



**NOVA**

**IMS**

Information  
Management  
School

# MGI

---

**Mestrado em Gestão de Informação**

Master Program in Information Management

## **DA TEORIA À PRÁTICA, IMPLEMENTAÇÃO DE SISTEMAS CRM**

João Manuel Ribeiro Folgado Correia

Relatório de Estágio apresentado como requisito parcial para  
obtenção do grau de Mestre em Gestão de Informação

NOVA Information Management School  
Instituto Superior de Estatística e Gestão de Informação  
Universidade Nova de Lisboa

2015

DA TEORIA À PRÁTICA, IMPLEMENTAÇÃO DE SISTEMAS  
CRM

João Manuel Ribeiro  
Folgado Correia

MGI



**NOVA Information Management School**  
**Instituto Superior de Estatística e Gestão de Informação**  
Universidade Nova de Lisboa

## **DA TEORIA À PRÁTICA, IMPLEMENTAÇÃO DE SISTEMAS CRM**

por

João Manuel Ribeiro Folgado Correia

Relatório de Estágio apresentado como requisito parcial para a obtenção do grau de Mestre em Gestão de Informação, Especialização em Gestão dos Sistemas de Informação

**Orientador:** Professor Vítor Manuel Duarte dos Santos

Novembro 2015

## RESUMO

Este Relatório de Estágio tem como objetivo principal dar a conhecer a empresa em que realizei o estágio (UNISYS), o contexto sobre o qual trabalhei (CRM), tecnologias que utilizei, as atividades e funções que fui tendo ao longo do estágio, bem como as *skills* necessárias a executar essas mesmas funções.

É ainda, feita uma ponte entre o tema “Ciclo de Vida de Desenvolvimento de Software” (matéria dada ao longo da Licenciatura e Mestrado) e o estágio em si. A pertinência desta interligação, surge uma vez que se trata de uma matéria que me fascina, e pelo facto de terem sido utilizadas diversas metodologias ao longo do meu estágio.

## PALAVRAS-CHAVE

CRM; SDLC; Gestão; Teoria e Prática; Metodologia; *Framework*;

# ÍNDICE

1. INTRODUÇÃO.....	1
1.1. CONTEXTO ACADÊMICO .....	1
1.2. CONTEXTO EMPRESARIAL .....	1
1.3. OBJETIVOS DO ESTÁGIO .....	3
2. CUSTOMER RELATIONSHIP MANAGEMENT (CRM) .....	4
2.1. MICROSOFT DYNAMICS CRM .....	6
2.1.1. HISTÓRIA DO MICROSOFT DYNAMICS CRM.....	8
3. CICLO DE VIDA DE DESENVOLVIMENTO DE <i>SOFTWARE</i> (CVDS) .....	9
3.1. MODELO CASCATA .....	12
3.2. MODELOS ITERATIVOS .....	14
3.3. PROTOTIPAGEM .....	15
4. MODELOS áGEIS .....	17
4.1. FUNDAMENTOS.....	17
4.2. ÁGIL VS MODELOS TRADICIONAIS SDLC.....	18
4.3. MODELO DE GESTÃO SCRUM.....	19
5. FRAMEWORK DE GESTÃO DE SKILLS (SFIA).....	21
5.1. CONTEXTO .....	21
5.2. <i>SKILLS</i> E NÍVEIS DE RESPONSABILIDADES .....	22
5.3. <i>SKILLS</i> PROFISSIONAIS, O QUE SÃO E O QUE NÃO SÃO .....	22
6. TECNOLOGIAS.....	24
6.1. .NET C#.....	24
6.2. JAVASCRIPT.....	24
6.3. HTML.....	24
6.4. CSS .....	25
7. ATIVIDADES DESENVOLVIDAS .....	26
7.1. PROJETO – IMPLEMENTAÇÃO DE SOLUÇÃO XRM .....	27
7.1.1. DESAFIO.....	28
7.1.2. TAREFAS E SKILLS.....	30
7.2. PROJETO – IMPLEMENTAÇÃO DE SOLUÇÃO CRM .....	33
7.2.1. DESAFIO.....	34
7.2.2. TAREFAS E SKILLS.....	35

8. CONCLUSÕES .....	39
8.1. APRECIÇÃO CRÍTICA DO TRABALHO DESENVOLVIDO .....	39
8.2. PRESPECTIVAS FUTURAS .....	39
8.3. APRECIÇÃO GLOBAL DO ESTÁGIO .....	40
BIBLIOGRAFIA .....	41
ANEXOS .....	42

## ÍNDICE DE FIGURAS

Figura 1 - Unisys Corporation Logo .....	1
Figura 2 - Competências Microsoft da Unisys Portugal .....	2
Figura 3 - Vertentes CRM .....	4
Figura 4 - Estrutura de personalizações de solução MS Dynamics CRM .....	6
Figura 5 - Estrutura da aplicação MS Dynamics CRM .....	7
Figura 6 - Crescimento de receitas no ano de 2014 de cada <i>software</i> CRM .....	8
Figura 7 - Ciclo de Vida de Desenvolvimento de <i>Software</i> .....	9
Figura 8 - Esquema do modelo <i>Waterfall</i> .....	12
Figura 9 - Esquema do Modelo Iterativo.....	14
Figura 10 - Modelo Agile .....	17
Figura 11 - Modelo Agile vs Modelos tradicionais .....	18
Figura 12 - Fases SCRUM .....	20
Figura 13 - <i>Sprint</i> SCRUM .....	20
Figura 14 - Estrutura SFIA.....	21
Figura 15 - Os sete níveis SFIA.....	22
Figura 16 - Esquemático sobre organização de uma <i>skill</i> .....	22
Figura 17 - Esquemático da plataforma de suporte do Sistema Integrado de Gestão de Atas (SIGA).....	27
Figura 18 - Tabela das diferentes fases da metodologia SCRUM .....	29
Figura 19 - Código de <i>Plugin</i> de controlo de Nome de um Ponto de Agenda .....	31
Figura 20 - Exemplo de ordenação dos Pontos de Agenda.....	31
Figura 21 - <i>Skills</i> utilizadas no Projeto xRM .....	32
Figura 22 - Exemplo de um sistema AS400 .....	33
Figura 23 - Fotografia da sala de trabalhos com desenhos da solução .....	36
Figura 24 - <i>Custom Action</i> de obtenção do Portfólio do Cliente .....	36
Figura 25 - Exemplo real da representação do Portfólio do Cliente.....	37
Figura 26 - Representação da chamada feita em tempo real para obtenção do <i>grading</i> de um Cliente .....	37
Figura 27 - <i>Skills</i> utilizadas no Projeto CRM .....	38



## ÍNDICE DE TABELAS

Tabela 1 - Vantagens e Desvantagens do Modelo <i>Waterfall</i> .....	13
Tabela 2 - Vantagens e Desvantagens do Modelo Iterativo .....	14
Tabela 3 - Vantagens e Desvantagens do Modelo de Prototipagem.....	16

## LISTA DE SIGLAS E ABREVIATURAS

<b>CE</b>	Caderno de Encargos
<b>CRM</b>	Customer Relationship Management
<b>CVDS</b>	Ciclo de Vida de Desenvolvimento de Software
<b>ERS</b>	Especificação de Requisitos de Software
<b>IMS</b>	Information Management School
<b>ISO</b>	International Standards Organization
<b>MS</b>	Microsoft
<b>SDLC</b>	Software Development Life Cycle
<b>SI</b>	Sistemas de Informação

# 1. INTRODUÇÃO

Este relatório pretende abordar a relação que existe entre a teoria, os conceitos e metodologias, dados nas unidades curriculares ao longo da formação na Nova Information Management School, e a prática, o que é realmente aplicado no mundo empresarial.

## 1.1. CONTEXTO ACADÉMICO

Segundo o INE (2007), cerca de 25% das grandes empresas em Portugal utilizavam sistemas de *Customer Relationship Management* (CRM). Em 2014, a Microsoft afirmou que a taxa de utilização dos seus sistemas de CRM, tinha vindo a aumentar de forma contínua em 1% ao ano, mantendo-se as perspetivas da continuação desse crescimento anual percentual até 2020.

Tendo concluído a Licenciatura em Sistemas e Tecnologias de Informação, e finalista do Mestrado de Gestão de Informação com especialização em Gestão dos Sistemas de Informação, a possibilidade de poder fazer parte de uma empresa como a UNISYS, conceituada na implementação de sistemas Microsoft CRM, é uma oportunidade de extremo valor para aumentar os conhecimentos não só de CRM e da sua implementação, como também os modelos de negócio de diferentes empresas em diferentes setores, e ainda, o impacto de um sistema de gestão de relação com clientes na obtenção dos resultados desejados por parte das empresas.

Com o estágio tive a possibilidade de crescer a nível de raciocínio lógico, em que foram desenvolvidas capacidades de programação, assim como competências de análise, resolução de problemas e levantamento de requisitos, bem como a nível de *skills*, desde trabalho em equipa e valorização pessoal, até ao contacto com o cliente.

## 1.2. CONTEXTO EMPRESARIAL

A UNISYS Corporation foi fundada em 1986. É normalmente conhecida apenas por UNISYS, uma empresa Norte-Americana de Sistemas de Informação que presta serviços de consultoria a nível global. A UNISYS emprega cerca de 20.000 empregados e encontra-se presente nos cinco continentes, América, Europa, Ásia, África e Oceânia, apresentando receitas de cerca 3.356M €, e 154M € em lucros no ano de 2014 (*UNISYS Annual Report 2014*).



Figura 1 - Unisys Corporation Logo

Em Portugal, a UNISYS (Portugal) – Sistemas de Informação SA (daqui adiante referenciada apenas como UNISYS) emprega atualmente cerca de 300 colaboradores, o que significa numa receita anual de cerca de 16M € (2014). Sendo uma empresa de consultoria informática, tem como base a prestação

de serviços como o desenvolvimento de *software* à medida, a personalizações de aplicações já existentes, a arquitetura e implementação de redes informáticas, entre outras. Apesar do grande reconhecimento por parte das empresas que contratam serviços à UNISYS, seja em que área for, a mesma tem um grau de especialização bastante elevado em tecnologias Microsoft, mais precisamente Microsoft Dynamics CRM. Esta especialização é de tal forma notável que a UNISYS tem uma parceria de “*Gold Partner*” com a Microsoft, o que significa que a Microsoft recomenda a UNISYS como empresa ideal para o fornecimento de solução MS Dynamics CRM.

Torna-se portanto importante referir o que significa em concreto ser *Gold Partner* da Microsoft:

*“As competências Gold demonstram a sua experiência de melhor qualidade no marketplace da Microsoft. Conquistar estas competências Gold é evidência do compromisso mais consistente e profundo com uma área de solução comercial específica, sob procura, juntamente com a distinção de estar entre apenas 1% dos parceiros internacionais da Microsoft que atingiram esse excelente grau de profissionalismo.”* (Microsoft)



Figura 2 - Competências Microsoft da Unisys Portugal

A UNISYS tem sido ao longo dos anos uma empresa chave no sucesso da aplicação MS Dynamics CRM da Microsoft, pois tem estado presente em todos os projetos de grande dimensão a nível nacional, desde Bancos a empresas públicas e instituições de renome. Este sucesso tem produzido excelentes resultados contando neste momento com cerca de 20 colaboradores a full-time na arquitetura, desenvolvimento, manutenção e suporte de plataformas CRM. Devido ao domínio sobre a ferramenta de gestão de relação com clientes da Microsoft, é agora possível à UNISYS ganhar concursos a nível internacional. Por exemplo, recentemente foi fechado um contrato em França para a primeira implementação de um sistema MS Dynamics CRM Online.

Para uma correta análise, desenho, desenvolvimento, implementação e controlo é necessário seguir uma abordagem coesa e estruturada desde o início até ao fim de um projeto. Para tal a UNISYS aplica *Frameworks* e *best practices* para o correto desenvolvimento de grandes projetos.

Sendo portanto a UNISYS, uma empresa de referência no mundo do CRM em Portugal, a oportunidade que me foi dada, de poder fazer parte da mesma era uma experiência que tinha de ser aproveitada ao máximo, de forma a conseguir enriquecer os meus conhecimentos e evoluir enquanto profissional num universo tão competitivo e exigente como é o dos Sistemas de Informação.

### 1.3. OBJETIVOS DO ESTÁGIO

Durante este estágio foi proposto a cada participante, que definisse os seus próprios objetivos pessoais e profissionais a atingir no fim do ano. No entanto, existem objetivos gerais que são atribuídos a todos os novos colaboradores de forma a atingirem as seguintes capacidades:

- Entender e dominar a ferramenta MS Dynamics CRM;
- Conhecer a potencialidade de evolução de CRM para xRM;
- Utilizar metodologias de trabalho;
- Analisar, desenhar e desenvolver soluções de forma a satisfazer requisitos;
- Elaborar estimativas e planear para a boa concretização de um projeto.

Com esta abordagem, é possível chegar ao fim do ano com elevado conhecimento de todo o processo de desenvolvimento de *software*, bem como o entendimento geral do funcionamento do sistema de CRM fornecido pela Microsoft, as suas funcionalidades e potencialidades.

De forma a atingir os resultados esperados, a UNISYS tem uma abordagem de desenvolvimento colocando os seus colaboradores a exercerem diferentes papéis ao longo do ano. Algumas das tarefas a executar são:

- Análise de Caderno de Encargos (daqui por adiante referenciado como CE) e preparação de propostas;
- Discussão e análise do problema com o Cliente;
- Desenho e Arquitetura da solução a ser implementada;
- Divisão de tarefas ao longo do tempo proposto para a concretização do projeto;
- Distribuição de tarefas pela equipa envolvida;
- Desenvolvimento de *software*;
- Elaboração de documentação final para o Cliente;
- Preparação de testes de qualidade e de aceitação;
- Execução de testes;
- Formação de utilizadores;
- Avaliação dos questionários de satisfação do Cliente após entrega da solução.

## 2. CUSTOMER RELATIONSHIP MANAGEMENT (CRM)

A abreviatura CRM – *Customer Relationship Management* – aparece nos anos 90 juntamente com o desenvolvimento dos Sistemas de Informação, e desde então, várias têm sido as definições de CRM (Buttle, 2012), como por exemplo:

- “Estratégia de negócios a todos os níveis da empresa concebida para reduzir custos e aumentar lucros, solidificando a satisfação e lealdade. Um verdadeiro CRM reúne todas as informações de todas as fontes de dados existentes numa organização (e, quando apropriado, de fora da organização) para dar uma visão holística de cada cliente em tempo real.” (DestinationCRM, 2010)
- “Atividade que engloba não só aspetos da identificação e desenvolvimento da visão dos clientes, mas também do estreitamento do relacionamento com eles.” (Srivastava, Shervani, & Fahey, 1999)
- “Uma filosofia e uma estratégia de negócio, suportada por um sistema e uma tecnologia, concebida para melhorar as interações humanas num ambiente de negócio.” (Greenberg, 2003).

Após a análise de diferentes definições de CRM, é possível defender que se trata de um conjunto de atividades de carácter informático ou não, que corretamente identificadas e alinhadas com a estratégia de uma empresa, permitem identificar novas oportunidades comerciais e/ou potenciais novos clientes.



Figura 3 - Vertentes CRM

A "Gestão de Clientes" é, atualmente, uma peça-chave para a obtenção de resultados e objetivos definidos na camada estratégica das empresas. Atualmente, torna-se cada vez mais importante a atenção no acompanhamento direto e personalizado a cada cliente de forma a conseguir atingir o maior grau de satisfação do mesmo, bem como a máxima eficácia das campanhas de Marketing concebidas e efetuadas pela empresa.

O cliente contemporâneo é um cliente cada vez mais informado que toma decisões de compra mesmo antes de a empresa interagir diretamente com ele. É portanto deveras importante que seja dada a possibilidade de contactar com os mais diversos clientes, conseguindo desta forma, aumentar o nível de interações, bem como proporcionar relacionamento com o cliente, com o objetivo de se tornar possível ter uma interação única e personalizada, melhorando a experiência do mesmo durante o processo de decisão e compra de um determinado bem, produto ou serviço.

*“As empresas precisam saber gerenciar as diferenças de necessidades de cada consumidor fazendo com que ele se sinta único; precisam entender que o consumidor passou a ser fundamental. Aos poucos, as empresas têm vindo a perceber a importância de tratar cada cliente individualmente, oferecendo produtos e serviços personalizados, objetivando sempre a sua satisfação” (Valente 2010)*

Dado as estratégias de marketing estarem em evolução e crescimento constantes, torna-se premente, que as empresas assumam responsabilidade na interação com o cliente, devendo desta forma, encontrar soluções inovadoras para os envolver em novos canais e proporcionarem experiências de qualidade, ao mesmo tempo que monitorizam os resultados do investimento em marketing.

Considerando que, menos de 30% das empresas Fortune 1000 consegue identificar os seus melhores clientes (Wayland & Cole), surge a necessidade de se ter um sistema informático de Gestão de Clientes (conhecido como CRM). Um sistema de CRM permite fazer uma análise cuidada e personalizada de cada cliente de forma individual, aumentando assim o relacionamento entre cliente e empresa, proporcionando uma maior cumplicidade entre os vários canais.

A aposta por parte das empresas num sistema de CRM tem vindo a aumentar ao longo dos anos, começando em 15% das empresas em 2007 até 19% em 2010 (Eurostat). Desta forma, torna-se importante compreender se este sistema é de facto ou não uma mais-valia para as empresas e se, paralelamente, consegue retornar o investimento feito ao longo do tempo.

Durante os anos 90 verificou-se uma mudança numa grande parte das organizações, sendo que estas começaram a sentir a necessidade de gerir relações, ao invés de gerir transações (Light, 2003). Desta forma, enquanto os sistemas de *Enterprise Resource Planning (ERP)* dominaram a era das transações, os sistemas de CRM têm vindo a dominar no que diz respeito a relações (Osarenkhoe e Bennani, 2007). Desde então, a definição de CRM não é o único tópico que tem sido debatido. Também a abrangência e área em que se insere também têm sido discutidas.

A ideia de criar relações com os clientes com base na qualidade, diálogo, inovação e aprendizagem é considerada como uma estratégia mais sustentável, no entanto pode ser facilmente imitada pela concorrência (Payne e Frow, 2006). Surge, então, a necessidade de criar condições de forma a conferir à empresa uma vantagem competitiva. Neste sentido, Nguyen e Mutum (Nguyen e Mutum, 2012) defendem que esta vantagem competitiva pode ser obtida através de um sistema de CRM. Assim, o envolvimento dos clientes irá permitir à organização apreender as necessidades individuais desses mesmos clientes, e conseqüentemente ajustar as suas estratégias de marketing (Frow e Payne, 2009).

Além de CRM, a gestão de clientes é outra expressão que aparece comumente associada. *Frow e Payne (2009)* definem gestão de clientes como uma área que está relacionada com os “*aspectos táticos da implementação do sistema de CRM que estão relacionados com a gestão as interações com clientes, incluindo o uso de ferramentas tais como gestão de campanhas, automação da força de vendas, e gestão de call centers*”.

## 2.1. MICROSOFT DYNAMICS CRM

Ao longo de todo o relatório refiro-me à ferramenta Microsoft Dynamics CRM (MS Dynamics CRM), uma vez que é a ferramenta utilizada pela UNISYS enquanto prestadora de serviços em tecnologias Microsoft na área do CRM.

Com a sua primeira versão lançada em 2003, a Microsoft Dynamics CRM é uma ferramenta de Gestão de relacionamento com o Cliente desenvolvida pela Microsoft. O produto *Out of the box (OOB)* foca-se em três áreas distintas mas que interagem entre si: Vendas, Serviços e Marketing.

No entanto, e apesar de ser um produto que oferece já um diverso e complexo leque de funcionalidades OOB, é a própria Microsoft que incentiva os seus utilizadores e implementarem eles mesmos soluções personalizadas, isto é, desenvolver uma plataforma xRM.

Para que seja possível fazer uma personalização eficaz e dinâmica, a Microsoft lançou em 2011, juntamente com a versão 2011 do MS Dynamics CRM, a funcionalidade de os *developers* criarem as suas soluções.

Soluções são blocos de componentes que derivam de objetos OOB do MS Dynamics CRM, mais qualquer tipo de personalização efetuada sobre o mesmo, possibilitando assim uma maneira de desenvolver sobre a aplicação e de poder mover essas soluções de instância para instância, ou até mesmo de as remover quando já não forem necessárias. Esta forma de gerir as personalizações é a mais utilizada no mundo de desenvolvimento de MS Dynamics CRM.

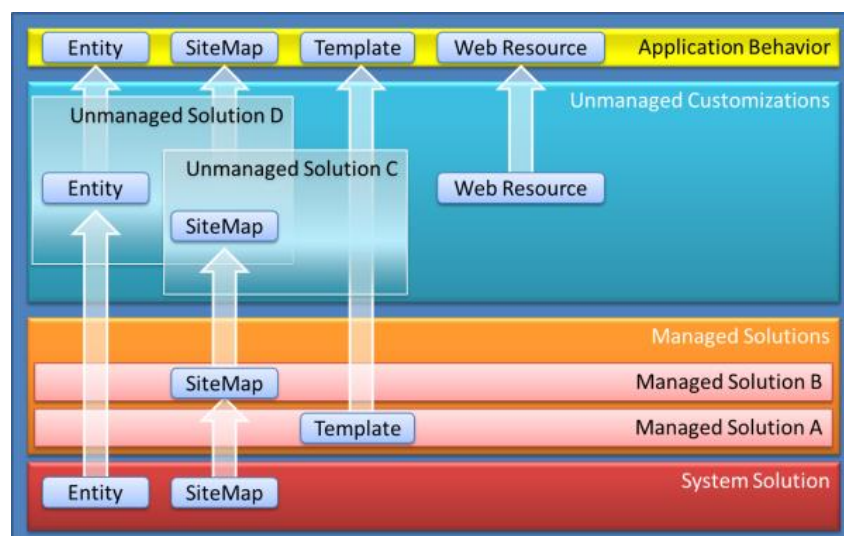


Figura 4 - Estrutura de personalizações de solução MS Dynamics CRM



A personalização da aplicação MS Dynamics CRM pode ser feita em duas vertentes: *Server-side* e *Client-Side*.

A *Server-side* é desenvolvida através da tecnologia C# e implementada na aplicação através de bibliotecas de funções (.dll) em formatos de *Plugin, Workflow, RealTime Workflow* ou *Custom Actions*. Estas customizações podem correr de forma síncrona ou assíncrona e são executadas através de chamada de determinados eventos definidos por quem desenvolveu estas bibliotecas.

Por sua vez, as *Client-side* são desenvolvidas em *Html* e *javascript*. Estas personalizações tendem a ser desenvolvidas com o intuito de serem interpretadas do lado do cliente, servindo muitas vezes como ponte entre a interação do utilizador com chamadas ao servidor. Estas customizações são implementadas na aplicação no seu formato original (.html, .js, .png etc) e são chamadas de *WebResources*.

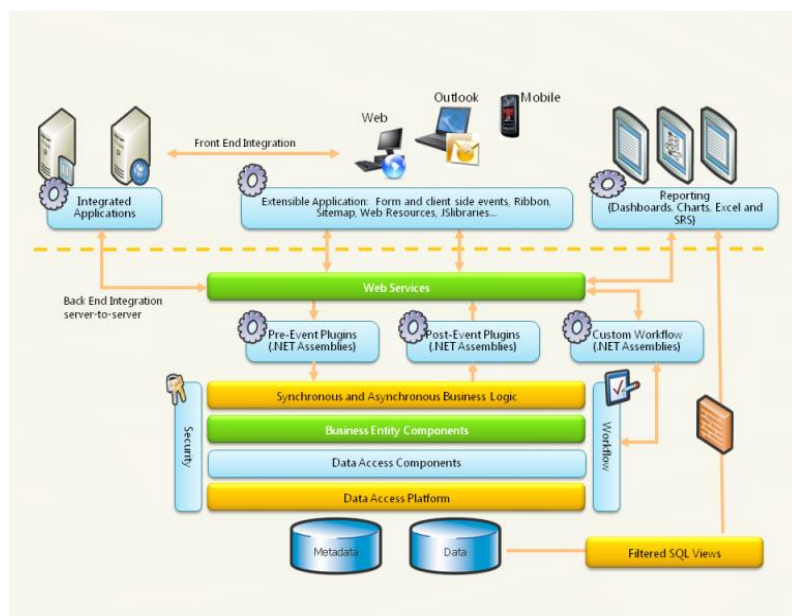


Figura 5 - Estrutura da aplicação MS Dynamics CRM

De forma simplificada, o MS Dynamics CRM é uma aplicação *server-side* isto é, uma aplicação que corre sobre uma *Internet Information Services* (IIS) podendo ser acedida por parte dos utilizadores de diversas formas, desde a utilização de um *browser* a um *plugin* de Outlook com ligação direta ao MS Dynamics CRM.

Apesar da já longa história da aplicação de Gestão de Relacionamento de Clientes da Microsoft (um pouco mais de 10 anos) só no ano de 2011 é que a aplicação se tornou uma referência relativamente aos seus concorrentes diretos, tais como Salesforce, SAP e Oracle. Desde então tem tido um crescimento exponencial e notável no setor da Gestão de Relacionamento de Clientes.

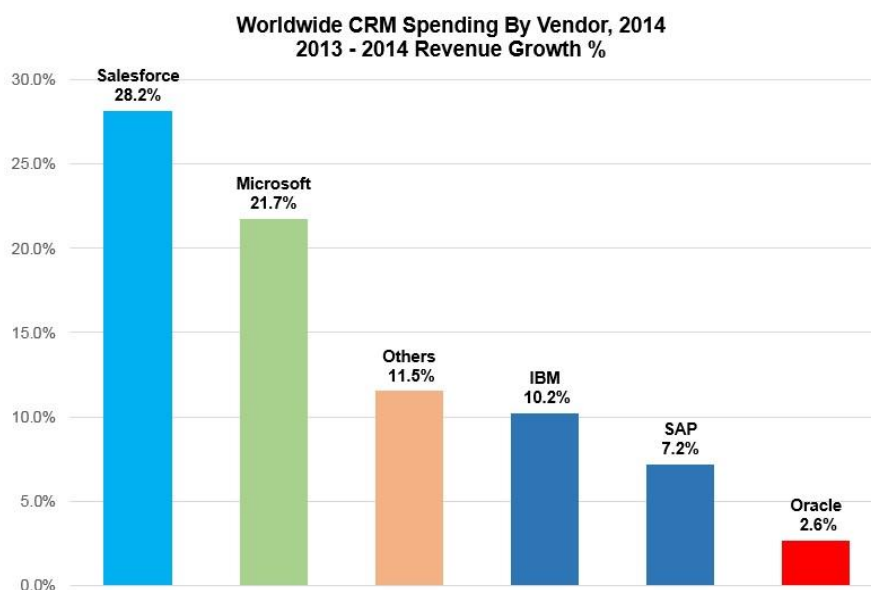


Figura 6 - Crescimento de receitas no ano de 2014 de cada *software* CRM

### 2.1.1. HISTÓRIA DO MICROSOFT DYNAMICS CRM

Como referido no ponto anterior, o MS Dynamics CRM conta já com pouco mais de 10 anos de existência, tendo tido ao longo desse tempo um conjunto de versões disponibilizadas para o mercado, desde a versão 1.0 em 2003, até à esperada e já anunciada versão 2016 para o ano de 2016:

- 2003 – Lançada a primeira versão do MS Dynamics CRM 1.0
- 2003 – Lançada a versão 1.2
- 2005 – Lançada a versão 3.0 (A Microsoft “saltou” a versão 2.0)
- 2007 – Lançada a versão 4.0 (Primeira versão altamente aceiteada pelo mercado das aplicações CRM, atingindo a marca de 1Milhão de utilizadores em 2009)
- 2011 – Lançada a versão 2011
- 2013 – Lançada a versão 2013
- 2015 – Lançada a versão 2015
- 2016 – Esperada a versão 2016 (Por lançar)

### 3. CICLO DE VIDA DE DESENVOLVIMENTO DE SOFTWARE (CVDS)

O Ciclo de Vida de Desenvolvimento de Sistemas (CVDS), do inglês *Systems Development Life Cycle* (SDLC), em engenharia de sistemas, sistemas de informação e engenharia de *software*, é um processo de criação ou alteração de sistemas de informação, bem como os modelos e metodologias que os técnicos utilizam para desenvolver esses sistemas.

- SDLC é também conhecido como *Software development process*;
- O SDLC é uma *framework* que define tarefas que têm de ser executadas em cada fase do desenvolvimento de *software*;
- ISO/IEC 12207 é um *standard* do SDLC. É um *standard* que tem como foco a definição das tarefas necessárias para o desenvolvimento e manutenção de um *software*.

Na *figura 7* é possível identificar as diferentes fases de um típico SDLC:

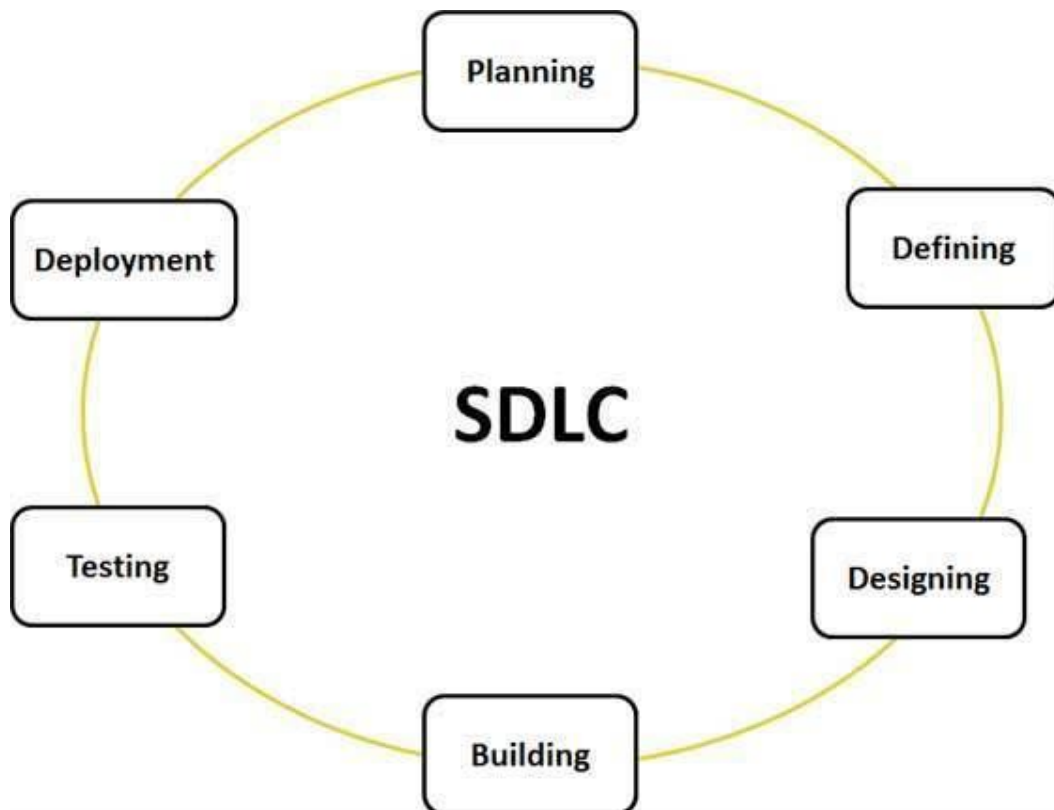


Figura 7 - Ciclo de Vida de Desenvolvimento de *Software*

#### **Fase 1: Planeamento e Análise de Requisitos**

A fase de análise de requisitos é provavelmente, a fase mais importante e fundamental no Ciclo de Vida de Desenvolvimento de Software. Esta fase é normalmente executada por alguém já com alguma experiência, em que o contacto com o cliente é gerido de forma a entender e garantir que as suas necessidades sejam compreendidas e possíveis de implementar. A informação proveniente deste

levantamento é fundamental para uma validação de viabilidade da solução e planeamento do projeto correto em termos económicos, operacionais e técnicos.

O planeamento para a garantia de qualidade e identificação dos riscos (também é realizado nesta fase). O resultado do estudo de viabilidade técnica implica definir as várias abordagens técnicas que devem ser seguidas a para que, com o mínimo de riscos associados, a implementação do projeto seja um sucesso.

## **Fase 2: Definição dos Requisitos**

Uma vez concluída a análise de requisitos, o próximo passo é definir claramente e documentar os requisitos do produto e levá-los a aprovação por parte do cliente. Este procedimento é apresentado no documento de Especificação de Requisitos de Software (daqui adiante referenciado como ERS) e consiste no conjunto dos requisitos do produto a serem desenhados e desenvolvidos durante o projeto de implementação.

## **Fase 3: Desenho da Arquitetura do Sistema**

Com base nos requisitos especificados no ERS, geralmente mais de uma abordagem de *design* para a arquitetura do produto é proposto e documentado em um Documento de Especificação (daqui adiante referenciado como DE).

Este DE é validado por todas as partes interessadas no projeto e, com base em vários parâmetros como a avaliação de risco, robustez do produto, restrições de orçamento e de tempo, é selecionada a melhor arquitetura para o desenvolvimento do sistema.

## **Fase 4:Desenvolvimento**

É nesta fase do Ciclo de Vida de Desenvolvimento do Software que se começa a desenvolver o produto final. Quanto mais detalhado for o desenho da solução, mais facilmente será desenvolvido o produto. Este será desenvolvido através de código que será interpretado por um compilador. A linguagem a ser escolhida para este desenvolvimento terá de ser cuidadosamente escolhida de forma a garantir a correta implementação bem como desempenho do produto.

## **Fase 5: Testes**

A fase de testes corresponde ao momento em que são reportados, detetados, corrigidos e testadas as não conformidades de requisitos, até que o produto final alcance a qualidade definida previamente no documento ERS.

## **Fase 6: Passagem a Produção e Manutenção**

Uma vez testado e pronto a ser colocado em produção, é preparada uma *release* do sistema. De uma forma geral, a primeira *release* é feita num ambiente controlado e mais restrito onde são realizados os testes de aceitação por parte dos utilizadores.

Após a passagem pelos testes de aceitação, o produto poderá ser exposto de novo a um ciclo de melhorias até que atinja o nível desejado, altura em que é será, colocado em produção. Seguem-se as atividades de suporte necessárias ao seu correto funcionamento.

### 3.1. MODELO CASCATA

Modelo Cascata, ou em inglês *Waterfall* (como será mencionado daqui adiante), foi um dos primeiros modelos de desenvolvimento formal de *software* a ser criado, é um modelo muito fácil de entender e que tem por base um ciclo de vida sequencial, o que significa que cada fase deverá de estar completa antes de se avançar para a seguinte.

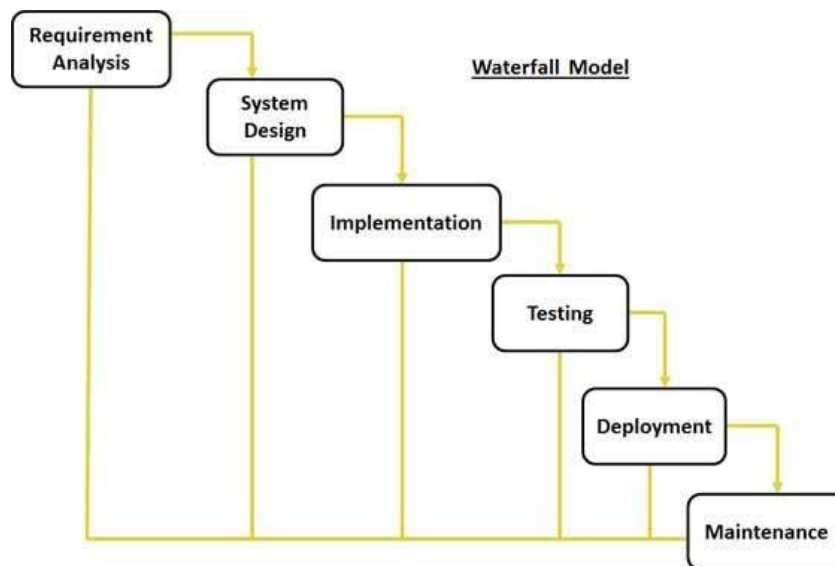


Figura 8 - Esquema do modelo *Waterfall*

Este, foi o primeiro modelo a ser amplamente adotado na Engenharia de Software com o objetivo de garantir o sucesso de um projeto. Este modelo divide o Ciclo de Vida de Desenvolvimento de Software em diferentes fases, em que o output de uma funciona como *input* da seguinte.

O *Waterfall* é dividido nas seguintes etapas:

- **Análise de Requisitos**

Nesta fase realiza-se o levantamento de todos os requisitos do sistema ser desenvolvido, bem como a sua documentação num ERS.

- **Desenho do Sistema**

A fase do desenho do Sistema recebe como *input* o ERS proveniente da fase anterior, esta fase é importante pois ajuda a especificar o hardware necessário bem como a arquitetura geral do sistema.

- **Implementação**

Com *inputs* vindos da fase do desenho de sistema, é neste momento que se inicia o desenvolvimento do sistema dividido em partes pequenas (chamadas de unidades), que serão integradas na fase seguinte. Cada unidade é desenvolvida e testada de forma singular (testes unitários).

- **Integração e Testes**

Todas as unidades desenvolvidas na fase anterior são integradas num único sistema, onde são executados testes globais.

- **Passagem a Produção**

Após a passagem e aprovação dos testes, o sistema é colocado em ambiente de produção, ou colocado em comercialização.

- **Manutenção**

Após a passagem a produção por vezes aparecem erros que têm de ser resolvidos, para tal são lançados *patches* de correção. No entanto também é possível serem lançados *patches* de melhoria do sistema, com mais funcionalidades ao mesmo.

O modelo *Waterfall* deverá de ser utilizado quando:

- Os requisitos estão bem definidos;
- Domínio sobre a tecnologia a ser utilizada;
- Projeto curto.

VANTAGENS	DESVANTAGENS
Controlo sobre o processo desenvolvido	Rigidez e Inflexibilidade
Deadlines específicos para cada fase	Não suporta modificações nos requisitos

Tabela 1 - Vantagens e Desvantagens do Modelo *Waterfall*

### 3.2. MODELOS ITERATIVOS

Nos modelos iterativos o processo começa com uma simples implementação de um pequeno conjunto dos requisitos do *software*, e de forma iterativa este vai sendo analisado de maneira a identificar os restantes requisitos. Este processo é então repetido, produzindo uma nova versão do *software* cada vez que é concluído uma iteração até que seja produzida a versão final do software que responde a todos os requisitos.

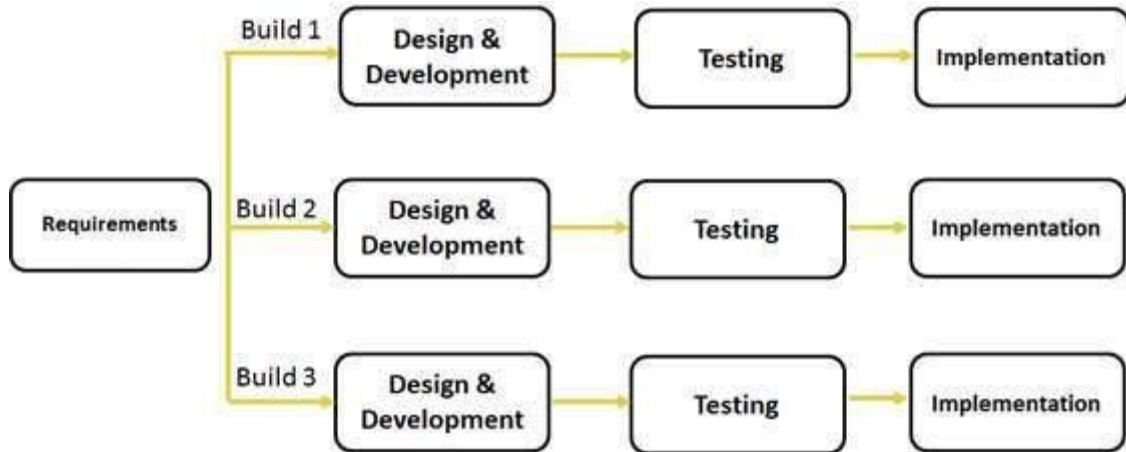


Figura 9 - Esquema do Modelo Iterativo

O fator principal para atingir o sucesso com estes modelos passa por uma validação rigorosa dos requisitos bem como uma verificação e bateria de testes sobre cada versão de *software* desenvolvida de forma a garantir que sejam respondidos os requisitos definidos. À medida que vão sendo efetuados mais ciclos de desenvolvimento, torna-se necessário garantir que a nova versão também passe nas validações e testes da anterior.

Tal como qualquer outro modelo de desenvolvimento de *software*, os modelos Iterativos são recomendados quando nos deparamos com os seguintes cenários:

- Requisitos bem definidos;
- Utilização de uma nova tecnologia.

VANTAGENS	DESVANTAGENS
<i>Software</i> funcional após cada ciclo de vida	Só aplicável a grandes projetos
Facilidade de correção de erros	Maior complexidade na gestão do projeto
Validação por parte do cliente	Não se sabe qual será o produto final
Baixo custo na alteração dos requisitos	

Tabela 2 - Vantagens e Desvantagens do Modelo Iterativo



### 3.3. PROTOTIPAGEM

O modelo de desenvolvimento de *software* de Prototipagem, tem vindo a tornar-se muito popular na medida em que permite entender os requisitos do cliente logo numa fase inicial do desenvolvimento, isto devido ao *feedback* fornecido por este, em cada protótipo produzido, o que ajuda a entender, desenhar e a desenvolver o produto final que realmente é esperado.

Mas afinal, o que é Prototipagem? Resumidamente um protótipo é um pedaço de *software* com limitações funcionais. Esta abordagem é utilizada para possibilitar ao utilizador verificar o que foi desenvolvido antes de ser implementado, e também para ajudar quem está a desenvolver, validar se foi feito um correto levantamento dos requisitos.

O desenvolvimento no modelo de Prototipagem é realizado nas seguintes fases:

- **Identificação de Requisitos básicos**

Nesta fase são identificados os requisitos mais básicos do *software* a ser desenvolvido, como por exemplo um simples interface, onde pode ser deixado de parte o levantamento de requisitos como *performance* ou *segurança* do *software*.

- **Desenvolvimento do primeiro Protótipo**

É nesta fase que é desenvolvido o primeiro Protótipo onde os requisitos básicos, como o interface, são apresentados pela primeira vez ao cliente. Neste pedaço de *software*, por vezes o funcionamento pode não estar correto e até mesmo estarem a ser utilizados *work arounds* para que seja dada a ideia do que pode vir a ser o *software* final.

- **Revisão do Protótipo**

É neste momento em que o Protótipo desenvolvido é então apresentado aos *stakeholders* do projeto. Derivado da apresentação é recolhido um conjunto de *feedbacks* para que se tenha em conta no próximo Protótipo.

- **Melhoramento do Protótipo**

Após a recolha do *feedback* fornecido pelos *stakeholders* é iniciada a negociação, tendo em contas as limitações de tempo, custos e de esforço de implementação das sugestões dadas. As mudanças acordadas serão então incorporadas no novo Protótipo, repetindo-se o ciclo até que sejam alcançadas as expectativas do cliente

O modelo de desenvolvimento de *software* de Prototipagem pode ser feito de diversas maneiras, tais como:

- **Prototipagem Rápida**

Consiste em desenvolver um Protótipo com baixo esforço e com pouca atenção aos requisitos. Este Protótipo é então analisado e após a compreensão clara dos requisitos é deitado fora e é desenvolvido um do início de forma limpa e concreta de forma a responder às necessidades do cliente.

- **Prototipagem Evolucionária**

Esta forma de prototipagem baseia-se no desenvolvimento de um Protótipo com poucas funcionalidades no início, com o objetivo de servir de base de evolução para os restantes Protótipos. Com esta abordagem de prototipagem vão sendo implementados os requisitos que são entendidos ao longo de cada Protótipo.

- **Prototipagem Incremental**

Consiste na integração de todos os Protótipos funcionais desenvolvidos até à data num único sistema, de forma a construir o sistema completo.

<b>VANTAGENS</b>	<b>DESVANTAGENS</b>
Alto envolvimento do cliente com o Projeto	Possibilidade de aumento do scope do projeto
Mais fácil entendido do <i>software</i> por parte do Cliente	Tendência a reutilizar protótipos anteriores mesmo quando não recomendado
Falha no levantamento dos requisitos pode ser facilmente identificada	Risco de se gastar demasiado esforço no desenvolvimento de um protótipo

Tabela 3 - Vantagens e Desvantagens do Modelo de Prototipagem

## 4. MODELOS ÁGEIS

Em 2001, foi publicada uma declaração de princípios que fundamentam o desenvolvimento ágil de *software*, o Manifesto Ágil.

“Ao desenvolver e ao ajudar outros a desenvolver *software*, temos vindo a descobrir melhores formas e o fazer. Através deste processo começámos a valorizar” (AgileManifesto):

- Indivíduos e interações acima de processos e ferramentas
- *Software* funcional acima de documentação detalhada;
- Colaboração com o cliente acima da negociação do contrato
- Capacidade de responder à mudança acima de execução de um plano pré-definido.

### 4.1. FUNDAMENTOS

O modelo de SDLC Ágil (do inglês, e daqui adiante referenciado como Agile) é uma combinação do modelo iterativo e incremental com foco na adaptação do processo e satisfação do cliente através de entregas rápidas de *software* funcional.

Este modelo divide o produto em pequenos desenvolvimentos incrementais, divididos por iterações. Cada iteração tem uma duração de uma a três semanas, em que a equipa funcional é envolvida em várias áreas, como: Planeamento, Análise de Requisitos, Desenho, Desenvolvimento, Testes e Aceitação, sendo que após a passagem por todas estas etapas, o produto elaborado ao longo da iteração é entregue aos *stakeholders*.

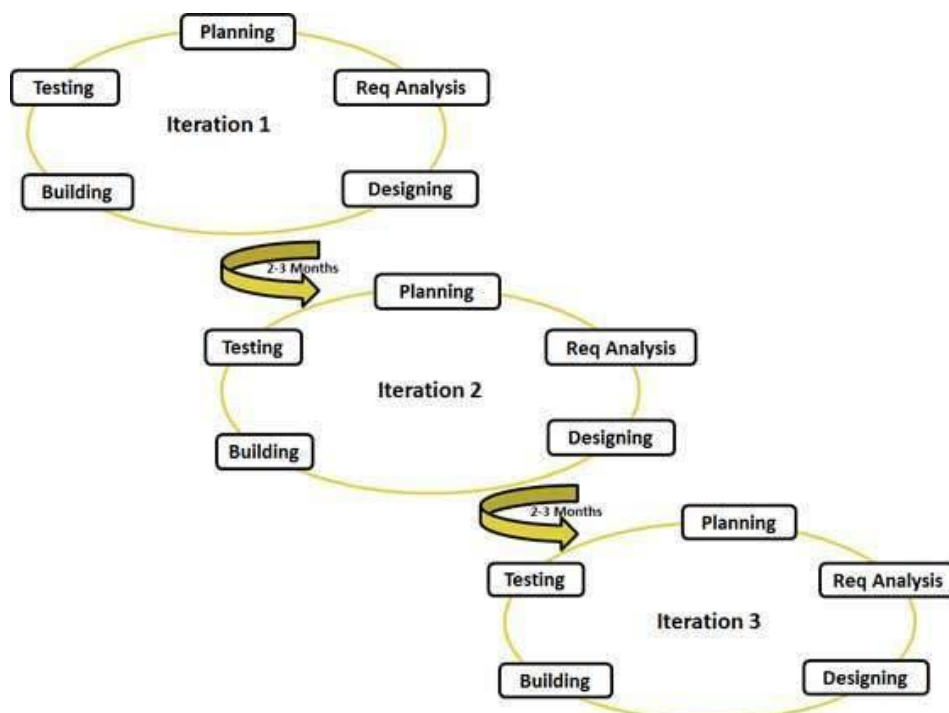


Figura 10 - Modelo Ágil

## 4.2. ÁGIL VS MODELOS TRADICIONAIS SDLC

A grande diferença entre modelos ágil e modelos tradicionais de desenvolvimento de *software*, é o facto de o primeiro se basear na capacidade de adaptação no desenvolvimento de software, enquanto o segundo, como o modelo *Waterfall*, se baseia numa perspetiva prescritiva.

Uma abordagem prescritiva funciona normalmente sobre um plano definido, com uma previsão detalhada das tarefas a serem entregues nos meses seguintes ou durante o ciclo de vida do produto. Estes modelos dependem apenas da análise de requisitos e planeamento do projeto elaborados no início do ciclo, tornando qualquer tipo de alteração complicada de gerir.

Por sua vez, modelos ágeis usam metodologias versáteis onde não há um planeamento rigoroso nem definição de futuras tarefas a serem executadas. Apenas se sabe quais as funcionalidades a serem desenvolvidas para o produto final. Estes modelos são orientados para o desenvolvimento de funcionalidades, em que as equipas vão se adaptando de forma dinâmica à mudança de requisitos ao longo do ciclo. É ainda feito um controlo rigoroso e repetido aos produtos de cada iteração, diminuindo assim a probabilidade de se vir a detetar um eventual erro grave no fim dos desenvolvimentos.

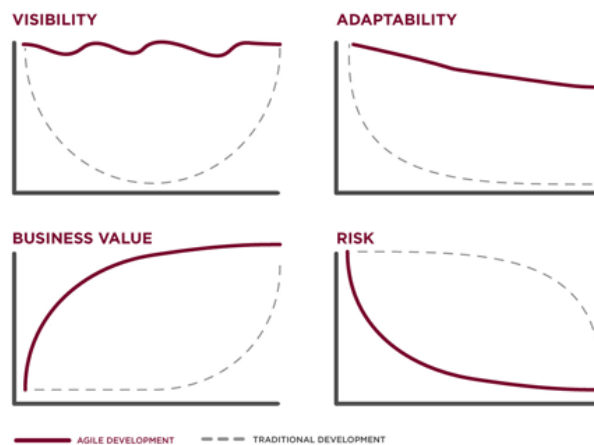


Figura 11 - Modelo Agile vs Modelos tradicionais

### 4.3. MODELO DE GESTÃO SCRUM

Desenvolvido nos anos 90, SCRUM é uma *framework* que tem sido utilizada para gerir o desenvolvimento de produtos complexos. Apesar de ser um modelo de gestão, SCRUM trata-se de uma das *frameworks* mais famosas de implementação *agile*, de tal maneira, que muitos pensem que SCRUM e *agile* sejam a mesma coisa. Pondo de parte essa confusão, SCRUM é uma ótima metodologia para que as equipas se habituem e comecem a trabalhar com metodologias ágeis. O que torna esta *framework* tão especial e única é o facto de a mesma oferecer uma metodologia de gestão composta por *sprints*. No entanto, antes de entrarmos em detalhes, é necessário fazer um contexto teórico de como é que está organizada.

A metodologia SCRUM divide o seu CVDS em 5 fases: *Product Backlog*, *Sprint Backlog (To do)*, *In Progress*, *To Verify* e *Done*.

- ***Product Backlog***

O *Product Backlog* é uma lista de funcionalidades, requisitos, melhorias e correções necessárias a desenvolver para uma release futura do produto.

- ***Sprint Backlog***

O *Sprint Backlog* é o conjunto de itens do *Product Backlog* seleccionados para a execução de um *Sprint*.

- ***In Progress***

Nesta fase estão os itens vindos do *Sprint Backlog* que estão a ser desenvolvidos.

- ***To Verify***

Após a fase de desenvolvimento, os *ítems*, passam por um conjunto de testes em que é validado o seu funcionamento.

- ***Done***

Com a validação concluída com sucesso, os *ítems* chegam agora, à fase do *Done*. O processo de desenvolvimento do produto só fica concluído quando todos os *ítems* tiverem chegado a esta fase, até lá, vão sendo executados vários *Sprints*.

A Figura 11 mostra um exemplo de *items* organizados ao longo das várias fases do SCRUM.



Figura 12 - Fases SCRUM

Como referido no início deste capítulo, o que torna especial o SCRUM é o facto de este se basear em *Sprints*, isto é, são seleccionados um conjunto de *items* do *Product Backlog* para um *Sprint Backlog* que serão executados num espaço de tempo bem definido (*Sprint*), geralmente entre três a quatro semanas. Durante esse *Sprint*, vão sendo feitas reuniões diárias de cerca de 15m, de forma a toda a equipa tenha a visão e conhecimento do que foi feito no dia anterior e quais os planos para o próprio dia. Os *items* desenvolvidos vão avançando pelas fases do SCRUM até chegarem todos à *Done*, o que leva à conclusão do *Sprint* atual.

Este modelo de execução dividido por *Sprints*, fornece às equipas uma metodologia que as incentiva a trabalhar por objetivos, em conjunto, bem como a um sentimento de satisfação no final de cada *Sprint*.

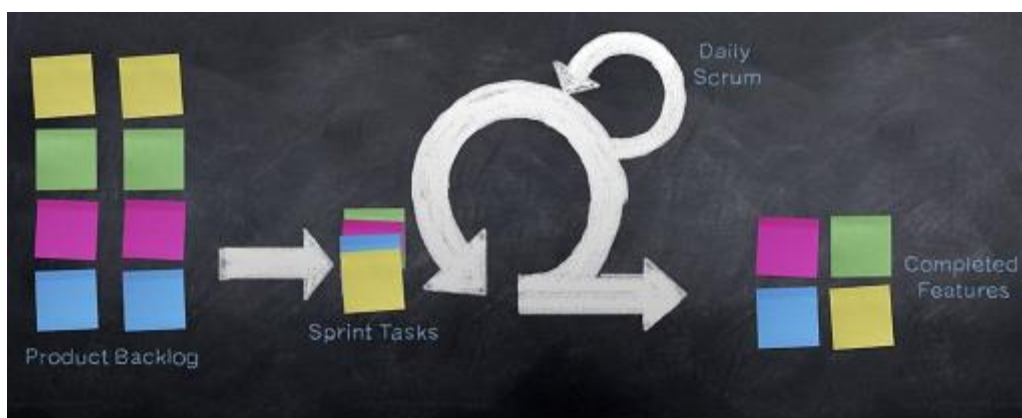


Figura 13 - *Sprint* SCRUM

## 5. FRAMEWORK DE GESTÃO DE SKILLS (SFIA)

De forma a ser possível fazer uma avaliação correta das tarefas desempenhadas ao longo do estágio, utilizarei como base *Skills Framework For The Information Age* (daqui adiante mencionada SFIA) para uma análise mais detalhada (mais adiante), neste relatório de estágio.

### 5.1. CONTEXTO

*“As IT gets more complex and pervasive there is an ever growing need clearly to define, recruit and grow the skilled resources that you need. SFIA provides a language that is the foundation for consistent, unambiguous and clear definitions of IT based skills.”* (SFIA Framework Reference)

A *framework* SFIA é o resultado de um trabalho desenvolvido entre o governo do Reino Unido, universidades, sindicatos e empresas de Sistemas de Informação (SI) (*Intel Diálogo TI*), com o objetivo de definir as descrições de todos os postos de trabalho de uma área ou de uma organização de SI.

A sua estrutura obedece a uma lógica simples representada em duas dimensões. A primeira, mede as *skills* dos profissionais de SI e divide-se em seis categorias: Estratégia e arquitetura; Mudança no negócio; Desenvolvimento e aplicação de soluções; Administração de serviços, Suporte no fornecimento e na administração e Interface com o cliente, enquanto a segunda dimensão analisa os seus níveis de maturidade.

O diagrama apresenta a estrutura SFIA, organizada em uma tabela com 6 linhas de categorias de habilidades e 7 colunas de níveis de maturidade. Cada categoria é representada por uma cor específica e um ícone de dois quadrados. O cabeçalho da tabela indica 'Níveis de maturidade' e as colunas são numeradas de 1 a 7. O logotipo SFIA está no canto superior esquerdo.

		Níveis de maturidade						
Categorias de habilidades		1	2	3	4	5	6	7
	Estratégia & arquitetura							
	Mudança no negócio							
	Desenvolvimento e aplicação de soluções							
	Administração de serviços							
	Suporte no fornecimento e na administração							
	Interface com o cliente							

Figura 14 - Estrutura SFIA

## 5.2. SKILLS E NÍVEIS DE RESPONSABILIDADES

O SFIA é um conjunto de competências (*skills*) empresariais genéricas divididas ao longo de sete níveis diferentes, sete níveis genéricos que são aplicados ao local de trabalho. Cada nível tem uma definição completa expressa em termos de autonomia, complexidade, influência e competências empresariais. Por outras palavras, os sete níveis representam o grau de responsabilidade de cada trabalhador.



Figura 15 - Os sete níveis SFIA

## 5.3. SKILLS PROFISSIONAIS, O QUE SÃO E O QUE NÃO SÃO

SFIA tem como objetivo fornecer uma *framework* de gestão para ajudar aqueles que tomem decisões sobre o uso ou desenvolvimento de *skills*. Por esta razão, as definições de *skills* são gerais e não prescritivas, o que significa que contêm informação suficiente para permitir uma gestão racional a fim de saber se alguém a possui, e em caso afirmativo a que nível. Estas definições, não têm como objetivo enumerar todas as tarefas que um trabalhador qualificado possa ser capaz de executar, no entanto, é importante salientar que as definições fornecem *guidelines* precisas dos vários níveis de *skills* necessários.

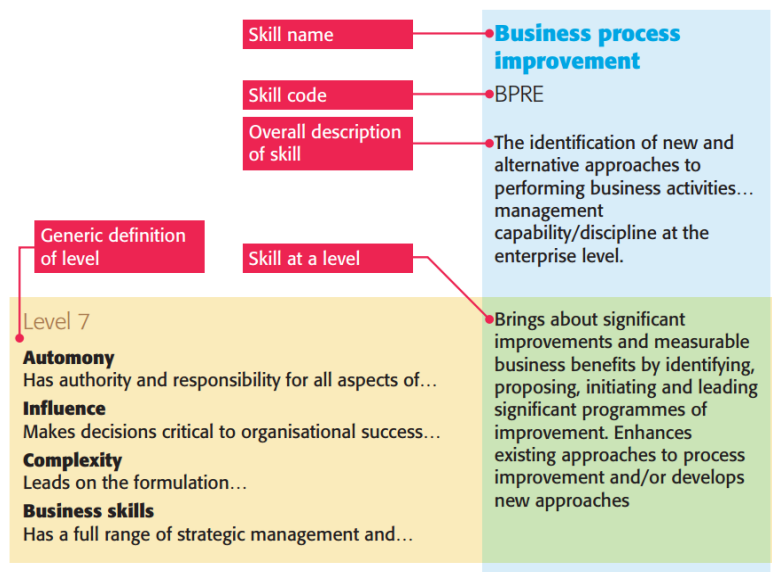


Figura 16 - Esquemático sobre organização de uma *skill*



A indústria de TI contém uma riqueza de informações, formal e informal, que suporta cada *skill*, isto abrange muitos aspetos complexos, processos e métodos que possam relacionar-se com a *skill* em concreto. O propósito SFIA não é incluir toda essa informação, mas para proporcionar uma *framework* de gestão que ajude os gestores a entenderem esta complexidade.

As descrições SFIA não são, de forma geral, em termos de tecnologias ou produtos, pois estas não descrevem processos, trabalhos, áreas gerais da atividade, ou mesmo partes de uma organização – são apenas *skills*.

## 6. TECNOLOGIAS

Enquanto Consultor Júnior, e de forma a conseguir desempenhar algumas das tarefas ao longo do meu estágio, tive a necessidade de aprender um conjunto de tecnologias. Para que fosse possível fazer customizações na aplicação do MS Dynamics CRM, era necessário dominar linguagens de programação orientadas a objetos (OOP), linguagens Web e linguagens script.

De seguida será apresentado um pequeno resumo de algumas dessas tecnologias que tive como foco e onde evolui os meus conhecimentos.

### 6.1. .NET C#

C# é uma linguagem de programação orientada a objetos (OOP) desenvolvida pela Microsoft durante a criação da *framework* .NET. É uma linguagem de aplicações gerais utilizada para o desenvolvimento de *Web Services*, aplicações Desktop, serviços Windows, etc.

Como referido no ponto 2.2 a base da aplicação Microsoft Dynamics CRM é o C#, o que leva a que seja necessário um conhecimento desta linguagem para que seja possível fazer customizações na aplicação. Estas customizações são desenvolvidas no *Visual Studio* e depois feitas o *deploy* nos servidores onde esteja instalada a aplicação MS Dynamics CRM, em forma de *Plugin* ou *Workflow*.

### 6.2. JAVASCRIPT

O javascript, tal como o nome indica, é uma linguagem de script, isto é, uma linguagem que corre sobre uma aplicação e que é executada do lado do cliente “*server-side*” sem que o script tenha de passar pelo servidor.

Quando é necessário fazer customizações na aplicação MS Dynamics CRM do lado do cliente, é utilizado javascript (js), em que são desenvolvidos scripts que interagem com o utilizador, servindo por vezes de ponte entre o utilizador e o servidor, em que é executada uma chamada do lado do cliente para a execução de um determinado comando do lado do servidor. Estes scripts são desenvolvidos em qualquer editor de texto e inseridos no servidor em formato de *WebResource*, podendo ser utilizados nos contextos em que forem definidos.

### 6.3. HTML

HTML é uma sigla inglesa para “*HyperText Markup Language*”, que é utilizada como linguagem de programação de páginas Web. A estrutura de um documento .html é composta por *tags*; essas *tags* representam a função que cada elemento tem no documento, como por exemplo: <h> para um *header*, ou um <p> para um parágrafo.

Para as customizações de MS Dynamics CRM é possível fazer o desenvolvimento de páginas .html e inseri-las na aplicação, de forma a proporcionar uma maior versatilidade de funções disponíveis ao *enduser*, mas também para em alguns casos, tornar a usabilidade mais agradável à vista.

## 6.4. CSS

CSS é a abreviatura para a sigla inglesa de “*Cascading Style Sheets*”, que representa uma linguagem de estilos, em que são definidos *layouts* de documentos HTML.

Depois de desenvolvidas as páginas .html a serem inseridas na aplicação de MS Dynamics CRM, por vezes torna-se necessário criar .css dedicados às páginas, para que estas tenham os estilos desejados, ou até mesmo para garantir que a customização da página .html e que funcione em qualquer dispositivo com acesso à internet.

## 7. ATIVIDADES DESENVOLVIDAS

Desde o primeiro dia, e durante todo o período de estágio na UNISYS, é inculcido, a todos os colaboradores, como missão principal, a entrega de produtos e serviços com nível de qualidade elevada, assegurando as necessidades e expectativas do cliente, em tempo e custo definidos no início do projeto.

Como consultor júnior foi-me dada a oportunidade de integrar a já conhecida e conceituada área de desenvolvimento da UNISYS em Microsoft Dynamics CRM. Ao longo do Estágio foi-me possível contactar com diversos setores e projetos, no entanto serão apenas mencionados dois desses projetos, em que estive envolvido desde o início.

De forma a organizar o relatório, a apresentação do relatório e facilitar o seu entendimento, os projetos referidos serão descritos em três partes: Contexto, Desafio, Tarefas e *Skills*, em que no contexto será apresentado o projeto em si e quais os seus objetivos, bem como a minha participação em cada um deles, o desafio e problemas encontrados ao longo do projeto para a empresa, e para mim enquanto consultor júnior, e por fim as tarefas e *skills*, às irei fazer uma breve abordagem.

## 7.1. PROJETO – IMPLEMENTAÇÃO DE SOLUÇÃO XRM

Acabado de entrar no mundo empresarial, e com apenas um dia de palestras sobre o funcionamento e regras gerais de privacidade da UNISYS, foi-me apresentado o meu primeiro desafio da carreira, fui colocado diretamente no cliente sem qualquer tipo de formação inicial.

Tratava-se de um cliente já com experiência em soluções MS Dynamics CRM, que queria de certa forma não só fazer um *update* à versão que tinha em uso, bem como desenvolver novas funcionalidades de forma a garantir o preenchimento das suas necessidades de negócio com uma ferramenta de CRM. O cliente tinha neste caso, a necessidade de um sistema de gestão de concursos e de atribuição de bolsas aos vencedores. Foi então aproveitado o facto de ser preenchida esta necessidade que surgiu a possibilidade de, também, executar um projeto de apoio à gestão interna do Cliente

Este projeto foi concebido para ser desenvolvido em duas fases distintas, a primeira (na qual participei e irei desenvolver mais detalhadamente) um projeto de carácter mais interno, isto é, para uso interno do cliente sem qualquer tipo de contacto com o exterior, em que foi elaborada toda a gestão de reuniões e atas (SIGA – Sistema Integrado de Gestão de Atas), e a segunda parte do projeto, um sistema de gestão de concursos e atribuição das respetivas bolsas (SIPP – Sistema Integrado de Sistemas e Projetos).

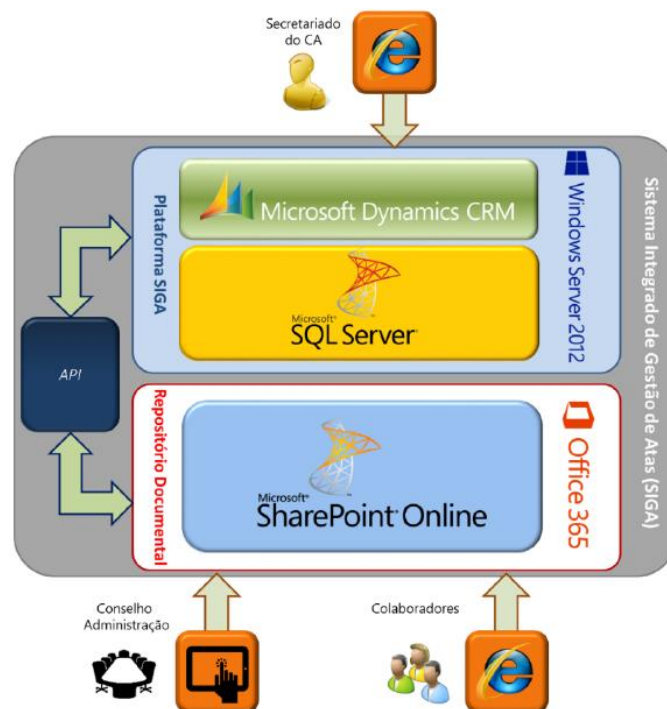


Figura 17 - Esquémático da plataforma de suporte do Sistema Integrado de Gestão de Atas (SIGA)

A primeira parte do projeto e em que participei (SIGA) consistiu no desenvolvimento de uma solução que permitisse à organização fazer uma gestão de reuniões e atas, isto é, uma aplicação que assegurasse agendar reuniões e organizar os seus pontos de agenda, com diversos órgãos da empresa (nacionais e internacionais) em que os mesmos eram notificados e confirmavam ou infirmavam a sua presença na reunião.

Esta aplicação tinha ainda como requisito fornecer uma ferramenta para elaborar as atas das reuniões de forma automática e com as respetivas deliberações dos pontos de agenda. A produção de atas teria e em simultâneo, uma integração com o SharePoint Online que serviria como aplicação de gestão documental das atas produzidas.

Em suma, tratava-se de um projeto claro de xRM com integração entre Microsoft Dynamics CRM e SharePoint Online.

### 7.1.1. DESAFIO

Como mencionado no ponto 5.2.1, entrei diretamente para o cliente sem qualquer formação sobre a ferramenta MS Dynamics CRM, o que me deixou um pouco preocupado sobre qual seria o meu papel, bem como quais as exigências e expectativas que teriam sobre mim. No entanto, logo no primeiro dia de trabalho, deparei-me com dois profissionais que me ajudaram desde o primeiro minuto a integrar-me e a sentir-me confortável. A equipa era então constituída pelos três, eu, um consultor e o meu Gestor de Projeto.

Foi então que iniciei a minha formação *on the job*. Entregaram-me uma vasta documentação sobre o funcionamento da ferramenta MS Dynamics CRM. Apesar de não ter conhecimentos sólidos sobre a ferramenta, o meu primeiro desafio foi, de facto, entender o porquê de a utilização de uma ferramenta de gestão de clientes para um sistema de gestão de agendas e atas. Este conceito pode de facto parecer um pouco estranho e até não ser compreensível por parte das pessoas (tal como eu não o entendia), no entanto o meu Gestor de Projeto teve o cuidado de perder algum tempo (ao longo de vários dias) para me passar a informação necessária para que me fosse possível entender do tema em causa e o porquê da escolha da ferramenta MS Dynamics CRM para este caso em concreto.

Foi nessa altura que me foi introduzida a expressão xRM, que por outras palavras significa: *Anything (X) Relationship (R) Management (M)*. A expressão xRM é uma nomenclatura muito utilizada pelos consultores MS Dynamics CRM para quando se querem referir a um nível de customização da ferramenta que deixa de ser exclusivamente orientada a clientes e passa a ser extensível para outras necessidades.

*“Software developers and business analysts commonly define xRM in one of two ways. In the first definition, the “x” in xRM refers to extended relationship management, which represents the extension of CRM platforms far beyond customer relationship management. In the second definition, the “x” is an algebraic variable that can represent almost any relationship that a business needs to manage. In both definitions, expanding the functionality of the CRM platform is the key. An xRM solution can manage more than relationships with customers; it can manage suppliers, employees, partners, assets, knowledge bases, and just about anything else a company might wish to manage in a relational database.” (xRM defined by xRM)*

Todo este conceito de transformação e adaptação da ferramenta conforme as necessidades foi algo que me fascinou e entusiasmou desde o primeiro dia que me foi apresentada. Apesar de ter toda a documentação para ler, de todo o entendimento do conceito xRM, bem como o facto de não ter qualquer tipo de experiência profissional, nunca me impediu, nem nunca foi pretexto para que tivesse sido posto de parte nos momentos de discussão e análise de requisitos juntamente com o cliente, o que me agradou bastante pois senti que eram momentos únicos em que tive a possibilidade de ter os primeiros contactos com o cliente, e de acima de tudo, começar a sentir os desafios presentes durante um verdadeiro projeto.

Tendo em conta todo o contexto envolvente do projeto, foi decidido utilizar a metodologia de gestão Scrum. Para o sucesso de implementação desta metodologia foi utilizada uma matriz dividida pelas seguintes colunas: *Backlog*, *Sprint*, *Doing*, *Test*, *Done*.

BACKLOG	SPRINT	DOING	TEST	DONE

Figura 18 - Tabela das diferentes fases da metodologia SCRUM

De forma a ser possível uma organização eficaz, foi feito um levantamento das tarefas a serem desenvolvidas e colocadas na primeira coluna, o *Backlog*. De seguida foi escolhida uma serie de tarefas a serem executadas e foram colocadas na coluna seguinte, *Sprint*. Durante os quinze dias seguintes, essas tarefas iam sendo colocadas no *doing* enquanto estavam a ser executadas, passavam para *test* quando finalizadas, e só depois de passarem a fase de testes é que iam sendo colocadas na coluna final de *Done*.

Esta metodologia foi bastante útil para entender em que ponto nos encontrávamos ao longo dos quinze dias de cada *sprint*, pois permitia uma leitura rápida de qual a situação em que nos encontrávamos diariamente.

A utilização desta metodologia revelou-se um sucesso ao longo do projeto, pois à medida que a equipa foi crescendo, era fácil para os novos elementos entenderem as tarefas que tinham de realizar, quais as suas dependências, e acima de tudo, uma visão geral de toda a equipa e do *status* do sprint ao longo do tempo. Esta metodologia ajudou a que o projeto fosse concluído com sucesso, isto é, com um alto nível de qualidade e dentro dos prazos e custos definidos.

O desafio foi portanto, muito grande neste primeiro projeto em que fui integrado, A que a minha principal responsabilidade foi aprender de forma rápida e consistente, não só a ferramenta em si, mas também a lidar com o *stress*, pressão e níveis de exigência elevados.

Tendo em conta o fator de ainda estar a estudar (finalizar o último ano de Mestrado) foi também um desafio; conseguir conciliar as duas vertentes, trabalho e faculdade. Este desafio terá sido de certa forma o mais duro e complicado de gerir, isto porque não só foi bastante cansativo e desgastante trabalhar de dia e estudar de noite, como muito stressante, pois tive como objetivo principal a finalização do curso, sem nunca querer deixar algo no trabalho por fazer ou feito com menor qualidade.

### **7.1.2. TAREFAS E SKILLS**

Com função de consultor Júnior e estando eu a começar a minha vida profissional, tendo muito material para ler e estudar antes de começar a ser uma mais-valia ativa para o projeto e para a empresa, desempenhei ao longo do projeto diversas tarefas, tais como: Análise e Desenho, Desenvolvimento, Manutenção Corretiva e Suporte.

- **Análise e desenho**

Desde o primeiro dia que integrei a equipa, foram-me colocados desafios de entendimento e análise de requisitos provenientes do cliente. Esta análise era feita em equipa na qual eu tive sempre a oportunidade de participar na decisão. Após esta análise de requisitos era feito o desenho da aplicação a implementar e era validado entre todos os membros da equipa, para que todos a entendessem, mas que acima de tudo, para que todos pudessem analisar e contribuir para um desenho melhor e mais eficaz.

- **Desenvolvimento**

Terá sido na tarefa de desenvolvimento que tive a minha maior participação neste primeiro projeto. Como o próprio nome da tarefa o indica, tive como objetivo o desenvolvimento e de *software* dedicado à aplicação de MS Dynamics CRM.

Durante a fase em que desempenhei estas tarefas, tive que desenvolver diversas customizações para o MS Dynamics CRM, como Plugins, a serem executados em pré e post events, Workflows, Custom Workflows e Webresources, tais como páginas .html, scripts javascript, e ficheiros de estilos, .css. Todos estes desenvolvimentos foram feitos fora da aplicação MS Dynamics CRM, e posteriormente inseridas na solução.

Alguns exemplos de customizações efetuadas são: Plugin de Post-Create de um ponto de agenda da reunião, em que o mesmo apresente toda a lógica necessária para garantir que o ponto de agenda criado tenha um nome único dentro de todos os pontos de agenda até então existentes nessa reunião, caso contrário impossibilita a criação do ponto de agenda fornecendo um erro ao utilizador para que este entenda o que tem a fazer.



```

{
    tracingService.Trace("1");
    // Obtain the target entity from the input parameters.
    Entity entity = (Entity)context.InputParameters["Target"];
    #endregion
    try
    {
        Utils.clsBusinessBase uis = new Utils.clsBusinessBase(cModule, service, tracingService);

        tracingService.Trace("2");
        var assembly = Assembly.GetExecutingAssembly();

        string Ano_Max = uis.Utils.GetEmbeddedText("FCG_SIGA_Plugins.FetchXml.circular_AnoMax.xml");
        string Ano_Numb = uis.Utils.GetEmbeddedText("FCG_SIGA_Plugins.FetchXml.circular_Ano_Numb.xml");

        if (!entity.Attributes.Contains("fcg_year"))
        {
            entity.Attributes.Add("fcg_year", entity.GetAttributeValue<DateTime>("fcg_date").Year.ToString());
        }
        tracingService.Trace("3");
        //se não faltar nenhum
        #region Se não faltar nenhum
        if (entity.Attributes.Contains("fcg_circularnumber"))
        {
            tracingService.Trace("não falta nenhum");
            string fetch;

            //Número fornecido é Int, assim posso ir Get MAX Number
            fetch = Ano_Numb.Replace("@year", (string)entity.Attributes["fcg_year"]).Replace("@circularnumber", entity.Attributes["fcg_circularnumber"].ToString());

            EntityCollection mEntCol = service.RetrieveMultiple(new FetchExpression(fetch));

            if (mEntCol.Entities.Count > 0)
            {
                throw new InvalidPluginExecutionException("A combinação Ano/Número de circular já existe, por favor escolha uma nova combinação");
            }
        }
    }
}

```

Figura 19 - Código de *Plugin* de controlo de Nome de um Ponto de Agenda

Webresouce .html que tem como objetivo fornecer a possibilidade do utilizador ordenar por importância qual a ordem de todos os pontos de agenda de uma determinada reunião. Este .html possibilita ao utilizador escolher de forma autónoma essa ordem, ou se preferir, fazer uma ordenação automática, onde os pontos de agenda sejam ordenados conforme parâmetros previamente configurados nos mesmos.

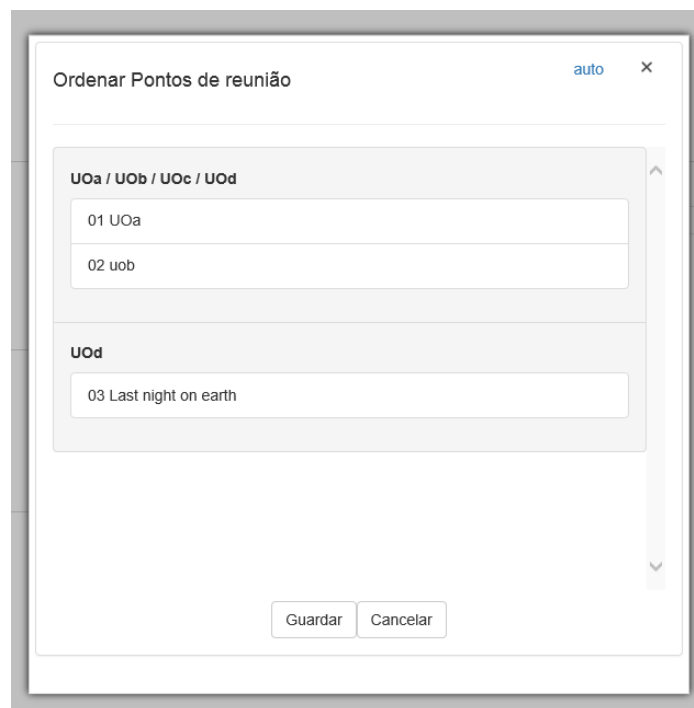


Figura 20 - Exemplo de ordenação dos Pontos de Agenda

- **Manutenção Corretiva**

Após o *deployment* e passagem a produção da solução, por vezes são detetadas falhas no sistema, falhas essas que correspondem a um requisito não estar a ser respondido de forma correta. Quando estes casos acontecem é necessário intervir-se na solução e corrigir o

problema de forma a garantir que a mesma responda aos requisitos acordados. Nesta perspetiva também executei tarefas de Manutenção Corretiva, pois apesar de ter sido colocado em outro projeto após a passagem a produção deste, quando surgiu algum problema neste âmbito que eu fosse capaz de o resolver, fui chamado para executar as modificações necessárias.

- **Suporte**

Já com a solução em ambiente de produção, por vezes é necessário dar suporte aos utilizadores. A meu cargo tive tarefas de suporte de segunda e terceira linha. No caso de suporte de segunda linha, fui chamado a intervir quando o problema era resolvido por falta de conhecimento da aplicação ou por uma incorreta utilização do mesmo. Já quando fui destacado um problema de desenho da solução, por vezes foi necessário fazer alterações ao código para que a solução executasse a sua tarefa tal como tinha sido acordada no ERS. Foi nestes casos que passei a exercer tarefas de suporte de terceira linha, em que tive de intervir diretamente no código da solução, fazendo as alterações necessárias para que a solução passasse a responder aos requisitos à qual foi desenhada.

Enquanto Consultor Júnior, para que conseguisse executar as tarefas a cima descritas, necessitei de aplicar determinadas *skills* representadas na *framework* SFIA. Apesar de em SFIA haver vários níveis para cada *skill*, nesta análise apenas serão referidas as *skills* utilizadas.

Category	Subcategory	Skill	Code	1	2	3	4	5	6	7
Solution development and implementation	Systems development	Systems design			2	3	4	5	6	
			DESN							
		Programming/software development	PROG		2	3	4	5		
		Testing	TEST	1	2	3	4	5	6	
	Installation and integration	Systems integration	SINT		2	3	4	5	6	
		Systems installation/decommissioning	HSIN	1	2	3	4	5		

Figura 21 - *Skills* utilizadas no Projeto xRM

## 7.2. PROJETO – IMPLEMENTAÇÃO DE SOLUÇÃO CRM

Após cinco meses de atividades na UNISYS, fui alocado a um novo desafio. Desta vez tratou-se de um Cliente do setor bancário que tinha como necessidade uma evolução tecnológica transversal à sua organização. O nosso papel enquanto consultores UNISYS foi desenvolver uma solução CRM que possibilitasse uma maior visão e controlo sobre os Clientes desta organização. Foi então proposta uma solução de Visão 360º sobre o cliente, que consistiu em fornecer uma visão global de toda a informação do mesmo, de forma rápida e concreta à distância de poucos cliques.

Devido à experiência prévia da UNISYS em implementações semelhantes no mesmo setor, era esperado um projeto calmo e controlado, o que não se veio a verificar ao longo do mesmo. Este projeto revelou-se um grande desafio para todos, pois o seu nível de complexidade do mesmo foi aumentando diariamente, bem como as exigências impostas pelo Cliente. Foi um projeto de elevada complexidade de implementação, bem como de muito *stress* e dificuldade de relação com um Cliente extremamente exigente e com pouco conhecimento sobre a aplicação em si.

A gestão de expectativas, também se revelou complicada devido ao facto do Cliente querer passar de não ter nada a ter tudo na mesma aplicação. A gestão da mudança foi ainda outra tema que levantou alguns problemas, pois os utilizadores finais estavam habituados à aplicação AS400:

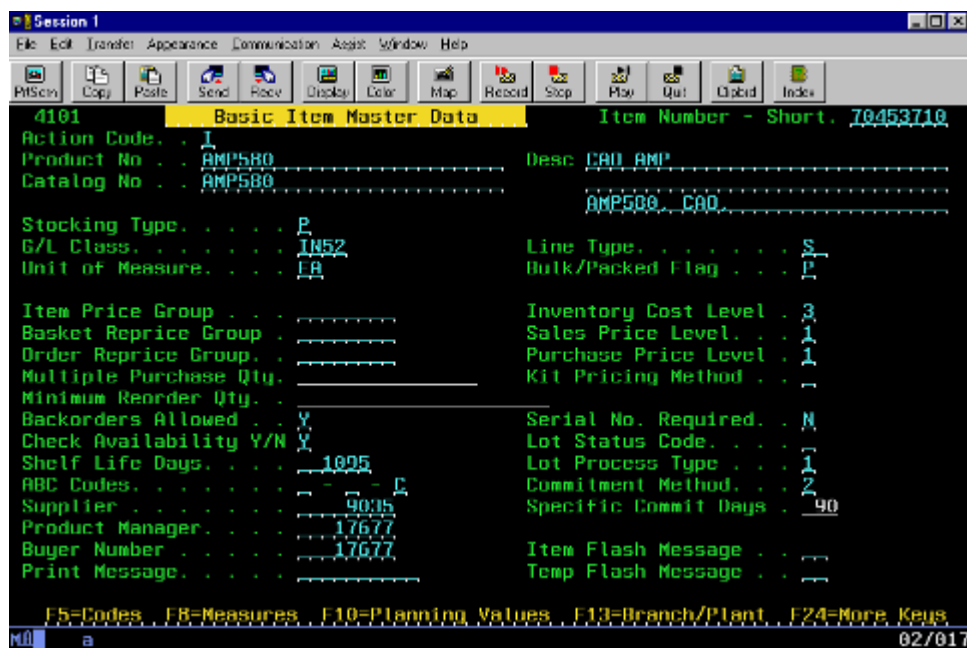


Figura 22 - Exemplo de um sistema AS400

Esta transição entre as duas aplicações tão distintas (AS400 para MS Dynamics CRM) foi complicada de gerir, não só pelo utilizador estar habituada durante anos de mexer com a antiga aplicação, mas também pelo aspeto gráfico e estrutura de dados do mesmo.

Esta solução MS Dynamics CRM teve como base de tornar-se o *front-end* de toda a informação de um Cliente, desde a sua informação básica (Nome, moradas, contactos etc), à sua informação mais específica como qual o seu gestor de conta; em que ações de campanha se encontra presente; quais

as atividades a que foi alvo; contratos de produtos subscrito; os saldos de contas; número de movimentos; a que segmento pertence, até qual situação atual do cliente com o banco entre outras informações disponíveis.

Foi ainda requisitado que houvesse uma sincronização entre dados provenientes do AS400 com os dados do MS Dynamics CRM e vice-versa, o que levou a uma arquitetura complexa que envolveu a utilização de Bases de Dados de *staging* com jobs agendados que serviram de ponte entre os dois sistemas.

Sendo o responsável do projeto por parte do cliente o departamento de Marketing, este quis funcionalidades que os ajudassem no seu dia-a-dia e tomarem decisões e a identificar comportamentos gerais de clientes. Foi portanto requisitado a implementação de um conjunto de diversos *dashboards* com informação sobre o cliente bem como que na aplicação fosse também possível fazer processos analíticos como gestão da carteira de clientes de cada gestor de conta, execução de análises de segmentação e análise comportamental com objetivo de implementar um sistema de *best next offer* sobre os clientes de forma individual.

### **7.2.1. DESAFIO**

Já com cinco meses de experiência profissional, respetivamente em MS Dynamics CRM, este era o meu primeiro projeto “a sério”. Senti que seria neste projeto que poderia de alguma forma mostrar o meu verdadeiro valor enquanto trabalhador e consultor.

Quando fui integrado na equipa, este era apenas formada por um elemento, elemento esse que era o Manager de CRM da UNISYS. O impacto foi enorme e senti uma pressão elevada para conseguir ajudar o único elemento da equipa até então.

As primeiras semanas foram de contextualização do projeto, e reuniões consecutivas com o cliente para definir o trabalho a desenvolver. Tive portanto como desafio o acompanhamento de toda a fase inicial de um projeto, o entendimento e análise de requisitos de um projeto que tinha acabado de começar.

Ao longo do tempo o maior desafio enfrentado não foi tecnológico mas sim o próprio cliente. Foi um projeto complicado em que a relação com o mesmo não foi a mais fácil nem a mais colaborativa. Tratou-se de um projeto muito desgastante com vários avanços e recuos, em que por vezes se tornava frustrante devido ao facto de vermos trabalho e horas de dedicação, deitados para o “lixo”.

Como o projeto se veio a revelar mais complicado do que inicialmente previsto, juntando ao facto de o cliente não ter grande conhecimento sobre a aplicação bem como as suas limitações, foi adotado para este projeto uma metodologia de desenvolvimento de *software* de Prototipagem

Ao longo do projeto foram sido feitos vários protótipos em que foram sendo apresentados ao cliente, para que este pudesse avaliar o que estava a ser desenvolvido. Estes protótipos foram alvo de críticas construtivas que levavam em cada iteração a uma abordagem evolutiva de forma a atingir os requisitos definidos e desejados.

Esta abordagem permitiu ao cliente ir tendo um contacto com a ferramenta desde os primeiros dias de início de desenvolvimento, o que ajudou a uma compreensão mútua entre a equipa do projeto (nós) e o cliente, pois para o cliente foi mais fácil visualizar e mexer na aplicação e entender o que era ou não possível fazer, e para nós enquanto consultores, tivemos a possibilidade de analisar a experiência e verificar as necessidades demonstradas pelo cliente.

## 7.2.2. TAREFAS E SKILLS

Como referido anteriormente, já com cinco meses de experiência profissional, as minhas tarefas enquanto consultor júnior, foram um pouco diferentes relativamente ao primeiro projeto neste relatório descritas. Para este projeto tive a cargo tarefas como: Levantamento e Análise de Requisitos, Planeamento do Projeto, Desenho da Arquitetura da Solução, Desenvolvimento, Testes e Formação.

- **Análise de Requisitos**

Devido à minha participação desde os primeiros dias do projeto, foi-me dada a possibilidade de fazer parte dos responsáveis dos levantamentos de requisitos impostos pelo cliente e presentes no Caderno de Encargos (CE).

Esta tarefa foi bastante complicada pois foi necessário um entendimento concreto do setor de atuação do cliente, bem como o entendimento das verdadeiras necessidades do mesmo. Devido à metodologia utilizada para o desenvolvimento deste projeto, foi uma tarefa que fui tendo e executando durante todo o seu ciclo de vida, pois com uma metodologia de prototipagem, foi necessário fazer um levantamento constante e correto das vontades e necessidades do cliente, de maneira a que o próximo protótipo respondesse aos requisitos impostos.

- **Planeamento do Projeto**

Tendo já alguma experiência sobre a matéria de desenvolvimento de uma solução CRM, tive a oportunidade de participar no planeamento do projeto. Este planeamento foi desde logo pensado com base na metodologia a ser utilizada para o desenvolvimento da solução, onde foram dados dias de *buffer* entre entregas de protótipos de forma a termos margens de manobra para eventuais alterações a serem feitas no protótipo entregue. O planeamento executado desta forma veio a revelar-se eficaz e extremamente preciso.

- **Desenho da Arquitetura da Solução**

Desde o início do desenvolvimento da solução que tive um papel presente e ativo na função de Desenho e Arquitetura da aplicação. A minha tarefa em conjunto com o meu Gestor de Projetos foi encontrar a melhor maneira de chegar à resolução de um determinado problema. Para que tal tivesse sido possível, foi feita uma análise do requisito ao pormenor, de forma a

entender quais os requisitos subjacentes, e após essa análise, era feito o desenho da solução, isto é, a arquitetura da própria solução.

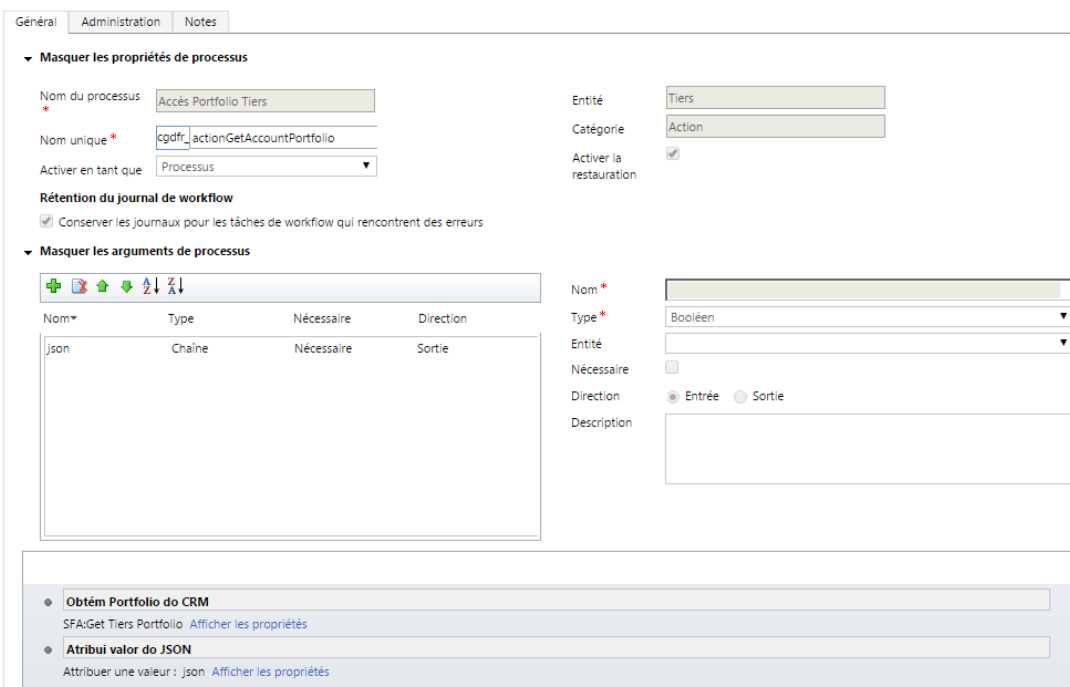


Figura 23 - Fotografia da sala de trabalhos com desenhos da solução

Tal como na análise de requisitos, esta foi uma tarefa em que também tive sempre presente. Para cada protótipo desenvolvido, foi feita toda uma análise e desenho do que viria a ser construído, de forma a garantir um correto funcionamento, integração com o material já desenvolvido e dinamismo para futuras alterações ou adições de funcionalidades.

## • Desenvolvimento

O meu papel em termos de desenvolvimento foi bastante idêntico ao já desempenhado no projeto anteriormente referido. No entanto, para este projeto foram-me colocados desafios de maior grau de exigência e mais complexos. Para este projeto em termos de customizações *client-side* tive a meu cargo o desenvolvimento de diversos *RealTime Workflows* e *Custom Actions*, em que um exemplo de *Custom Action* passou por executar toda a lógica necessária para obtenção dos contratos de um determinado cliente em tempo real:



Général Administration Notes

▼ Masquer les propriétés de processus

Nom du processus  Entité   
Nom unique\*  Catégorie   
Activer en tant que  Activer la restauration

Rétention du journal de workflow  
 Conserver les journaux pour les tâches de workflow qui rencontrent des erreurs

▼ Masquer les arguments de processus

Nom	Type	Nécessaire	Direction
json	Chaîne	Nécessaire	Sortie

Nom\*   
Type\*   
Entité   
Nécessaire   
Direction  Entrée  Sortie  
Description

● **Obtém Portfolio do CRM**  
SFA:Get Tiers Portfolio [Afficher les propriétés](#)

● **Atribui valor do JSON**  
Attribuer une valeur : json [Afficher les propriétés](#)

Figura 24 - Custom Action de obtenção do Portfólio do Cliente

Após a execução da *action*, foi então necessário fazer uma representação de forma *userfriendly* ao utilizador desses mesmos contratos. Para tal tive de desenvolver uma página .html que agrupasse de forma dinâmica e interativa todos os contratos desse cliente:

Figura 25 - Exemplo real da representação do Portfólio do Cliente

Tive ainda a cargo o desenvolvimento de outros componentes, tais como interfaces de conexão a *WebServices* fornecidos pelo cliente, de forma a consumir informação disponibilizada pelo mesmo e a apresentá-la ao utilizador na aplicação de CRM

Code	Description	Grading
DATBIL	Pas de Bilan Récent	R

Figura 26 - Representação da chamada feita em tempo real para obtenção do *grading* de um Cliente

- **Testes**

Devido ao meu envolvimento e participação desde o primeiro dia no projeto, tive como tarefa a elaboração e execução de testes finais sobre a solução. Os testes foram elaborados com base nos requisitos aprovados no ERS do projeto, enquanto que a sua realização visou apenas em validar se esses requisitos estavam a ser cumpridos ou não.

- **Formação**

Com a chegada ao fim do projeto, é iniciada a fase da formação de determinados utilizadores, como os responsáveis do projeto e responsáveis da formação interna. Nesta tarefa tive como papel principal a passagem de conhecimento a estes utilizadores de como utilizar a ferramenta e o que a mesma era capaz de fazer. Foi um desafio extremamente interessante pois após meses a desenvolver uma aplicação, esta parece simples e intuitiva, o que por vezes não é verdade, tendo sido nesta fase da formação identificadas e corrigidas algumas falhas de usabilidade.

Com o aumento de responsabilidades, para este projeto tive de utilizar mais *skills* do que no anterior. Tal como na análise do Projeto de xRM, nesta também será feita apenas sobre as *skills*. É notório um acréscimo de *skills* de desenvolvimento de sistemas, fatores humanos e instalação e integração:

Category	Subcategory	Skill	Code	1	2	3	4	5	6	7
Solution development and implementation	Systems development	Systems design	DESN		2	3	4	5	6	
		Programming/software development	PROG		2	3	4	5		
		Sustainability engineering	SUEN				4	5	6	
		Testing	TEST	1	2	3	4	5	6	
	Human factors	User experience analysis	UNAN			3	4	5		
		User experience evaluation	USEV		2	3	4	5		
	Installation and integration	Systems integration	SINT		2	3	4	5	6	
		Porting/software integration	PORT			3	4	5	6	
		Systems installation/decommissioning	HSIN	1	2	3	4	5		

Figura 27 - *Skills* utilizadas no Projeto CRM



## 8. CONCLUSÕES

Após a conclusão do estágio, é possível fazer um levantamento do trabalho desenvolvido e capacidades adquiridas. Neste capítulo será feita uma análise do documento produzido, tendo em conta o contexto em que foi elaborado, quais as perspectivas futuras, e apreciação global do estágio.

### 8.1. APRECIÇÃO CRÍTICA DO TRABALHO DESENVOLVIDO

Tendo em conta todas as variáveis em torno do desenvolvimento deste trabalho/relatório de estágio, devo considerar que se trata de um trabalho com extremo valor e de qualidade acima da média.

Trata-se de um relatório de estágio que para além de descrever o que foi aprendido e desenvolvido ao longo de um ano de trabalho, também tem com foco a ligação entre o que foi aprendido ao longo de quatro anos de faculdade (3 de licenciatura e 1 de mestrado) com a realidade e experiência profissional adquirida durante o estágio.

A realização deste relatório de estágio foi uma mais-valia para o meu desenvolvimento pessoal, isto porque me obrigou a ultrapassar desafios diários a nível de organização e responsabilidade, que são certamente *skills* de carácter extremamente importante para um sucesso profissional e pessoal. Este relatório de estágio, foi ainda importante para no fim ser-me possível fazer uma análise mais cuidada do que fiz e da minha evolução em termos profissionais ao longo deste último ano, o que me fornece uma ferramenta para analisar e refletir sobre todo o processo pelo qual passei e fui alvo.

Analisando o contexto em que o mesmo foi desenvolvido, no qual é importante salientar o facto de trabalhar e estudar ao mesmo tempo, é de referir que fui sempre capaz de responder às necessidades da empresa bem como aos objetivos traçados pela Universidade, o que me leva a crer e a concluir que o trabalho desenvolvido é de facto um trabalho que mostra ser possível, com muito esforço, dedicação e disciplina, conciliar vida profissional e académica, atingindo um nível excelência e qualidade acima da média.

### 8.2. PRESPECTIVAS FUTURAS

Apesar dos meus conhecimentos há um ano sobre matéria área de CRM serem escassos, depois de concluído este estágio enquanto profissional e consultor, acredito ter adquirido *know-how* suficiente para continuar a poder fazer parte de projetos de grande dimensão como os que tive possibilidade de participar.

Na minha análise o CRM é uma área de grande potencial de crescimento e com uma procura cada vez maior, o que me leva a crer ser uma das áreas a que será dada maior importância nas próximas décadas.

Será do meu agrado continuar nesta área, tendo como objetivo ter uma evolução de carreira estável e sustentada, dando o salto para a Gestão de Projetos dentro de um a dois anos, pois, apesar de gostar da área em que me encontro atualmente, tenho também um gosto muito acentuado pelas áreas de Gestão dos Sistemas de Informação, no entanto tenho noção que para chegar a um patamar de gestão, é primeiro necessário conhecer o que se vai gerir, e para isso, nada melhor do que ter as experiências que tenho tido.

### **8.3. APRECIÇÃO GLOBAL DO ESTÁGIO**

Se fosse necessário fazer uma avaliação do estágio, de uma escala de fraco a excelente, teria atribuído a classificação de muito bom.

Desde o primeiro dia que iniciei as minhas funções enquanto Consultor Júnior na UNISYS, que tenho sido acompanhado pelos melhores especialistas na área de consultoria de MS Dynamics CRM de Portugal. Desde os meus colegas diretos ao meu Manager de CRM, passando pelo meu Gestor de Projetos que me acompanhou desde o primeiro dia até aos dias que correm, ajudando-me sempre a entender o contexto em que me encontro, não só a nível do projeto como a nível organizacional, a explicar novos conceitos e tecnologias, e acima de tudo na transmissão de um espírito de brio e qualidade no trabalho desenvolvido, de forma a garantir que a solução final não só atinja as expectativas do clientes, como seja, também, de elevada qualidade e se possível com toques de criatividade e inovação, tentando sempre exceder o expectado pelo cliente.

Ao longo do estágio fui tendo diversos desafios, como a necessidade de aprendizagem de diversas matérias num curto espaço de tempo, colocada à prova a minha capacidade de resolução de problemas, relação com o cliente, entre outras. No entanto, estaria a mentir se não referisse que o maior desafio e mais complicado foi a gestão do equilíbrio entre vida Profissional, Académica e Pessoal.

No princípio do estágio, e ainda com toda a força para iniciar a minha vida profissional, pareceu-me que viria a ser fácil de manter um ritmo elevado de forma a manter esse mesmo equilíbrio entre as três vertentes, no entanto, mês após mês, noitada após noitada, o desgaste surgiu e começou a fazer-se sentir, não só desgaste físico, mas também desgaste psicológico.

Foi por vezes complicado manter a concentração em todas as vertentes, levando-me a pensar sobre desistir do estágio ou do curso, pois por vezes pensava que já não ia ser capaz de fazer ambas. No entanto, com ajuda e compreensão por parte da UNISYS, foi-me possível conciliar todas as vertentes de forma a encontrar um equilíbrio saudável e sustentável, fornecendo-me assim as condições necessárias para a finalização e conclusão do Mestrado.

O facto de trabalhar e estudar simultaneamente é algo muito exigente. No entanto, considero uma experiência enriquecedora que me permitiu obter ensinamentos valiosos para o resto da vida, obrigando-me a ter métodos de gestão do meu tempo muito rigorosos e até descobrir capacidades que até então as desconhecia.

Acredito que foi um ano extremamente importante para mim enquanto pessoa e profissional, que me proporcionou amadurecimento e me conferiu maior capacidade para superar limites e me conhecer melhor. Tive ainda a possibilidade de participar em projetos extremamente aliciantes que me deram conhecimentos valiosos para prosseguir a minha carreira profissional.

## BIBLIOGRAFIA

EUROSTAT (2007-2010). Enterprises using software solutions like CRM to analyse information about clients for marketing purposes

Frow, P., & Payne, A. (2009). Customer relationship management: a strategic perspective.

<http://dialogoti.intel.com/pt-br/documento/melhorando-o-rendimento-das-pessoas-em-sua-gestao>

<http://www.agilemanifesto.org/iso/ptpt/>

[http://www.xrm.com/xrm/xrm\\_defined.aspx](http://www.xrm.com/xrm/xrm_defined.aspx)

INE (2007). Lista das 1000 maiores empresas portuguesas. Ficheiro de Unidades Estatísticas - FUE - Base Belém: Instituto Nacional de Estatística.

Light, B. (2003). CRM packaged software: a study of organisational experiences.

Microsoft Brasil. Definição de nomenclatura Gold Partner

Nguyen, B., & Mutum, D. S. (2012). A review of customer relationship management: successes, advances, pitfalls and futures.

Osarenkhoe, A., & Bennani, A.-E. (2007). An exploratory study of implementation of customer relationship management strategy.

Payne, A., & Frow, P. (2006). Customer relationship management: from strategy to implementation.

SFIA Framework Reference, Skills defined in categories and subcategories. SFIA Foundation 2011

Valente T. (2010). Marketing de Relacionamento e CRM uma análise da gestão de clientes no setor financeiro

Wayland & Cole (2010). Customer Connections: New Strategies for Growth

Category	Subcategory	Skill	Code	1	2	3	4	5	6	7
Solution development and implementation	Systems development	Systems development management	DLMG					5	6	7
			DTAN	2	2	3	4	5		
			DESN	2	2	3	4	5	6	
			NTDS					5	6	
			DBDS	2	2	3	4	5	6	
			PROG	2	2	3	4	5		
			ADEV			3	4	5	6	
			SFEN			3	4	5	6	
			SUEN				4	5	6	
			INCA	1	2	3	4	5	6	
			TEST	1	2	3	4	5	6	
			UNAN			3	4	5		
			HCEV			3	4	5	6	
			USEV	2	2	3	4	5		
			HFIN					5	6	7
			SINT	2	2	3	4	5	6	
			PORT			3	4	5	6	
			HSIN	1	2	3	4	5		

## ANEXOS