pelo menos três golos.

Choose goals market to display: **Over 0.5 goals**

Overall table											Match count		%		Fair odds		Average goals per match		
#	Team	Played	Won	Drawn	Lost	Goals scored	Goals conceded	GD	Points	League position	Over 0.5 goals	Under 0.5 goals	Over 0.5 goals	Under 0.5 goals	Over 0.5 goals	Under 0.5 goals	Scored	Conceded	Total
1	Benfica	34	28	3	3	103	31	72	87	1	34	0	100	0	1,00	-	3,0	0,9	3,9
2	Sp Braga	34	21	4	9	56	37	19	67	4	34	0	100	0	1,00	-	1,6	1,1	2,7
3	Moreirense	34	16	4	14	39	44	-5	52	6	34	0	100	0	1,00	-	1,1	1,3	2,4
4	Portimonense	34	11	6	17	44	59	-15	39	11	34	0	100	0	1,00	-	1,3	1,7	3,0
5	Rio Ave	34	12	9	13	50	52	-2	45	7	33	1	97	3	1,03	34,00	1,5	1,5	3,0
6	Tondela	34	9	8	17	40	54	-14	35	15	33	1	97	3	1,03	34,00	1,2	1,6	2,8
7	Porto	34	27	4	3	74	20	54	85	2	32	2	94	6	1,06	17,00	2,2	0,6	2,8
8	Sp Lisbon	34	23	5	6	72	33	39	74	3	32	2	94	6	1,06	17,00	2,1	1,0	3,1
9	Guimaraes	34	15	7	12	46	34	12	52	5	32	2	94	6	1,06	17,00	1,4	1,0	2,4
10	Maritimo	34	12	3	19	26	44	-18	39	12	32	2	94	6	1,06	17,00	0,8	1,3	2,1
11	Aves	34	10	6	18	35	49	-14	36	14	32	2	94	6	1,06	17,00	1,0	1,4	2,5
12	Boavista	34	13	5	16	34	40	-6	44	8	31	3	91	9	1,10	11,33	1,0	1,2	2,2
13	Santa Clara	34	11	9	14	43	45	-2	42	10	31	3	91	9	1,10	11,33	1,3	1,3	2,6
14	Chaves	34	8	8	18	34	57	-23	32	16	31	3	91	9	1,10	11,33	1,0	1,7	2,7
15	Nacional	34	7	7	20	33	73	-40	28	17	30	4	88	12	1,13	8,50	1,0	2,1	3,1
16	Feirense	34	3	11	20	27	64	-37	20	18	30	4	88	12	1,13	8,50	0,8	1,9	2,7
17	Belenenses	34	10	13	11	42	51	-9	43	9	29	5	85	15	1,17	6,80	1,2	1,5	2,7
18	Setubal	34	8	12	14	28	39	-11	36	13	28	6	82	18	1,21	5,67	0,8	1,1	2,0

Figura 42: Probabilidade de haver pelo menos um golo

Choose goals market to display: **Over 1.5 goals**

Overall table											Match count		%		Fair odds		Average goals per match		
#	Team	Played	Won	Drawn	Lost	Goals scored	Goals conceded	GD	Points	League position	Over 1.5 goals	Under 1.5 goals	Over 1.5 goals	Under 1.5 goals	Over 1.5 goals	Under 1.5 goals	Scored	Conceded	Total
1	Benfica	34	28	3	3	103	31	72	87	1	29	5	85	15	1,17	6,80	3,0	0,9	3,9
2	Rio Ave	34	12	9	13	50	52	-2	45	7	29	5	85	15	1,17	6,80	1,5	1,5	3,0
3	Portimonense	34	11	6	17	44	59	-15	39	11	29	5	85	15	1,17	6,80	1,3	1,7	3,0
4	Sp Lisbon	34	23	5	6	72	33	39	74	3	27	7	79	21	1,26	4,86	2,1	1,0	3,1
5	Porto	34	27	4	3	74	20	54	85	2	26	8	76	24	1,31	4,25	2,2	0,6	2,8
6	Tondela	34	9	8	17	40	54	-14	35	15	26	8	76	24	1,31	4,25	1,2	1,6	2,8
7	Moreirense	34	16	4	14	39	44	-5	52	6	25	9	74	26	1,36	3,78	1,1	1,3	2,4
8	Aves	34	10	6	18	35	49	-14	36	14	25	9	74	26	1,36	3,78	1,0	1,4	2,5
9	Sp Braga	34	21	4	9	56	37	19	67	4	24	10	71	29	1,42	3,40	1,6	1,1	2,7
10	Feirense	34	3	11	20	27	64	-37	20	18	24	10	71	29	1,42	3,40	0,8	1,9	2,7
11	Belenenses	34	10	13	11	42	51	-9	43	9	23	11	68	32	1,48	3,09	1,2	1,5	2,7
12	Nacional	34	7	7	20	33	73	-40	28	17	23	11	68	32	1,48	3,09	1,0	2,1	3,1
13	Santa Clara	34	11	9	14	43	45	-2	42	10	22	12	65	35	1,55	2,83	1,3	1,3	2,6
14	Setubal	34	8	12	14	28	39	-11	36	13	22	12	65	35	1,55	2,83	0,8	1,1	2,0
15	Chaves	34	8	8	18	34	57	-23	32	16	22	12	65	35	1,55	2,83	1,0	1,7	2,7
16	Boavista	34	13	5	16	34	40	-6	44	8	21	13	62	38	1,62	2,62	1,0	1,2	2,2
17	Guimaraes	34	15	7	12	46	34	12	52	5	20	14	59	41	1,70	2,43	1,4	1,0	2,4
18	Maritimo	34	12	3	19	26	44	-18	39	12	17	17	50	50	2,00	2,00	0,8	1,3	2,1

Figura 43: Probabilidade de haver pelo menos dois golos

Choose goals market to display: **Over 2.5 goals**

Overall table											Match count		%		Fair odds		Average goals per match		
#	Team	Played	Won	Drawn	Lost	Goals scored	Goals conceded	GD	Points	League position	Over 2.5 goals	Under 2.5 goals	Over 2.5 goals	Under 2.5 goals	Over 2.5 goals	Under 2.5 goals	Scored	Conceded	Total
1	Benfica	34	28	3	3	103	31	72	87	1	23	11	68	32	1,48	3,09	3,0	0,9	3,9
2	Sp Lisbon	34	23	5	6	72	33	39	74	3	22	12	65	35	1,55	2,83	2,1	1,0	3,1
3	Rio Ave	34	12	9	13	50	52	-2	45	7	22	12	65	35	1,55	2,83	1,5	1,5	3,0
4	Porto	34	27	4	3	74	20	54	85	2	20	14	59	41	1,70	2,43	2,2	0,6	2,8
5	Nacional	34	7	7	20	33	73	-40	28	17	20	14	59	41	1,70	2,43	1,0	2,1	3,1
6	Chaves	34	8	8	18	34	57	-23	32	16	18	16	53	47	1,89	2,13	1,0	1,7	2,7
7	Belenenses	34	10	13	11	42	51	-9	43	9	17	17	50	50	2,00	2,00	1,2	1,5	2,7
8	Portimonense	34	11	6	17	44	59	-15	39	11	17	17	50	50	2,00	2,00	1,3	1,7	3,0
9	Tondela	34	9	8	17	40	54	-14	35	15	17	17	50	50	2,00	2,00	1,2	1,6	2,8
10	Moreirense	34	16	4	14	39	44	-5	52	6	16	18	47	53	2,13	1,89	1,1	1,3	2,4
11	Santa Clara	34	11	9	14	43	45	-2	42	10	16	18	47	53	2,13	1,89	1,3	1,3	2,6
12	Sp Braga	34	21	4	9	56	37	19	67	4	15	19	44	56	2,27	1,79	1,6	1,1	2,7
13	Aves	34	10	6	18	35	49	-14	36	14	15	19	44	56	2,27	1,79	1,0	1,4	2,5
14	Feirense	34	3	11	20	27	64	-37	20	18	15	19	44	56	2,27	1,79	0,8	1,9	2,7
15	Guimaraes	34	15	7	12	46	34	12	52	5	12	22	35	65	2,83	1,55	1,4	1,0	2,4
16	Setubal	34	8	12	14	28	39	-11	36	13	12	22	35	65	2,83	1,55	0,8	1,1	2,0
17	Boavista	34	13	5	16	34	40	-6	44	8	11	23	32	68	3,09	1,48	1,0	1,2	2,2
18	Maritimo	34	12	3	19	26	44	-18	39	12	10	24	29	71	3,40	1,42	0,8	1,3	2,1

Figura 44: Probabilidade de haver pelo menos três golos