



Instituto Superior de  
Tecnologias Avançadas  
Licenciatura em Informática

PROJETO GLOBAL  
HelpDesk - ITNext

Aluno:	Tiago André da Cunha Diogo Nº 1615 - Turma A - Lisboa Ano Letivo 2014/15
Coordenador:	Prof. Pedro Brandão
Orientador:	Prof. Rui Carriço



# Dedicatória

---

Dedico este trabalho à minha família e aos meus amigos que estão constantemente presentes em tudo o que faço. Obrigado pelo apoio e dedicação demonstradas, principalmente numa altura em que estive mais em baixo.

Dedico particularmente este trabalho à minha mãe e à minha irmã, que sempre fizeram questão de me lembrar que nunca devemos desistir e que devemos sempre acreditar que somos capazes. Agradeço-lhes, ainda, o apoio incondicional e constante.

Ao meu Pai, pois embora tenha partido muito cedo desta vida, está sempre presente em tudo o que digo e faço. Por todos os ensinamentos que me passaste, todos os bons e maus momentos, muito obrigado por tudo, Pai.

Por fim, ao meu grande amigo Henrique Graça, por todo o apoio e ajuda que me deu para a conclusão deste projeto. Sem ele, nada disto seria possível.

# Resumo

---

Neste trabalho, realizado no âmbito do trabalho final do curso de informática do ISTECS - Instituto Superior de Tecnologias Avançadas, pretende-se descrever o processo de conceção do projeto “*HelpDeskITNext*”.

O trabalho tem como objetivo fundamental o desenvolvimento de uma aplicação *Helpdesk – Servicedesk*, cuja função principal é a gestão de todas as avarias ocorridas numa determinada empresa. Resumidamente, o procedimento é iniciado por um utilizador, previamente registado e identificado na aplicação, que abre um registo de intervenção, no qual é gravada a descrição da avaria e a prioridade da mesma. Este passo leva à criação automática de um *ticket*<sup>1</sup> o qual é tratado pelo técnico da ferramenta, consoante a prioridade do mesmo. Após a resolução da avaria, o *ticket* é encerrado, sendo feita a respetiva comunicação do encerramento pelo técnico ao utilizador.

É importante notar que este trabalho, na medida em que assume uma vertente mais prática, não é apenas um projeto académico. O projeto em causa será implementado na empresa MCDiogo – Mediação de Seguros, o que permitirá facilitar o meu trabalho como informático residente e ainda o trabalho dos utilizadores. Embora a conceção do projeto se adeque às necessidades da empresa MCDiogo, o trabalho retém a sua natureza inicial académica, i.e., de projeto de final de curso.

**Palavras-chave:** *IT, HelpDesk*, aplicação, programa.

---

<sup>1</sup> Neste documento será usado o termo inglês *ticket* visto ser o termo mais usual de designação e a sua tradução poderia introduzir incoerências no texto e comprometer a sua compreensão. Ao longo deste trabalho serão utilizadas expressões de língua inglesa sempre que o seu uso seja mais recorrente do que os termos em língua portuguesa correspondentes.

# Abstract

---

This work is an integral part of my degree in Computer Engineering in ISTECS - Instituto Superior de Tecnologias Avançadas.

The main part of the work addresses a project entitled "*HelpDeskITNext*" process and design. It aims at developing an *Helpdesk – Servicedesk* application, which has the capability to conduct fault management in a company.

The fault management process is performed as follows:

The process is started by a user (previously registered and identified in the application). Users must open an intervention registration where the faults are described and recorded.

This step leads to the automatic creation of a ticket. A random technician handles the incident and solves the incidence depending on the priority of it. After solving the problem, the ticket is closed.

This project will be implemented in MCDiogo – Medicação de Seguros as a working tool and will be available to users. As such, this project suited MCDiogo's particular needs, while maintaining its academic nature.

**Palavras-chave:** *HelpDesk*, *Ticket*, aplicação.

# Lista de Abreviaturas

---

AJAX -AsynchronousJavaScriptand XML

ASP.NET - Active ServerPages

C# - C sharp

CLI - CommonLanguageInfrastructure

CLR - CommonLanguageRuntime

CSS - Cascading Style Sheets

DLL - Dynamic-link library

HTML - HyperText Markup Language

HTTP - HyperText Transfer Protocol

IIS - Internet Information Services

ITIL -InformationTechnologyInfrastructureLibrary

SGBD - Serviço de Gestão de Base de Dados

BD - Base de dados

SLA - Service Level Agreements

SPOC - Single Point Of Contact

SQL - StructuredQueryLanguage

TI - Tecnologia da Informação

XML - eXtensibleMarkupLanguage

MVC - Model View Controler

EF- Entity Framework

SMS - Short Message Service

# Índice

---

Dedicatória .....	I
Resumo .....	II
Abstract .....	III
Lista de Abreviaturas.....	IV
Índice .....	V
Índice de Figuras .....	VI
Enquadramento.....	1
Organização do Relatório.....	1
1. Introdução .....	2
1.1) Conceito de <i>HelpDesk</i> .....	3
1.2) <i>HelpDesk</i> ou <i>ServiceDesk</i> : Semântica ou Estratégia?.....	5
2. Estado da Arte.....	7
2.1) Características de um sistema de <i>HelpDesk</i> .....	7
2.2) Fundamentos de ITIL adotados .....	9
2.3) Soluções existentes .....	13
3. Metodologia .....	13
4. Implementação .....	14
4.1) Tecnologias utilizadas .....	15
5. Análise da Aplicação .....	27
5.1) Visão original do sistema.....	27
5.2) Requisitos não funcionais .....	27
5.3) Requisitos funcionais.....	28
6. Conclusão.....	38
Referências .....	39
Bibliografia .....	39

# Índice de Figuras

---

Figura 1 – Ciclo interminável de PDCA.....	13
Figura 2 – Ambiente de desenvolvimento web.....	17
Figura 3 – Processo de conversão do código fonte para nativo.....	18
Figura 4 – Arquitectura. NET <i>Framework</i> . .....	19
Figura 5 – Diagrama das diferentes tabelas e seus relacionamentos. ....	22
Figura 6 – Esquema da base de dados de utilizadores.....	23
Figura 7 – Diagrama daEntity Framework – cenário 1.....	24
Figura 8 –Diagrama daEntity Framework – cenário 2.....	25
Figura 9 – Diagrama daEntity Framework - cenário 3. ....	25
Figura 10 –Esquema de actividade do utilizador.....	29
Figura 11 – Esquema de acesso à aplicação. ....	29
Figura 12 – Ecrã de <i>login</i> da aplicação com a indicação para registo de utilizador .....	30
Figura 13 – Criação de nova conta de utilizador Tiago Diogo (a título de exemplo).....	31
Figura 14 – Ecrã com login efectuado com sucesso. ....	31
Figura 15 – Tabela criada com as prioridades e o grau de importância de cada estado .....	33
Figura 16 – Mensagem recebida por um dos técnicos assignados, com um <i>ticket</i> atribuído ...	33
Figura 17 – Ecrã de relatório de <i>ticket</i> com destaque para a edição do mesmo. ....	34
Figura 18 – Relatório do pedido e estado do <i>ticket</i> via <i>SMS</i> . ....	35
Figura 19 – Ciclo de vida de um <i>ticket</i> . ....	35
Figura 20 – Ecrã de gestão de <i>tickets</i> por um dos técnicos da ferramenta.....	36
Figura 21 – Ecrã de gestão de acessos do Administrador.....	37



# Enquadramento

---

O projeto acima identificado foi realizado no âmbito do projeto final de curso e tem como objetivo o desenvolvimento de uma aplicação de *ServiceDesk* que tenha a capacidade de observar os seguintes requisitos:

- ✓ Implementação de uma aplicação *web* que permita aos utilizadores reportar problemas, dúvidas e questões ao nível do *hardware* ou *software*;
- ✓ Facilitação e otimização da resolução de problemas técnicos nos vários departamentos da empresa MCDiogo através da comunicação das anomalias detetadas ao técnico associado; A aplicação deverá registar todas as anomalias e informar os utilizadores do estado de resolução dos seus pedidos de suporte técnico, através de *SMS*;
- ✓ Expansão e arquivo de conhecimento relativo à resolução de anomalias, sendo criando um repositório das ocorrências passadas e respetivas resoluções, sem prejuízo de eventuais alterações do pessoal que componha a equipa técnica.

# Organização do Relatório

---

O presente relatório encontra-se organizado nos 6 capítulos seguintes:

- 1) **Introdução:** Breve descrição geral do trabalho final de curso e dos conceitos fundamentais que serviram de base ao trabalho (*HelpDesk* e *ServiceDesk*), objetivos e enquadramento do trabalho desenvolvido;
- 2) **Estado da Arte:** Apresentação das principais tecnologias usadas no âmbito do projeto;
- 3) **Metodologia:** Apresentação das metodologias de pesquisa e desenvolvimento usadas no projeto;
- 4) **Implementação:** Descrição detalhada da parte técnica do projeto;
- 5) **Análise da aplicação:** Descrição detalhada da parte funcional do projeto;
- 6) **Conclusão:** Considerações finais sobre a execução do projeto e reflexão sobre o cumprimento dos objetivos propostos.

# 1. Introdução

---

O crescente desenvolvimento das Tecnologias da Informação (“TI”) fez depender os negócios / as empresas cada vez mais da informática.

Devido a esta dependência, os departamentos de TI das empresas têm vindo a adotar metodologias para manter e melhorar a qualidade dos serviços prestados, sejam em regimes internos ou em regime de *outsourcing*, com o objetivo de responder ou até mesmo de desenvolver e melhorar a prestação dos serviços face às necessidades dos clientes, aumentando, assim, a satisfação e produtividade dos clientes<sup>2</sup>.

Quando um cliente ou utilizador se depara com algum problema, reclamação ou incidente, é necessário encontrar respostas e soluções, sendo a rapidez da solução um aspeto essencial para a satisfação do cliente e para que haja o mínimo impacto no negócio. É de extrema importância conseguir evitar o processo frustrante de ter que passar por diversas entidades até se conseguir resolver uma determinada situação.

De modo a poder responder às necessidades dos clientes, muitas organizações têm implementado um ponto central de contacto com os mesmos, com o objetivo de prestar um serviço mais rápido e eficaz. Esta função, conhecida como *ServiceDesk*, é a principal interface aplicacional dentro de uma empresa entre o departamento de TI e os utilizadores, servindo, sempre que exista necessidade de reportar uma determinada ocorrência, como primeiro contacto dos utilizadores com o de TI<sup>2</sup>.

O *ServiceDesk* é a evolução do *HelpDesk* uma vez que possui abrangência e qualidade superiores para atender a pedidos. O *ServiceDesk* permite centralizar as necessidades de uma empresa num único lugar, registando a entrada e saída de pedidos de suporte e manutenção, daí decorrendo um maior controlo do trabalho realizado.

---

<sup>2</sup>MAGALHÃES, I. L. e PINHEIRO, W. B., *Gerenciamento de Serviços de IT na prática: uma abordagem com base na ITIL*, São Paulo : Novatec, 2007.

A missão principal do *ServiceDesk* é o restabelecimento da operação normal dos serviços dos utilizadores o mais rapidamente possível, minimizando o impacto nos negócios por falhas de TI. O uso de ferramentas de gestão de serviços de TI bem estruturadas é também muito importante para a prestação de um bom serviço.

Por outro lado, esta aplicação pode ser considerada como uma biblioteca de resoluções de problemas, funcionando como arquivo de todos os problemas e questões levantados pelos utilizadores e das soluções previamente utilizadas.

## 1.1) Conceito de *HelpDesk*

A tecnologia invadiu completamente o nosso dia-a-dia, desde o cidadão comum até à pessoa mais bem conceituada. Acompanhar a velocidade da evolução pode ser um processo bastante complexo, sendo o tempo um fator crucial independentemente da área de negócio de cada empresa.

Se há uns anos atrás as funcionalidades das aplicações podiam ser descritas como básicas, hoje existem aplicações com capacidade para efetuar tarefas bastante complexas para as diferentes áreas.

De facto, o desenvolvimento de *software* é um dos setores tecnológicos com maior taxa de evolução. A computação no seu todo tem sofrido grandes avanços a nível tecnológico e continua em permanente alteração e evolução. Esta evolução contínua é causada pela necessidade de melhorar ou aumentar a eficiência das diferentes aplicações, podendo ser estimulada pelo aparecimento de novas necessidades relacionadas com alterações no meio empresarial, pela evolução de *hardware* mais sofisticado e com maiores potencialidades ou até mesmo pela resolução de problemas (ou *bugs*) existentes nas versões anteriores das aplicações que ponham em causa o seu uso adequado.

O *Helpdesk* centraliza a receção dos pedidos dos clientes, sejam eles para o esclarecimento de dúvidas, reclamações, requisição de serviços ou abertura de uma evidência maior. Esse apoio aos clientes pode funcionar dentro da própria empresa ou externamente em empresas terceiras prestadoras de serviços em regime de *outsourcing*.

Nas empresas, o *Helpdesk* opera através da concentração de vários canais de comunicação tais como, central telefónica, *e-mail*, mensagens, *intranet* ou qualquer outro meio de comunicação que receba as solicitações dos clientes. Estes pedidos são atendidos por profissionais qualificados, denominados de analistas ou consultores de TI.

As ligações dos clientes são registadas num sistema específico para o controlo de atendimento, sendo esta ação denominada de “abertura de *ticket*”. As solicitações mais complexas são encaminhadas para os respetivos departamentos responsáveis, onde são atendidas e registadas no sistema até a resolução final do problema.

Cada vez mais as empresas dependem do bom funcionamento tanto do seu *hardware* como do *software*, nem que se trate de uma aplicação de planeamento de recursos empresariais (ERP – *EnterpriseResourcePlanning*), i.e. uma aplicação que abrange todos os departamentos, dados e processos de uma organização desde a área financeira aos recursos humanos. Todavia, nenhuma tecnologia garante total fiabilidade podendo, por isso, ocorrer erros, designadamente ao nível da implementação, configuração ou modo de utilização por parte do utilizador.

Revela-se, assim, essencial a implementação de mecanismos de suporte técnico para prestação de apoio aos utilizadores (suporte interno) ou clientes (suporte externo). Em ambos os casos o objetivo é idêntico, o aperfeiçoamento do nível do serviço prestado e a manutenção da qualidade e eficiência dos serviços. Como resposta a esta necessidade surgiram os sistemas de *HelpDesk* que têm como finalidade o apoio ao utilizador e a resolução de problemas técnicos, constituindo um ponto central onde os problemas podem ser reportados e conseqüentemente geridos e resolvidos<sup>3</sup>.

---

<sup>3</sup>*Helpdesk World. The HelpDesk Software and Information Portal*, [Online], (<http://www.help-desk-world.com/>), consulta em 23 de Outubro de 2015.

## 1.2) *HelpDesk* ou *ServiceDesk*: Semântica ou Estratégia?

Poderá equacionar-se se as diferenças entre o *HelpDesk* e o *ServiceDesk* decorrem simplesmente da semântica e terminologia dos mesmos ou se têm verdadeiramente um impacto distinto na estratégia ou departamento de TI de uma organização.

Esta é, de facto, uma velha discussão para a qual existem respostas, interpretações e pontos de vista diferentes. As expectativas do utilizador final são muito maiores hoje em dia.

Uma simples pesquisa revela um número igual de pessoas que consideram e que não consideram existir uma diferença entre os dois conceitos.

Algumas das discussões e pontos de vista sobre este tema são baseados nas interpretações das organizações, usando alguma especificidade quanto às normas de ITIL (*InformationTechnologyInfrastructureLibrary*).<sup>4</sup>

Não existem comentários mais corretos do que os outros. Não existem regras rígidas e rápidas que obriguem ao uso de melhores práticas. Na verdade, tudo depende do que melhor funciona do ponto de vista organizacional.

Eis algumas definições padrão dos dois conceitos juntamente com alguns exemplos e diferenças entre cada um.

### 1.2.1) *HelpDesk*

É considerada uma ferramenta tática ou como resposta imediata para resolução rápida de questões técnicas ou incidentes reportados pelos clientes finais. Esta ferramenta pode ser integrada com o *ServiceDesk* para melhorar o atendimento ao cliente da organização. O

---

<sup>4</sup>Conceitos de *HelpDesk* e *ServiceDesk* [Online] (<http://www.arincmanagementservices.com/blog/2011/02/part-3-what-are-the-critical-differences-between-a-service-desk-and-a-help-desk/>), consulta a 4 de Junho de 2015.

objetivo final do *HelpDesk* é garantir a resolução de quaisquer incidentes no primeiro contacto e da forma mais eficiente e rápida possível.<sup>5</sup>

O *HelpDesk* tem como principais características:

- Ponto único de contacto (SPOC);
- Gestão de incidentes básicos e de *troubleshooting*;
- Integração limitada com outros processos de gestão de serviços de TI;
- Algumas áreas ou aplicações são suportados por grupos ou organizações fora do *HelpDesk*;
- Resolução de problemas;
- Base de conhecimento;
- Acordos de Nível de Serviço (SLA's);

### 1.2.2) *ServiceDesk*

O *ServiceDesk* é considerado principalmente ao nível da estratégica organizacional (por vezes, é inclusivamente referido como “foco organizacional”).

O *ServiceDesk* está mais direcionado para as necessidades do negócio e não tanto para as necessidades do utilizador. Tem em conta um contexto mais amplo de negócio.

A definição *ITIL* da Central de Serviço (*ServiceOperation*) é o único ponto de contacto entre o prestador de serviços e o utilizador. O *ServiceDesk* típico gere incidentes e solicitações de serviços, bem como a comunicação com os utilizadores.

O *ServiceDesk* normalmente tem uma componente de *HelpDesk* e tem como objetivo global o melhoramento das boas práticas das equipas e TI e dos processos de negócios em toda a organização.

O *ServiceDesk* tem como principais características:

- Integração total com outros processos de gestão de serviços;
- Gestão centralizada para processos / aplicações das TI;
- Gestão de configurações;

---

<sup>5</sup> Significados [Online], (<http://www.significados.com.br/service-desk/>), consulta a 1º de Junho de 2015.

- Gestão de incidentes:
- *Configuration management database* (CMBD) e de ativos de pesquisa e gestão.

## 2. Estado da Arte

---

Neste capítulo serão apresentadas as características mais comuns dos sistemas de *ServiceDesk* que, de algum modo, serviram de base para o desenvolvimento do projeto. Serão também apresentados alguns fundamentos das boas práticas de ITIL (*InformationTechnologyInfrastructureLibrary*)<sup>6</sup>, tais como:

- ✓ Centralização, num único ponto, todos os pedidos de assistência técnica por forma a facilitar a comunicação entre os utilizadores e o departamento técnico<sup>7</sup>.
- ✓ Criação de níveis de suporte que consistam na separação da equipa técnica por níveis, dependendo da sua área de formação e/ou do seu conhecimento técnico.

---

<sup>6</sup> MANSUR, R. O que é ITIL? *Profissionais de Tecnologia*. [Online] 1 de Maio de 2009. [Citação: 5 de Setembro de 2013.] <http://www.profissionaisdetecnologia.com.br/blog/?p=168>.

<sup>7</sup> UNIVERSITY OF MANITOBA. [Online] 22 de Agosto de 2013. [Citação: 2013 de Setembro de 3.] [http://umanitoba.ca/campus/physical\\_plant/media/Single\\_Point\\_of\\_Contact-Mar\\_4\\_2011.pdf](http://umanitoba.ca/campus/physical_plant/media/Single_Point_of_Contact-Mar_4_2011.pdf)



## 2.1) Características de um sistema de *HelpDesk*

Os sistemas de *Helpdesk / ServiceDesk* têm como base a utilização de tecnologia web e apresentam uma interface apelativa ao utilizador. Têm também a grande vantagem de não necessitarem de instalação de *software* nos postos de trabalho dos utilizadores sendo as atualizações da total responsabilidade do fornecedor do software<sup>8</sup>.

De uma maneira geral, as soluções existentes, bem como o projeto *HelpDeskITNext* possuem as seguintes características<sup>9</sup>:

### ✓ **Administração e consulta de incidentes**

Com esta ferramenta, os diferentes utilizadores do sistema podem registar os incidentes ocorridos, utilizando a interface *web* proporcionada ou através do envio de uma mensagem de *e-mail* para algum dos endereços que sejam disponibilizados. Após serem inseridos, os pedidos são categorizados e encaminhados para os respetivos técnicos responsáveis pela resolução do problema.

### ✓ **Informação sobre o histórico de pedidos**

Com esta solução é possível observar os pedidos ocorridos numa empresa ou instituição e administrados pelo departamento de TI num determinado período de tempo. É possível conhecer as informações estatísticas relevantes sobre as necessidades de seus utilizadores. Esta solução permite, assim, a facilitação dos procedimentos e uma consequente melhoria contínua do serviço.

### ✓ **Facilidade de acesso à ferramenta**

Qualquer um dos *softwares* apresentados possui uma interface simples e moderna, disponível por via *online*, a qual permite que os utilizadores inseriram os seus pedidos e os acompanhem por esta via. Esta característica elimina a necessidade de instalações e de distribuição de pacotes.

---

<sup>8</sup> QUALITOR. [Online] [Citação: 4 de Setembro de 2013.]  
<http://www.qualitor.com.br/site/content/caracteristicas/>.

<sup>9</sup> ITNOVA. Service Desk, IT Management Software. [Online] [Citação: 1 de Setembro de 2013.]  
<http://www.itnova.com.br/pdf/Itnova-Service-Desk.pdf>



### ✓ **Centralização e organização das informações**

É possível administrar os procedimentos de suporte de várias organizações ou projetos a partir de um único ponto centralizado. Além disso, é possível a visualização de toda a informação relacionada a partir do mesmo ponto de acesso.

### ✓ **Notificação sobre modificações nos incidentes**

Com esta ferramenta, tanto os utilizadores como os técnicos envolvidos na resolução do incidente podem receber notificações através de *e-mail* ou de mensagens enviadas diretamente ao equipamento tendo, por isso, acesso constante à informação atualizada sobre as mudanças no estado do incidente.

## 2.2) Fundamentos de ITIL adotados

Um erro que vulgarmente se verifica em diversos meios de divulgação é a classificação de ITIL como uma metodologia. É necessário compreender que a ITIL é um conjunto das melhores práticas para a gestão de serviços de TI de uma empresa ou entidade, podendo ser aplicada de diversas formas, dependendo das necessidades específicas de cada empresa ou negócio<sup>10</sup>. Por outro lado, uma metodologia define regras e padrões que devem ser seguidos e executados exatamente como previsto, oferecendo, por isso, muito menos flexibilidade. A classificação da ITIL através dos termos “biblioteca de boas práticas”, “biblioteca” ou “*framework*” considera-se muito mais apropriada.

### ✓ **Ponto Único de Contacto (SPOC)**

A aplicação representa um ponto focal de informações, capaz de acompanhar todo o atendimento do utilizador desde o início (i.e. desde a altura em que o *ticket* é criado) até à sua resolução. Uma das grandes vantagens da aplicação é o facto de dispensar o contacto direto com outros departamentos, eliminando a ambiguidade da procura da pessoa ou departamento correto para resolver o problema, permitindo, por isso, uma muito menor perda de tempo. Outra das grandes vantagens da aplicação é a redução de custos. O facto de a aplicação dispor

---

<sup>10</sup>MENDONÇA, F. H. de. Benefícios da aplicação da ITIL em empresas – com foco na Gestão de Problemas. [Online], (<http://www.fatecsp.br/dti/tcc/tcc0032.pdf>), consulta a 9 de Outubro de 2015.

de uma plataforma exclusivamente dedicada à resolução dos problemas que sejam enviados implica que outros setores não diretamente relacionados com o problema não sofram perdas de tempo a redirecionar o problema para o setor certo. Sendo assim, o trabalho planeado e previsto pode continuar sem inconvenientes. Por último, como todos os *tickets* passam pelo mesmo ponto, torna-se mais simples verificar o estado de cada um e, em particular, se estão a ser resolvidos dentro da janela temporal esperada, podendo proceder-se a uma escalção em caso negativo. Isto torna o sistema simples e eficaz.<sup>11</sup>

### ✓ Níveis de Suporte

Outro conceito bastante utilizado e proposto pela ITIL é o da criação de níveis de suporte. Este conceito consiste na divisão da equipa de atendimento em grupos ou níveis diferentes, definidos pelo grau de conhecimento ou competência técnica do grupo. Nesta divisão, os profissionais do 1º nível de suporte detêm o mínimo de conhecimento técnico e podem servir de interface entre os clientes ou utilizadores (internos ou externos) e o 2º nível de suporte deterá maiores conhecimentos técnicos e terá a capacidade de resolver problemas mais complicados que estejam para lá das limitações do 1º nível<sup>18</sup>.

No caso concreto da aplicação *HelpDeskITNext* são implementados apenas dois níveis de atendimento. Esta limitação foi aplicada devido à natureza do projeto e às necessidades específicas da empresa que conta com um número reduzido de elementos técnicos no seu departamento de TI.

A separação por níveis tem como objetivo a otimização do investimento na área de *HelpDesk*. O cenário ideal passa pela concentração de um maior número de elementos iniciantes do 1º nível e de um menor número de funcionários dos níveis superiores. Assim, os elementos do 1.º nível podem filtrar os problemas, resolvendo rapidamente os de menor complexidade ou cuja solução se encontre na base de conhecimento, enquanto os funcionários de níveis superiores recebem muito menos casos para resolver, podendo assim concentrar-se nos incidentes mais complexos, passando-os para os níveis seguintes caso não consigam encontrar a solução.

---

<sup>115</sup> Idéias da ITIL que Podem Ajudar sua Área de *HelpDesk*. *Marttrend*. [Online], (<http://www.marttrend.com.br/?p=84>), consulta a 3 de Outubro de 2015.

Esta filtragem, combinada com a separação por níveis, permite um atendimento rápido de incidentes de qualquer grau de complexidade.

#### ✓ Acordos de Nível de Serviço (SLA)

O estabelecimento de um SLA entre a área de *HelpDesk* e os clientes (internos ou externos) permite diminuir o fluxo de ligações efetuadas para o serviço de atendimento para obtenção de informações sobre o estado de resolução de um problema uma vez que o SLA garante a disponibilidade dos serviços de atendimento e o tempo de atendimento da central de suporte<sup>18</sup>. A título exemplificativo, quando se estabelece que as ligações por parte dos postos de trabalho serão atendidas no tempo máximo de 30 minutos e resolvidas em 8 horas no máximo, fica garantida uma escala de tempo expeável para a resolução e contacto por parte do *HelpDesk*, deixando, assim, de haver lugar a novos contactos por parte do cliente antes do fim do prazo definido<sup>12</sup>.

O SLA é adaptado neste projeto ao nível da prioridade, sendo associado a diferentes tempos de resolução. São implementados quatro níveis de prioridade: “crítica, elevada, média e baixa” os quais definirão em quanto tempo e de que forma o serviço será prestado. Estes níveis de prioridade serão descritos em maior detalhe mais adiante.

#### ✓ Base de conhecimento

A base de conhecimento, conceito também sugerido pela ITIL, é uma ferramenta que, quando utilizada convenientemente, permite uma ainda maior redução do tempo de atendimento de um *HelpDesk*. Esta base, acessível pelos técnicos através de buscas, consiste no registo de um conjunto de informações centralizadas referentes a erros reportados e respetivas soluções. Através do acesso à base é possível encontrar rapidamente soluções até então desconhecidas. Sempre que problemas de maior complexidade sejam resolvidos, a sua resolução deve ser documentada e adicionada à base de conhecimento para que possa ser partilhada por toda a equipa de *HelpDesk*. Aquando da reincidência do mesmo problema

---

<sup>12</sup> MENDONÇA, F. H. de. Benefícios da aplicação da ITIL em empresas – com foco na Gestão de Problemas. [Online], (<http://www.fatecsp.br/dti/tcc/tcc0032.pdf>), consulta a 9 de Outubro de 2015.

complexo, a sua resolução poderá ser muito mais rápida uma vez que uma simples busca na base permitirá encontrar a solução adequada. O acesso à base pode também ser oferecido aos próprios utilizadores de modo a que os mesmos possam encontrar soluções para problemas conhecidos, diminuindo assim o volume de chamadas dirigidas ao *HelpDesk*.

Resumindo, algumas das vantagens de uma base de conhecimento são as seguintes:

- a) Retenção do conhecimento: Quando um técnico sair da equipa de atendimento o seu conhecimento fica na base de conhecimento através da documentação das soluções por ele encontradas para os diferentes problemas com os quais se tenha deparado.
- b) Ausências com baixo impacto: Em caso de necessidade, designadamente por motivo de ausência de um técnico mais avançado, os técnicos de níveis mais baixos podem resolver problemas mais complexos desde que a solução adequada se encontre já documentada e disponível na base de conhecimento.
- c) Padronização: Recorrendo às soluções presentes na base de conhecimento, os técnicos podem proceder à resolução de problemas conhecidos de forma padrão.

#### ✓ **Melhoria contínua**

Este último conceito é comumente utilizado em todas as metodologias de processos do mercado. O ciclo conhecido como PDCA (*Plan, Do, Check, Act* ou Planear, Executar, Verificar, Agir) é de grande importância uma vez que garante que os processos utilizados numa determinada área se mantêm atualizados e em constante evolução, melhorando-se a produtividade. O ciclo PDCA pode ser explicado de forma simplificada como:

- a) Planear: saber quais os objetivos do departamento de TI de uma determinada empresa e planear os seus processos em concordância;
- b) Executar: executar os processos conforme planeado na etapa anterior;
- c) Verificar: analisar os resultados da execução e concluir se os objetivos foram os não atingidos da melhor forma possível;

- d) Agir: procurar pontos nos quais seja possível obter melhorias nos processos e trabalhar no sentido de as obter, recomeçando o ciclo novamente a partir da etapa de planeamento.

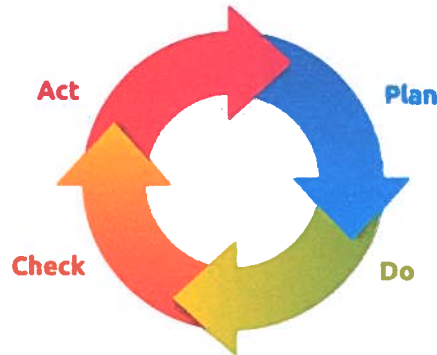


Figura 1 – Ciclo interminável de PDCA.<sup>13</sup>

## 2.3) Soluções existentes

Inseridas na área da TI e com um mercado em constante evolução e crescimento, a diversidade de soluções, tanto para *HelpDdesk* como para *ServiceDesk*, é muito vasta.

Existem soluções com alguma relevância nesta área:

*Remedy ServiceDesk;(BMC)*

*OpenviewServiceDesk(HPTM)*

*Client2Central;*

## 3. Metodologia

---

A metodologia de desenvolvimento consiste numa abordagem organizada através de um conjunto de procedimentos pré-estabelecidos para a obtenção de um determinado objetivo.

A metodologia de desenvolvimento neste projeto divide-se em duas fases.

<sup>13</sup> Figura criada pelo aluno tendo como base o artigo [http://www.mindtools.com/pages/article/newPPM\\_89.htm](http://www.mindtools.com/pages/article/newPPM_89.htm)

A primeira fase corresponde à pesquisa de informações e análise das principais tecnologias e ferramentas de *software* utilizadas em sistemas de *ServiceDesk*. Durante esta etapa, o trabalho adquire um cunho exploratório, tendo por objetivo proporcionar uma maior familiaridade com o tema e o aprimoramento de ideias. Nesta fase foram utilizadas as abordagens metodológicas, de pesquisa bibliográfica e documental. Todavia, embora semelhantes no desenvolvimento, diferem na natureza das fontes<sup>14</sup>. A pesquisa bibliográfica recaiu praticamente na sua totalidade na análise de material impresso localizado em bibliotecas. A pesquisa documental serve-se de matérias que, por serem muito diversificadas e dispersas, ainda não foram objeto de tratamento analítico. A análise foi realizada essencialmente a jornais, revistas, *websites*, folhetos, manuais, entre outros.

A próxima etapa do projeto consiste no desenvolvimento da aplicação *HelpDeskITNext*. Durante esta fase do trabalho foi utilizada a metodologia de pesquisa-ação. A escolha dessa metodologia foi motivada pela perspectiva dada ao trabalho, desempenhando um papel ativo no equacionamento do problema, considerado como central na pesquisa.

## 4. Implementação

---

A ferramenta *HelpDesk* foi dividida em duas partes:

- ✓ **Interface do utilizador:** este é o meio através do qual o utilizador interage com a aplicação *HelpDesk*. É a partir desta interface que o utilizador realiza as solicitações de suporte ao sistema e verifica o estado de uma solicitação aberta. Esta interface foi desenvolvida e implementada com recurso a tecnologia ASP.NET MVC (*Active Server Pages*), plataforma constituída por um conjunto de classes cujo principal objetivo é o de servir pedidos efetuados através do protocolo http. O *ASP.NET* é um produto que representa a visão da *Microsoft* para a construção de aplicações dinâmicas web com suporte em base de dados através da utilização da plataforma NET. O ASP.NET é baseado na framework.NET herdando todas as suas

---

<sup>14</sup>FRADA, J. J. *C.Guia prático para elaboração e apresentação de trabalhos científicos*. Lisboa : Edições Cosmo, 1995.

características, como qualquer aplicação .NET. As aplicações para essa plataforma podem ser escritas em várias linguagens, no entanto, a linguagem C# foi a escolhida para este projeto. O ambiente de desenvolvimento mais comum das aplicações *ASP.NET* é o *Visual Studio* já que possui algumas características que facilitam o trabalho do programador, como os componentes visuais para criação de formulários de páginas *Web*.<sup>15</sup>

- ✓ Base de Dados: Responsável por armazenar todas as informações relevantes do sistema, como por exemplo, problemas comuns e respetivas soluções. O Sistema de Gestão de Base de Dados escolhido foi o SQL Server 2012. A escolha do SQL prende-se como facto de este ser confiável e robusto e de ter boa integração com aplicações *Web*.<sup>16</sup>

## 4.1) Tecnologias utilizadas

Na elaboração deste projeto foram utilizadas diversas tecnologias por forma a facilitar a sua elaboração e manutenção. Serão agora descritas, sucintamente, as várias tecnologias utilizadas, designadamente a base da arquitetura de *software* utilizado e o modo como se enquadram no projeto.

### 4.1.1) Arquitetura de *Software*

Um dos principais requisitos para a elaboração deste projeto consiste na obrigatoriedade da aplicação ser desenvolvida de forma a poder ser integrada numa plataforma *WEB*. A aplicação deve estar apta a ser desenvolvida no ambiente *WEB*, evitando a instalação e configuração nos postos, beneficiando, por isso, das vantagens de facilidade de acesso ao

---

<sup>15</sup>REMOALDO, P., *O Guia Prático do ASP.NET com SQL Server 2008*. Lisboa : Centro Atlântico, 2008.

<sup>16</sup>DAMAS, L., *SQL - Structured Query Language*. s.l. : FCA.



programa, de partilha de documentos e de dados em geral e de colaboração *on-line* simplificada.

O *ASP.NET*<sup>17</sup> foi a escolha para alcançar este requisito. A razão para esta escolha está associada à natureza do projeto, o qual se trata de um trabalho académico, de final de curso, cujo objetivo é o de aplicar e desenvolver as competências adquiridas ao longo dos 3 anos de licenciatura em Engenharia Informática.

Criado pela *Microsoft*, com base no *.NET Framework*, o *ASP.NET* é uma plataforma de desenvolvimento de *websites* e aplicações web que oferece uma vasta gama de opções e características para o desenvolvimento de *sites*.

O *Visual Studio.NET* é a principal ferramenta utilizada para o desenvolvimento de aplicações na plataforma *.NET*<sup>18</sup>.

O *ASP.NET* inclui um compilador que compila todos os componentes da aplicação *web* (incluindo páginas e controlos) para uma *assembly* que o ambiente de alojamento *web* utiliza. As páginas *ASP.NET*, possuem a extensão *.CSHTML* e consistem em código *HTML*, *CSS*, *JavaScript* e código *C#*.

## 4.1.2) Ambiente de desenvolvimento *ASP.NET*

O ambiente principal de desenvolvimento do projeto (baseado em tecnologias *Microsoft*) engloba as seguintes tecnologias:

- ✓ Sistema Operativo: *Windows 7 / 8 / 10*
- ✓ Web Server: *IIS (Internet Information Services)*
- ✓ Linguagem de programação: *ASP.NET* (com *C#*)
- ✓ Microsoft Visual Studio Community Edition 2015
- ✓ Sistema de gestão de base de dados: *SQL Server 2012*

---

<sup>17</sup>*ASP.NET* – Introdução e conceitos, [Online], (<http://pplware.sapo.pt/tutoriais/asp-net-%E2%80%93-introducao-e-conceitos/>), consulta a 22 de Setembro de 2015.

<sup>18</sup>O que é *ASP.NET*? *Anselme it*. [Online], (<http://www.anselmeit.com/2012/03/o-que-e-aspnet.html>), consulta a 23 de Setembro de 2015.



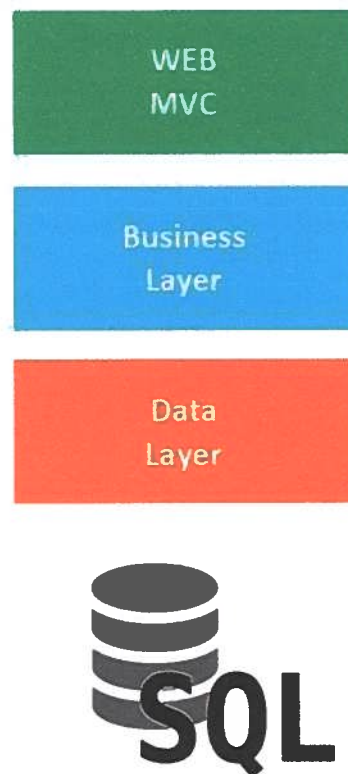


Figura 2 – Ambiente de desenvolvimento web.<sup>19</sup>

### 4.1.3) Arquitetura ASP.NET

A .NET Framework foi desenhada tendo em consideração:

**Interoperabilidade** – Entre o novo código e as bibliotecas existentes.

**CommonRuntimeEngine** – As linguagens de programação são compiladas para uma linguagem intermédia (CLI – *CommonLanguageInfrastructure*). O CLR (*CommonLanguageRuntime*) define um ambiente de execução para este código CIL e converte-o em código nativo.

---

<sup>19</sup> Imagem criada pelo aluno Tiago Diogo N°1615

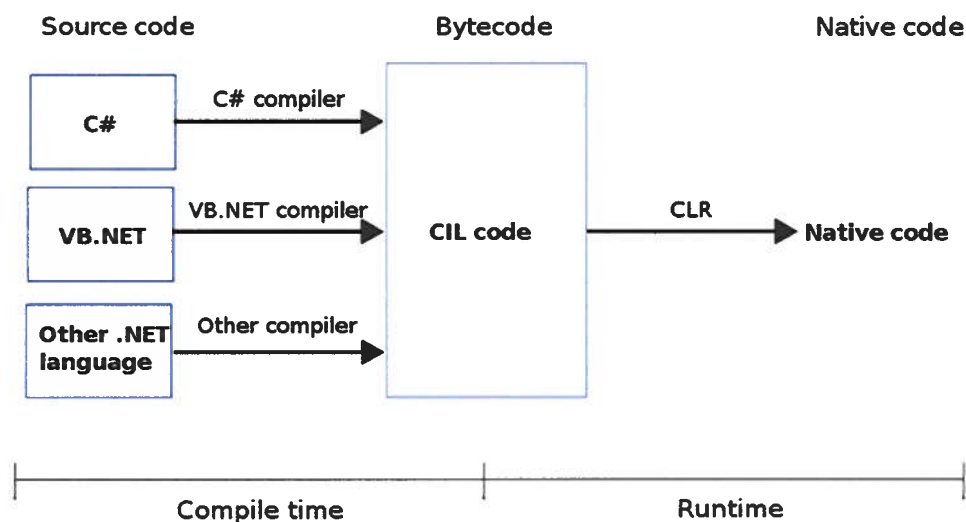


Figura 3 – Processo de conversão do código fonte para nativo<sup>20</sup>.

**Independência na Linguagem** – a *.NET Framework* suporta o desenvolvimento recorrendo a múltiplas linguagens de programação (C#, Jscript).

**Biblioteca Base de Classes** – Biblioteca de classes disponível para todas as linguagens que utilizam a *.NET Framework*.

**Desenvolvimento Simplificado** – a instalação de aplicações é simplificada na *.NET Framework* evitando os problemas de registo no *registry* e de incompatibilidades das DLL's.

**Segurança** – a *.NET Framework* permite que o código seja executado em diversos níveis de confiança sendo disponibilizado um modelo de segurança comum para todas as aplicações.

**Portabilidade** – Um programa escrito utilizando a *.NET Framework* deve poder ser executado sem alterações em qualquer tipo de sistema (Ex.: *Windows Azure*).

<sup>20</sup>Common Language Runtime, *Wikipedia*, [Online], ([http://en.wikipedia.org/wiki/Common\\_Language\\_Runtime](http://en.wikipedia.org/wiki/Common_Language_Runtime)), consulta a 10 de Outubro de 2015.

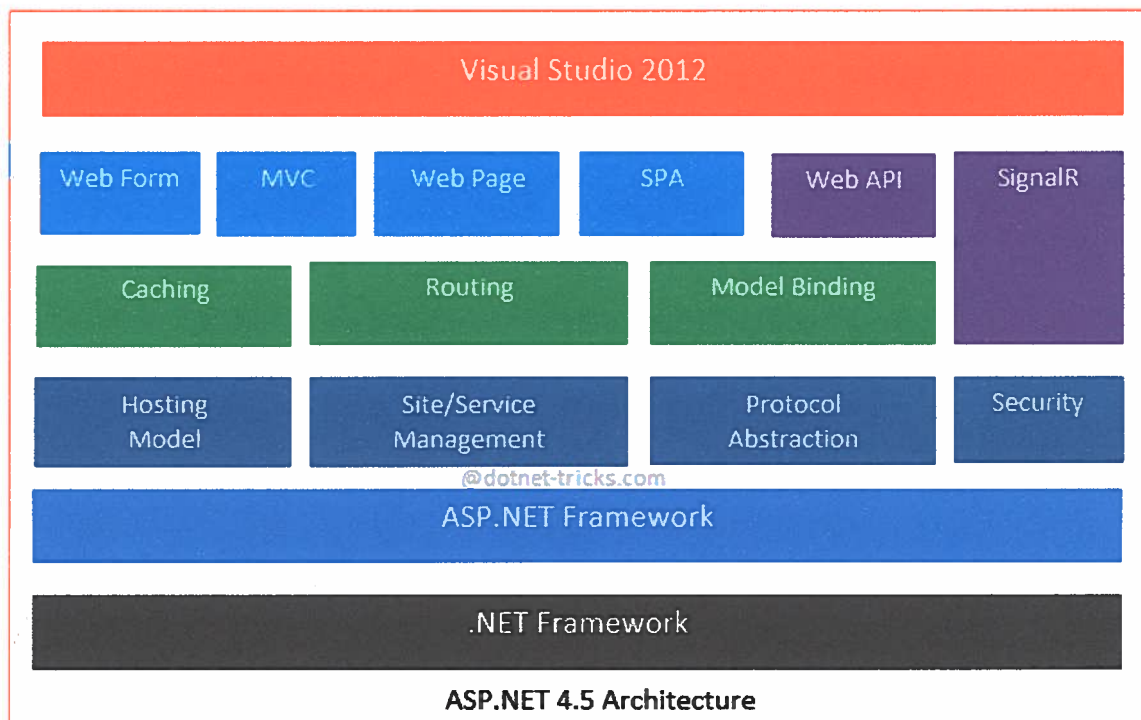


Figura 4 – Arquitectura. NET Framework<sup>21</sup>.

#### 4.1.4) Base de Dados

A maior parte dos *websites* existentes na *Web* utiliza páginas de forma dinâmica que recorrem a dados armazenados em diferentes formatos, como por exemplo, base de dados, XML, JSON e até mesmo ficheiros de texto. A utilização de ficheiros texto para armazenamento de dados, embora gratuita, apresenta problemas quando se pretende aceder à informação rapidamente, e, por outro lado, não é muito escalável. Por seu lado, o XML é muito útil para manipular pequenas quantidades de dados. Todavia, quando se trata de processar grandes quantidades de informação, fica prejudicada a velocidade de transferência.<sup>22</sup>

<sup>21</sup>Arquitectura .NET , [Online], (<http://www.dotnet-tricks.com/>), consulta a 24 de Outubro de 2015.

<sup>22</sup>PEREIRA, J. L. *Tecnologia de Bases de Dados*, 3ª Edição, FCA, 1998.

#### 4.1.4.1) XML vs JSON: A diferença de desempenho

A escolha entre o XML ou o JSON é praticamente como a escolha entre um banco de dados relacional ou um NoSQL: tudo se resume a escolher um esquema explícito ou um implícito.

Um *schema* é uma descrição de um conjunto de dados, o qual determina o tipo de dados, o respetivo formato, a relação entre eles, como estão estruturados, etc. Se esses dados forem para consumo humano, o *schema* deixa de ser tão relevante já que temos a capacidade de ler e interpretar diretamente o conteúdo ainda que o mesmo esteja mal estruturado e desarrumado. Todavia, se os dados são para consumo da máquina (i.e. se serão lidos e manipulados por outros programas) então um *schema* formal será sempre necessário, esteja codificado em *meta-data* ou apenas convencionado.

Tal como sucede numa BD relacional, será necessário criar tabelas, definir as colunas, chaves e restrições num arquivo XML. Este poderá conter o mesmo tipo de descrição, dizendo quais os elementos aceites, quais os atributos e subelementos que podem ter, etc. Assim, será possível automatizar o processo de verificação de um dado XML por forma a concluir se é ou não válido de acordo com sua especificação. Ou seja, da mesma forma que não será possível atribuir uma coluna que não existe numa BD, não será possível criar um elemento que não exista no *schema* e o XML ainda ser considerado válido.

Por outro lado, é possível criar um XML sem *schema* no qual todo conteúdo seja válido desde que a sintaxe esteja correta. Será, portanto, responsabilidade do programador garantir que esse conteúdo seja gerado e consumido segundo uma regra lógica. Caso haja uma equipa grande ou um conjunto de equipas ou mesmo de organizações distintas tornar-se-á mais difícil controlar as alterações e garantir que todos os sub-sistemas envolvidos tratam das mesmas corretamente<sup>23</sup>.

Por outro lado, se se verifica que, dadas as características particulares de uma situação, o uso de um *schema* específico não é o mais apropriado, porque não se usa, ao invés, o XML?

---

<sup>23</sup>StackOverFlow, [Online], (<http://pt.stackoverflow.com/questions/19423/por-que-e-quando-usar-xml-em-vez-de-json>), consulta a 5 de Setembro de 2015.

Neste caso, as desvantagens do XML tornam-no uma solução menos vantajosa pois a sintaxe é verbosa, os arquivos tornam-se grandes e, portanto, difíceis de ler e escrever. Por esse motivo, torna o processamento de XML mais difícil e lento.

Para este efeito, uma solução mais "leve" será preferível. Da mesma forma que uma BD *schema-less* será mais simples de usar e terá melhor desempenho do que um BD relacional tentando tratar dados arbitrários, sem estrutura.

O JSON mostrou-se uma solução adequada para esses casos. Não é a única, nem necessariamente a melhor mas como muitas vezes o lado "cliente" da comunicação é um *browser* que suporta JavaScript nativamente, essa proximidade adicional entre o formato e os literais da própria linguagem traz a vantagem adicional de simplificar bastante o código cliente.

Foi um fator decisivo para consolidar o JSON como a alternativa "leve" preferida ao XML. E considerando que o JSON passou a receber suporte na maioria das plataformas, tornou-o cada vez mais uma preferência para quem programa<sup>24</sup>.

#### 4.1.4.2) Base de dados SQL - (*StructuredQueryLanguage*)

O SQL *Server* é um sistema de gestão de base de dados relacionais da *Microsoft* adequado para ambientes empresariais e para sites *web*. É utilizada a versão *SQL Server 2012 Express* por ser uma versão gratuita, sendo ideal para a natureza deste projeto. Além de ser de fácil utilização e escalável, o motor de base de dados é bastante fiável<sup>25</sup>.

As bases de dados relacionais são o mecanismo mais utilizado para armazenar dados consistindo em conjuntos de tabelas relacionadas entre si através de uma chave e em que cada tabela representa tipicamente uma entidade do mundo real. O modelo relacional estabelece as regras para a divisão da informação entre tabelas de modo a prevenir a duplicação de

---

<sup>24</sup>StackOverFlow, [Online], (<http://pt.stackoverflow.com/questions/19423/por-que-e-quando-usar-xml-em-vez-de-json>), consulta a 5 de Setembro de 2015.

<sup>25</sup>Notassobre SQ, [Online], (<http://www.di.uevora.pt/~ccaldeira/p/docs/docbook/sql.html>), consulta a 5 de Setembro de 2015.

informação<sup>26</sup>. Esta dispersão de informação por diferentes tabelas é facilmente manipulável através da linguagem SQL uma vez que a ligação entre as mesmas se realiza através de chaves estrangeiras<sup>27</sup>.

A figura seguinte representa as tabelas usadas na base de dados da aplicação *web HelpDeskITNext*.

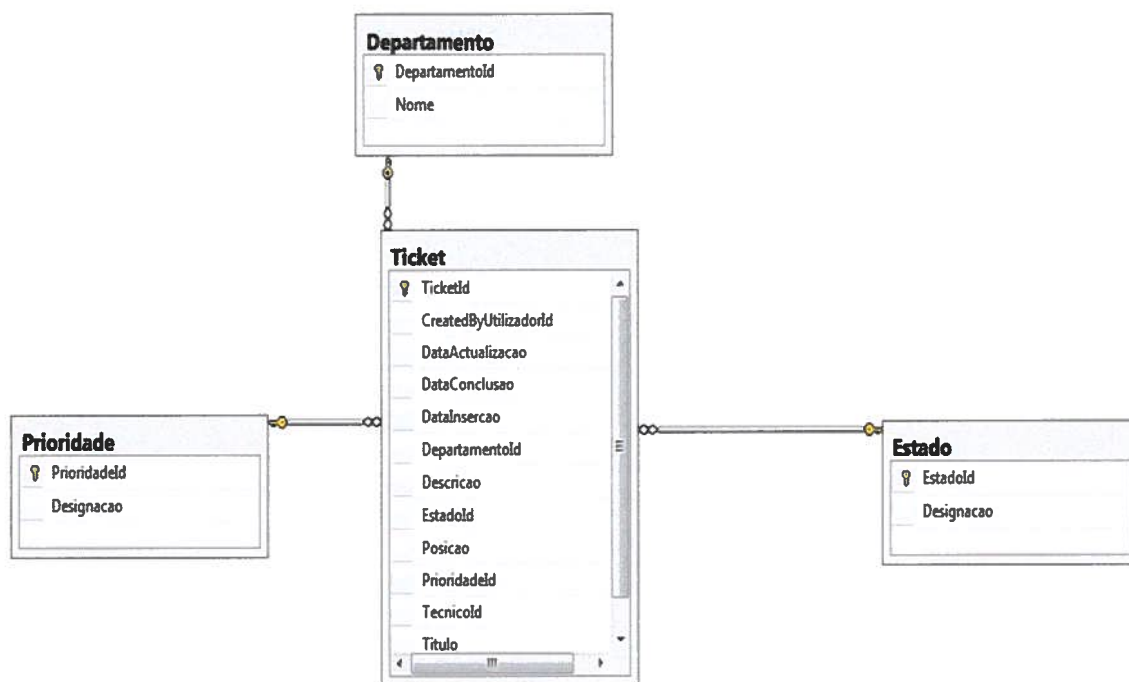


Figura 5 – Diagrama das diferentes tabelas e seus relacionamentos.<sup>28</sup>

<sup>26</sup>SQL - STRUCTURED QUERY LANGUAGE, [Online], (<http://www.prof2000.pt/users/mjoaol/eotd/unidade5/sql.htm>), consulta a 4 de Junho de 2015.

<sup>27</sup>Foreign Key, [Online], ([http://en.wikipedia.org/wiki/Foreign\\_key](http://en.wikipedia.org/wiki/Foreign_key)), consulta a 5 de Junho de 2015.

<sup>28</sup> Figura desenhada pelo aluno AL1615 Tiago Diogo através do SQL.



Figura 6 – Esquema da base de dados de utilizadores.<sup>29</sup>

Neste projeto é utilizado o SGBD (Serviço de Gestão de Base de Dados) do SQL Server 2012 para guardar os dados utilizados na produção das páginas. Este sistema, gerador de base de dados relacionais, adaptado para a computação cliente/servidor distribuída tem as seguintes características:

- ✓ Independência física de dados
- ✓ Independência lógica de dados
- ✓ Otimização de consulta
- ✓ Integridade de dados
- ✓ Controlo concorrente
- ✓ Backup e restauro
- ✓ Segurança e autorização

<sup>29</sup> Figura desenhada pelo aluno AL1615 Tiago Diogo através do SQL.

### 4.1.4.3) Acesso à Base de Dados (*Entity Framework*)

A escrita de código ADO.Net para acesso a dados é um trabalho tedioso e monótono. A *Microsoft* forneceu uma *Framework O/RM* chamada "Entity Framework" para automatizar as atividades relacionadas com as bases de dados para a aplicação.

A *Microsoft* ADO.NET Entity Framework é uma *Object / RelationalMapping (ORM)* que permite trabalhar com dados relacionais como objetos específicos de domínio, eliminando a necessidade da maior parte do código de canalização de acesso a dados que normalmente são necessários. Usando o *Entity Framework* será possível realiza e consultas usando o LINQ (*Language-IntegratedQuery*), e em seguida, recuperar e manipular dados. Com a implementação do ORM do Entity Framework podemos nos concentrar na lógica específica para a aplicação.

A Entity Framework é uma estrutura de O/RM. É um acessório para ADO.NET que permite um mecanismo automatizado para aceder e armazenar os dados na base de dados.

Esta ferramenta é útil para os 3 seguintes cenários:

- Caso exista uma base de dados ou caso queiramos desenhar uma base de dados à frente de partes da aplicação:

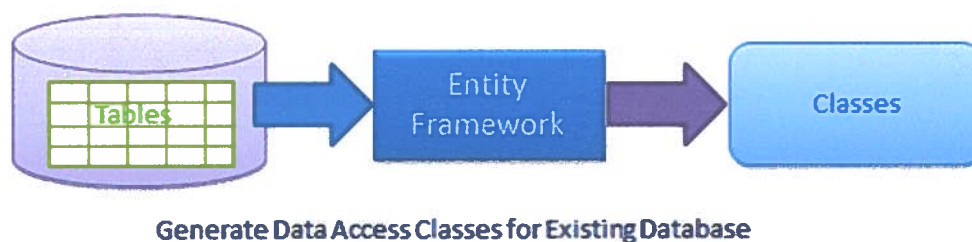


Figura 7 – Diagrama da Entity Framework – cenário 1.<sup>30</sup>

<sup>30</sup>Figura desenhada pelo aluno AL1615 Tiago Diogo



- Se for necessário focarmo-nos nas classes de domínio e depois criar uma base de dados através delas:

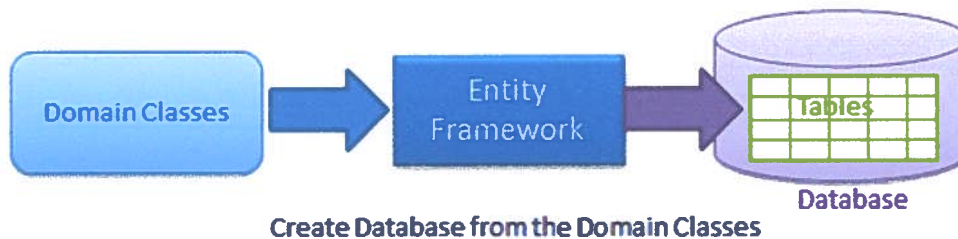


Figura 8 –Diagrama da Entity Framework – cenário 2.<sup>31</sup>

- Caso seja necessário desenhar uma *databaseschema* no *design* visual e, de seguida, criar a base de dados e classes:

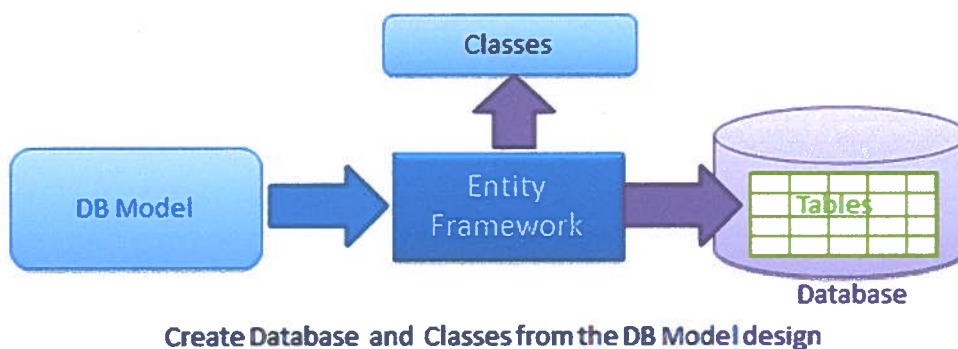


Figura 9 – Diagrama da Entity Framework - cenário 3.<sup>32</sup>

#### 4.1.4.4) Configuração da aplicação

O conceito de ficheiro de configuração foi introduzido desde a primeira versão da plataforma ASP.NET. Um ficheiro de configuração não é mais do que um ficheiro XML que obedece a um esquema predefinido.

Este ficheiro (web.config) encontra-se localizado na raiz da aplicação e contém as definições aplicadas a cada recurso, sendo calculadas de forma cumulativa, isto é, as

<sup>31</sup>Figura desenhada pelo aluno AL1615 Tiago Diogo

<sup>32</sup>Figura desenhada pelo aluno AL1615 Tiago Diogo

definições utilizadas em cada local resultam da união de todas as definições dos ficheiros de configuração existentes na hierarquia, até à pasta onde nos encontramos.

Um ficheiro de configuração é, antes de mais, um ficheiro de XML. Logo, aplicam-se as regras normalmente associadas a este tipo de documentos. Todas as definições introduzidas num ficheiro webconfig têm de ser delimitadas por um elemento designado de <configuration><sup>33</sup>.

## 4.1.5) O que é o Twilio?

O Twilio é uma plataforma de API que facilita a realização e receção de chamadas telefónicas, o envio e receção de mensagens de texto e, ainda, a inserção de chamadas VoIP em aplicações móveis.

### 4.1.5.1) Receção de chamadas e mensagens de texto

O Twilio permite-nos comprar números de telefone programáveis que podem ser usados para enviar e receber chamadas e mensagens de texto. Quando um número do Twilio recebe uma chamada de entrada ou texto, o Twilio envia uma solicitação HTTP POST ou GET ao seu aplicativo Web, solicitando instruções sobre como lidar com o texto ou a chamada. O servidor responderá à solicitação HTTP do Twilio com TwiML, um conjunto simples de marcas XML que contém instruções sobre como lidar com uma chamada ou texto.

### 4.1.5.2) Realização de chamadas e envio de mensagens de texto

Ao fazer solicitações HTTP para a API do serviço web do Twilio, poderão ser enviar mensagens de texto ou iniciadas chamadas telefónicas de saída. Para chamadas de saída,

---

<sup>33</sup>ABREU, L. e CARREIRO, J. P. *ASP.NET 2.0 - Curso Completo*. s.l. : FCA, 2006.

deverá especificar-se uma URL que retorne instruções TwiML sobre como lidar com a chamada de saída.

## 5. Análise da Aplicação

---

Neste capítulo a aplicação criada será apresentada, sendo desenvolvidos todos os componentes através do recurso a diagramas para exemplificar o processamento da informação.

### 5.1) Visão original do sistema

A visão original do projeto consistiu na análise, desenho e desenvolvimento do sistema *HelpDesk-ITNext* com base nas seguintes características fundamentais:

- ✓ Estrutura web;
- ✓ Permissão de diferentes níveis de acesso;
- ✓ Facilidade de utilização - uso e segurança.

### 5.2) Requisitos não funcionais

Nesta secção apresentam-se sucintamente os principais requisitos não funcionais definidos originalmente no planeamento do projeto.

- ✓ Tecnologia Web:

Usar tecnologia *web* foi uma condição imposta desde o início do projeto. Por essa razão, foram implementadas tecnologias HTML, JavaScript, CSS e jQuery.

- ✓ Uso:

Outro requisito foi o do uso da aplicação. Pretende-se que a utilização seja de fácil utilização para qualquer utilizador.

✓ Segurança:

Os parâmetros de segurança para a aplicação *HelpDesk* foram determinados através da utilização de *ASP.NET Identity*. É o novo sistema de associação para aplicações *ASP.NET*. Torna muito fácil a personalização do perfil e a adição de funcionalidades de *Login / Logout*. Foi determinado o uso de um *browser* para efetuar a gestão da aplicação *ASP.NET* local ou remotamente com os privilégios administrativos. É possível criar ou delegar tarefas administrativas, bem como privilégios ou restrições.

Neste projeto foram utilizados os seguintes mecanismos:

- ✓ Segurança – Configuração da segurança da aplicação web e dos respetivos recursos.
- ✓ Fornecedor – Indica o *databaseprovider* para a base de dados ASPNETDB (utilizado pelo SQL).

## 5.3) Requisitos funcionais

Nesta secção apresentam-se os principais requisitos funcionais definidos no início do planeamento do projeto.

### 5.3.1) Criação de utilizadores

A finalidade deste projeto é a de servir os utilizadores, ajudando-os a encontrar resposta às suas questões e resolver os seus problemas informáticos. Daí que o início da aplicação seja a criação de um perfil de utilizador. Cada utilizador pode especificar o grupo de utilizadores ao qual pertence. Os utilizadores terão, assim, permissão para criar, visualizar/consultar, editar e fechar os *tickets* que por ele foram criados na aplicação. (Fig. 10)



Figura 10 - Esquema de atividade do utilizador.

**Pré-Requisitos:** O nome do utilizador criado é o seu *e-mail* de registo e a *password* tem o limite mínimo de 6 caracteres sendo obrigatório o uso de letras maiúsculas e minúsculas, um algarismo ou caracteres especiais. Todos os campos são de preenchimento obrigatório, com exceção do campo Departamento, o qual é de preenchimento é opcional (para os caso sem que o utilizador não tenha conhecimento de qual é o seu departamento ou para situações nas quais tenha recentemente criado um departamento novo).

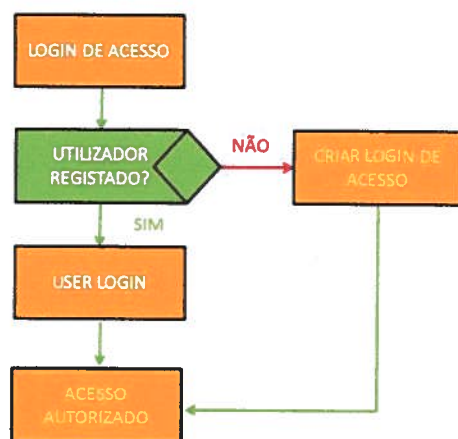
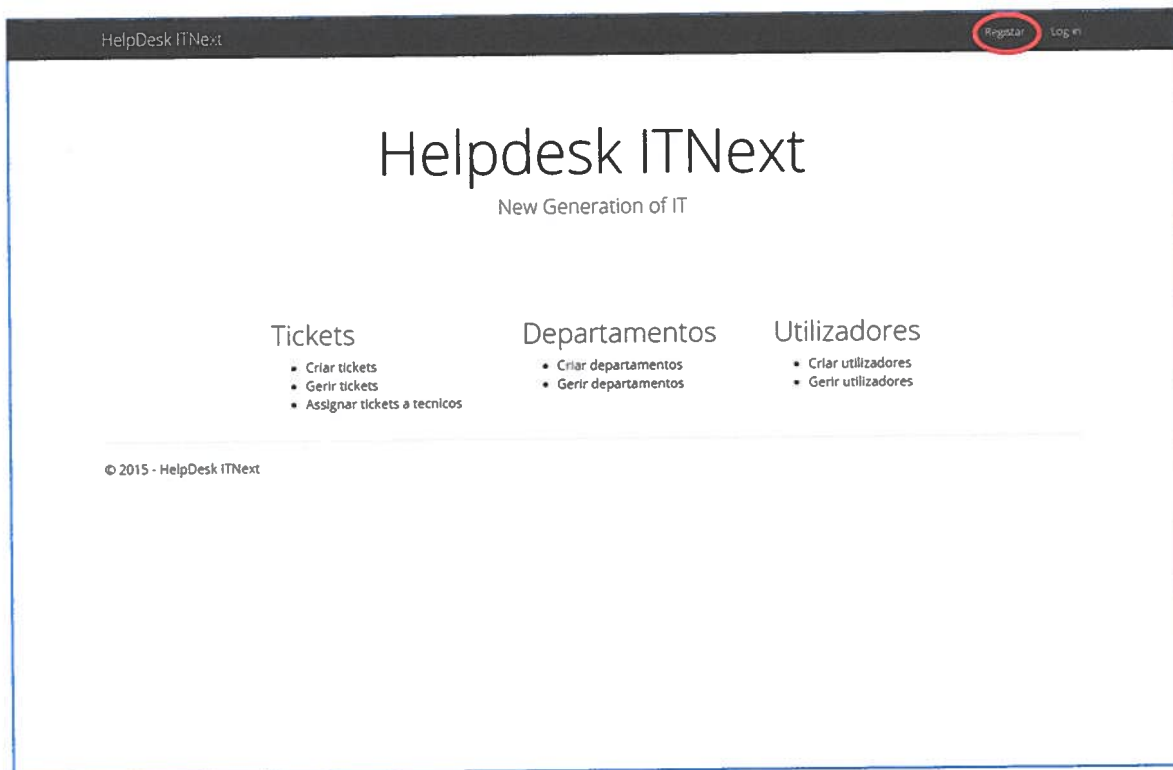


Figura 11 – Esquema de acesso à aplicação.

- 1 – Caso o utilizador não inicie a sessão, não conseguirá aceder às opções. Terá de fazer o Login. (fig. 12).
- 2 – Caso não esteja registado, pode fazê-lo na opção, “registre-se” (fig. 13).
- 3 – Durante o registo, o novo utilizador deve preencher todos os campos, tendo a *password* de obedecer aos pré-requisitos estipulados acima (fig. 13).

4 – Por fim, a sessão é aberta com identificação do respetivo nome do utilizador no canto superior direito do ecrã.



**Figura 12** – Ecrã de *login* da aplicação com a indicação para registo de utilizador

HelpDesk ITNext Register Log in

Criar novo utilizador

- The Email field is required.
- The Password field is required.
- The Nome field is required.
- The Telefone field is required.

Nome:   
The Nome field is required.

Email:   
The Email field is required.

Telefone:   
The Telefone field is required.

Password:   
The Password field is required.

Confirmar password:

Departamento:

© 2015 - HelpDesk ITNext

**Figura 13** – Criação de nova conta de utilizador Tiago Diogo (a título de exemplo).

HelpDesk ITNext Tickets  Log off

# Helpdesk ITNext

New Generation of IT

**Tickets**

- Criar tickets
- Gerir tickets
- Assignar tickets a técnicos

**Departamentos**

- Criar departamentos
- Gerir departamentos

**Utilizadores**

- Criar utilizadores
- Gerir utilizadores

© 2015 - HelpDesk ITNext

**Figura 14** – Ecrã com login efectuado com sucesso.

### 5.3.2) Abertura de *Tickets*

Uma vez aberta a sessão com um utilizador registado, será possível reportar o problema em causa. Por razões de simplificações e de rapidez, foram criados simplesmente os campos “Título”, “Descrição”, “Prioridade” e “Departamento”. Os restantes campos, designadamente o referente ao estado do *ticket* ou à data da abertura do *ticket*, serão inseridos automaticamente.

**Título:** destina-se ao *subject* do “problema” (por exemplo: criação de uma conta de *e-mail* ou problemas de internet, entre outros). Tem um limite de 20 caracteres. Diz respeito ao tópico do que levou à criação do incidente e não ao corpo da mensagem.

**Descrição:** Neste campo o utilizador deverá reportar o pedido que originou a criação do incidente. O utilizador deverá explicar o incidente de forma sucinta e coesa uma vez que este campo tem um limite de 200 caracteres.

**Prioridade:** A prioridade fica ao critério do utilizador tendo em conta a tabela de prioridades (Figura 12).

**Departamento:** Tem como finalidade identificar o departamento ao qual será afeto o erro ou requisição uma vez que o utilizador em causa poderá não estar integrado no departamento ao qual o incidente se destina.

A data da abertura do *ticket* é a data atribuída pelo sistema e o estado do mesmo é colocado sempre no estado Novo.

A gravação do *ticket* desencadeia as seguintes ações por parte da aplicação:

- ✓ Criação de um número de *ticket*;
- ✓ Atribuição de data e hora;
- ✓ Definição do estado do *ticket* como Aberto;
- ✓ Alocação automática a um dos técnicos assignados pelo Administrador;
- ✓ Envio de mensagem ao técnico com abertura de *ticket* (*Fig.*);



Tabela de Prioridades			
Prioridade	Resposta	Resolução	Exemplos
<b>Alta</b>	≤ 30 min	≤ 6 horas	Falha na comunicação dos dados Compromisso de segurança (difusão de vírus/ malware) Problemas com aplicações que envolvam transações financeiras
<b>Média</b>	≤ 1 hora	≤ 2 dias	Problemas causados por falhas de rede local Falha de acesso à conta de utilizador Falha de acesso à plataforma
<b>Baixa</b>	≤ 2 dias	Logo que possível	Apoio logístico nas ferramentas <i>office</i> Problemas com configuração de <i>e-mail</i> Problemas de impressão Instalação de programas e apoio aos utilizadores

Figura 15 – Tabela criada com as prioridades e o grau de importância de cada estado

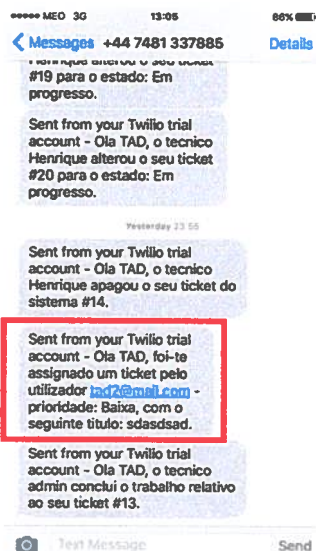


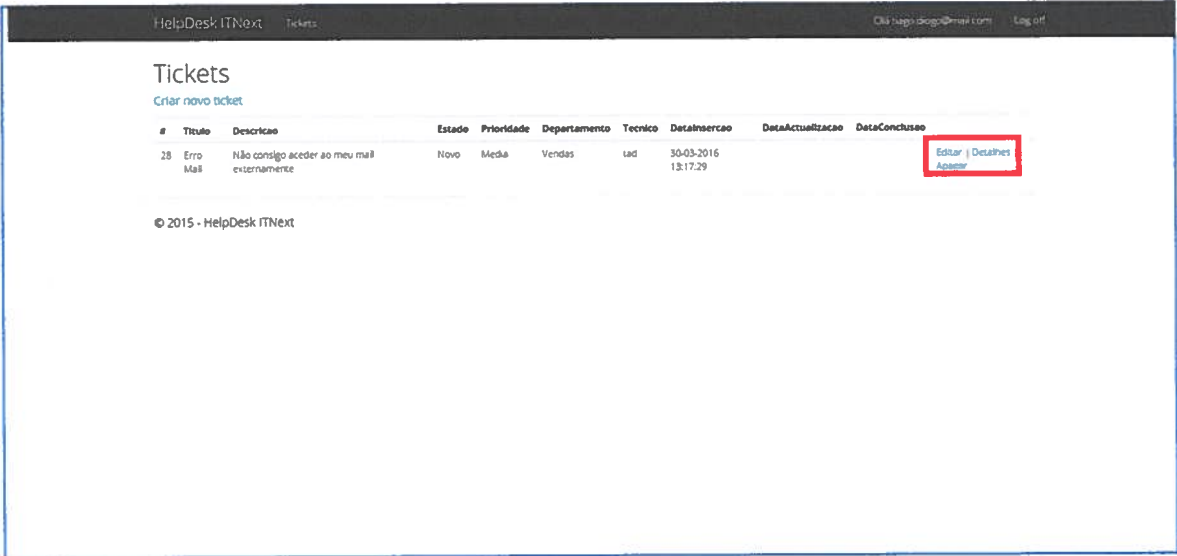
Figura 16 – Mensagem recebida por um dos técnicos assignados, com um ticket atribuído

Após a criação do *ticket*, o utilizador tem acesso ao relatório de criação do incidente com a data e hora em que o mesmo foi criado, bem como à identificação do técnico alocado para a resolução do mesmo. Caso algo tenha surgido alguma falha no pedido, o utilizador poderá ainda editar o pedido ou até mesmo apagá-lo caso o mesmo já não se confirme (Fig. 13).

A consulta do estado do *ticket* pode ser efetuada via *SMS*, através do envio do seguinte texto na mensagem:

- “CONSULTA: 20” (Exemplo).

Sendo que “CONSULTA” é o parâmetro de pesquisa, e “20” é o número do *ticket* atribuído no ato da criação. (Fig. 17)



#	Título	Descrição	Estado	Prioridade	Departamento	Técnico	DataInsercao	DataActualizacao	DataConclusao	
28	Erro Mail	Não consigo aceder ao meu mail externamente	Novo	Meda	Vendas	Iad	30-03-2016 13:17:29			Editar   Detalhes Apagar

Figura 17 – Ecrã de relatório de *ticket* com destaque para a edição do mesmo.

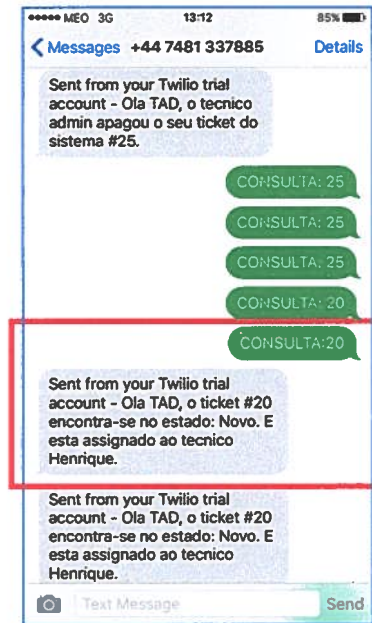


Figura 18 – Relatório do pedido e estado do *ticket* via SMS.

### 5.3.3) Utilizador

Todos os utilizadores registados na aplicação são, por defeito, utilizadores comuns, tendo apenas permissão para criar, editar e fechar *tickets*. Cabe ao Administrador da ferramenta alterar a categoria dos utilizadores para Administrador ou Técnico, consoante a necessidade.

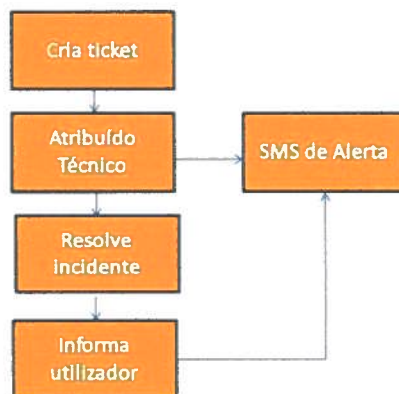


Figura 19 – Ciclo de vida de um *ticket*.

### 5.3.4) Técnico

A mensagem enviada para o técnico contém no assunto a identificação do criador do *ticket* e no corpo da mensagem a descrição do problema, bem como a sua prioridade.

O Técnico é o responsável pela gestão de todos os incidentes da aplicação *Helpdesk*, designadamente pelos estados dos tickets, passando-os do estado “Novo” para os estados “Em Progresso” ou “Concluído”, com base nas prioridades anteriormente definidas pelo o utilizador.

Na opção “Gerir *Tickets*”, o Técnico tem acesso à lista dos *tickets* por estados, podendo os mesmos ser alterados com um simples *Drag and Drop*. Para melhor identificação da prioridade dos pedidos, estão categorizados por cores:

- ✓ Verde: baixo
- ✓ Amarelo: Médio
- ✓ Vermelho: Alta

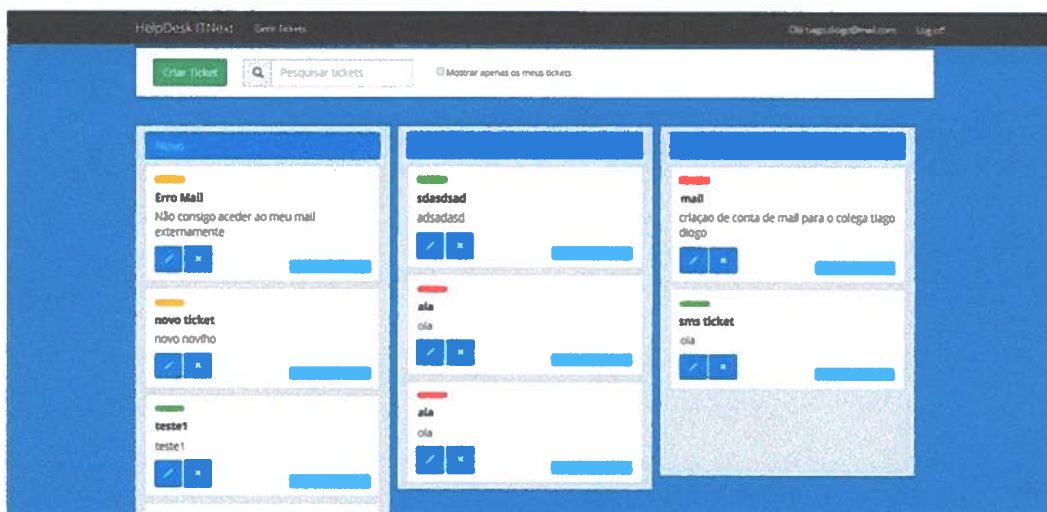


Figura 20 – Ecrã de gestão de *tickets* por um dos técnicos da ferramenta.

### 5.3.5) Administrador

Ao contrário do Técnico, o Administrador não recebe a mensagem enviada para o sistema sendo o responsável pela gestão da ferramenta. Pode também atribuir ou mover os estados dos *tickets*, colocando-os num dos três estados possíveis. É também responsável pelos departamentos, utilizadores, prioridades e estados existentes na aplicação, sendo-lhe permitido criar, modificar ou até apagá-los consoante as necessidades. Tem acesso ao separador “Admin” que lhe permite os acessos acima descritos. (Fig. 21)

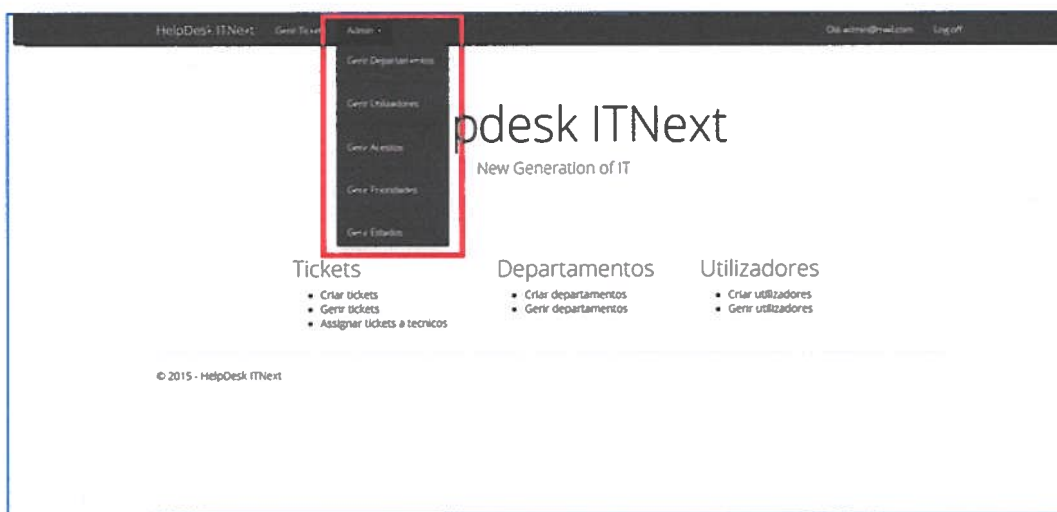


Figura 21 – Ecrã de gestão de acessos do Administrador

## 6. Conclusão

---

Analisando os objetivos a que me propus e o trabalho desenvolvido no sentido da criação de uma aplicação / ferramenta de *Helpdesk* pode considerar-se que o projeto foi bem-sucedido tendo sido atingidos os seguintes objetivos:

- ✓ Desenvolvimento de um sistema de suporte ao utilizador (*Helpdesk*) capaz de registar os pedidos de assistência técnica numa base de dados, permitindo simultaneamente o aumento da qualidade de serviço da equipa de TI na empresa “MCDiogo – Mediação de Seguros LDA”;
- ✓ Desenvolvimento de um trabalho através da aplicação de técnicas e conceitos abordados durante a licenciatura em Informática;
- ✓ Aperfeiçoamento dos conhecimentos adquiridos e estudo de novas ferramentas e tecnologias de desenvolvimento.

No âmbito deste último ponto foi possível ampliar os conhecimentos adquiridos relativos ao desenvolvimento de aplicações usando a tecnologia ASP.NET, MVC e Twilio. Este trabalho permitiu também o aprofundamento das competências no uso da ferramenta de base de dados SQL.

É ainda importante referir que, após a apresentação do trabalho junto da MCDiogo, a administração adjudicou a implementação do projeto na empresa. Esta adjudicação abona fortemente a favor do projeto, constituindo mais um fator no sentido de se considerar o desenvolvimento desta aplicação um sucesso.

# Referências

---

## Bibliografia

**ABREU, L. e CARREIRO, J. P.** *ASP.NET 2.0 - Curso Completo*. s.l. : FCA, 2006.

**FRADA, J. J. C.** *Guia prático para elaboração e apresentação de trabalhos científicos*. Lisboa : Edições Cosmo, 1995.

**MAGALHÃES, I. L. e PINHEIRO, W. B.**, *Gerenciamento de Serviços de IT na prática: uma abordagem com base na ITIL*, São Paulo : Novatec, 2007.

**MENDONÇA, F. H. de.** Benefícios da aplicação da ITIL em empresas – com foco na Gestão de Problemas. [Online], (<http://www.fatecsp.br/dti/tcc/tcc0032.pdf>),

**PEREIRA, J. L.** *Tecnologia de Bases de Dados*, 3ª Edição, FCA, 1998.

### **HELPDESK**

<http://www.significados.com.br/help-desk/>

### **ServiceDesk**

<http://www.significados.com.br/service-desk/>

<http://www.bmc.com/blogs/help-desk-vs-service-desk-whats-difference/>

<http://www.arincmanagementservices.com/blog/2011/02/part-3-what-are-the-critical-differences-between-a-service-desk-and-a-help-desk/>

### **ITIL**

[http://umanitoba.ca/campus/physical\\_plant/media/Single Point of Contact-Mar 4 2011.pdf](http://umanitoba.ca/campus/physical_plant/media/Single_Point_of_Contact-Mar_4_2011.pdf)

<http://www.arincmanagementservices.com/blog/2011/02/part-3-what-are-the-critical-differences-between-a-service-desk-and-a-help-desk/>

### **Arquitectura .NET**

<https://msdn.microsoft.com/es-es/library/ms954595.aspx>

[https://msdn.microsoft.com/es-es/library/9s7k7ce5\(v=vs.90\).aspx](https://msdn.microsoft.com/es-es/library/9s7k7ce5(v=vs.90).aspx)

[http://www.colimbo.net/documentos/documentacion/208/IPO05-Conceptos\\_de\\_dot\\_NET.pdf](http://www.colimbo.net/documentos/documentacion/208/IPO05-Conceptos_de_dot_NET.pdf)

### **Common Language runtime**

[https://en.wikipedia.org/wiki/Common\\_Language\\_Runtime](https://en.wikipedia.org/wiki/Common_Language_Runtime)

### **SQL - STRUCTURED QUERY LANGUAGE**

<http://www.prof2000.pt/users/mjoaol/eotd/unidade5/sql.htm>

[https://msdn.microsoft.com/en-us/library/ms714670\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/ms714670(v=vs.85).aspx)

### **Chaves estrangeiras**

[https://en.wikipedia.org/wiki/Foreign\\_key](https://en.wikipedia.org/wiki/Foreign_key)

### **Entity framework**

<http://www.entityframeworktutorial.net/what-is-entityframework.aspx>

<https://msdn.microsoft.com/en-us/library/ms178359.aspx#dbfmfcf>

<https://msdn.microsoft.com/pt-br/data/aa937709.aspx>

### **XML vs JSON**

<http://pt.stackoverflow.com/questions/19423/por-que-e-quando-usar-xml-em-vez-de-json>

<http://www.infoq.com/br/news/2013/11/xml-json-performance>

### **Twilio**

<https://www.twilio.com/>

<https://azure.microsoft.com/pt-br/documentation/articles/partner-twilio-nodejs-how-to-use-voice-sms/>