



Instituto Superior de Tecnologias Avançadas

Licenciatura Informática

Turma da manhã

**Elaboração de uma aplicação para uma empresa de
restauração**

Realizado por Rafael Serra Martins – N° 1833

Orientador: Professor Doutor Pedro Ramos Brandão

Orientador Metodológico: Professor Doutor Pedro Ramos Brandão

Lisboa

2014/2015

Instituto Superior de Tecnologias Avançadas

Licenciatura em Informática

Turma da Manhã

Projecto Global

Professor Doutor Pedro Ramos Brandão

**Elaboração de uma aplicação para uma empresa de
restauração**

12

Rafael Serra Martins

Setembro

2015

Dedicado aos meus pais e à minha irmã

Agradecimentos

Gostaria de agradecer, a todos os que de forma direta ou indireta, colaboraram na elaboração deste projeto, nomeadamente:

- Professor Doutor Pedro Ramos Brandão pela sua disponibilidade e acompanhamento metodológico ao longo deste trabalho;

- Professor José Neves pela sua total disponibilidade na ajuda em componentes científicas do trabalho;

- Da Joana de Almeida Brito pela revisão do capítulo “*abstract*” e ao seu incansável apoio e motivação;

- Ao Instituto Superior de Tecnologias Avançadas por me ter dado a formação necessária no curso de Informática possibilitando assim o desenvolvimento deste projeto.

Resumo

A presente sociedade em que vivemos é uma sociedade competitiva em que as organizações dependem cada vez mais da tecnologia de Sistemas de Informação de forma a serem capazes de gerir os seus negócios e torná-los rentáveis. A Informação Digital é permitida pelos Sistemas de Informação que têm vindo a ganhar mais importância devido a serem recursos tecnológicos e estratégicos por parte das entidades que os utilizam. Desta forma, estas mesmas entidades poderão atingir resultados positivos e satisfatórios no que toca ao ponto de vista económico e à comercialização dos seus produtos.

As aplicações que pertencem às tecnologias de Sistemas de Informação facultam os serviços prestados por uma unidade organizacional e devem ser disponibilizados de forma eficaz, segura e prática com o objetivo de ir ao encontro das necessidades dos seus clientes.

Assim, torna-se imprescindível que as organizações atuais possuam plataformas que permitam gerir os seus serviços de forma adequada, criando um meio de comunicação com os seus clientes e solicitar ajuda, sempre que o entendam.

O presente trabalho pretende ir ao encontro das necessidades de uma empresa de restauração, com o objetivo de salientar a importância da Gestão de Pedidos, desenvolvendo-se para isso, uma aplicação que vai de encontro às necessidades dos seus clientes.

Palavras chave: *Website*, Gestão de encomendas.

Abstract

This society we live in is a competitive society, where organizations increasingly rely on technology information systems to be able to manage their businesses and make them profitable. Digital information is permitted by the Information Systems that have been gaining more importance due to be technological and strategic resources by the entities that use them. Thus, these same organizations can achieve positive results regarding the marketing of its products and the economic side.

Applications from Information Systems technologies provide the services of an organizational unit and must be made available in effective, safe and practical way, in order to meet their customer's needs. Thus, it is essential that existing organizations have a platform that allows the administrator to manage the company services properly, creating a means of communication with their clients, be up to date and ask for help whenever they need.

This paper aims to meet the needs of a restauration company, in order to meet the importance of Order Management, developing for this, an application that meets the needs of its customers.

Abreviaturas

BI – Business Intelligence

.gif – Graphic Interchange Format

HA – High Availability

IIS – Internet Information Services

PHP – Personal Home Page

.png – Portable Network Graphics

T-SQL – Transact Structured Query Language

URL – Uniform Resource Locator

TI – Tecnologia de Informação

SQL – Structured Query Language

.jpeg – Joing Photographic Experts Group

ASP - Active Server Pages

ÍNDICE

Introdução	11
Estado de Arte	13
Contextualização	18
Desenvolvimento	19
Desenvolvimento empírico do Website	21
Testes de usabilidade	38
Análise dos dados recolhidos	39
Conclusão	43
Referências Bibliográficas	45
Anexos	47
Anexo I – Questionário	47

ÍNDICE DE FIGURAS E GRÁFICOS

Figura 1 – ETAPAS DE DESENVOLVIMENTO	19
Figura 2 – CRIAÇÃO DE UMA TABELA	21
Figura 3 – CONEXÃO À BASE DE DADOS	22
Figura 4 – CRIAÇÃO DE UMA TABELA	22
Figura 5 - fileupload	22
Figura 6 – FORMULÁRIO A INTRODUIR NA BASE DE DADOS	23
Figura 7 – INSERÇÃO NA BASE DE DADOS	23
Figura 8 – LIBRARY 1	23
Figura 9 - CONTROLO	24
Figura 10 – LEITURA DA EXTENSÃO DE FICHEIROS	24
Figura 11 – CONDIÇÃO DE CONTROLO DA EXTENSÃO	24
Figura 12 – COMANDO SQL INSERT	25
Figura 13 – INSTANCIÇÃO SQLCOMMAND	25
Figura 14 – PARAMETROS DEFINIDOS NO SQLCOMMAND	25
Figura 15 – RESPOSTA POSITIVA OU NEGATIVA	25
Figura 16 – CONEXÃO À BASE DE DADOS	26
Figura 17 – CONTROLO TRY OU CATCH	26
Figura 18 – FECHO DE CONEXÃO	26
Figura 19 – GRID 1	27
Figura 20 - BOUNDFIELD	27
Figura 21 - IMAGEFIELD	27
Figura 22 – LIBRARY 2	28
Figura 23 - DATATABLE	28
Figura 24 - CONNECTION STRING	28
Figura 25 - COMANDO SELECT	29
Figura 26 - SQLCOMMAND	29
Figura 27 - SQL CONNECTION STRING	29
Figura 28 - SQL DATA ADAPTER	29
Figura 29 - COMMAND TYPE	29
Figura 30 - COMANDOS INJETADOS NA BASE DE DADOS	29
Figura 31 - CONTROLO TRY	29
Figura 32 - SELECT COMMAND	30
Figura 33 - PREENCHER A TABELA COM DADOS	30
Figura 34 - ID GRIDVIEW 1	30
Figura 35 - DADOS DA TABELA	30
Figura 36 - CATCH	30
Figura 37 - FECHAR A CONEXÃO	31
Figura 38 - LINKS	31
Figura 39 - DESCRIÇÃO DA EMPRESA	31
Figura 40 - CONTACTOS	31
Figura 41 - LIBRARY 3	32
Figura 42 - EXECUTAR A LIBRARY	32
Figura 43 - PAGE LOAD	32
Figura 44 - ENCOMENDA WEBSITE	33
Figura 45 - LIBRARY 4	33
Figura 46 - CONNECTION STRING	33
Figura 47 - COMANDO INSERT	34
Figura 48 - MASTERPAGE	34
Figura 49 - THEME	35
Figura 50 - ACESSO AO CONTEXTO DA APLICAÇÃO	35
Figura 51 - INSTANCIÇÃO DE ROLESTORE	35

Figura 52 - INSTANCIÇÃO DE ROLEMANAGER -----	36
Figura 53 - CRIAÇÃO DE ROLE -----	36
Figura 54 - CRIAÇÃO DO UTILIZADOR PARA A ROLE CAN EDIT -----	36
Figura 55 - ORGANIZAÇÃO DA PASTA ADMIN -----	36
Figura 56 - CONFIGURAÇÃO DA PASTA ADMIN -----	37
Figura 57 - TORNAR A PÁGINA ADMIN VISIVIEL AOS UTILIZADORES DA ROLE CAN EDIT-----	37
Figura 58 - género dos inquiridos -----	39
Figura 59 - IDADE DOS INQUIRIDOS-----	39
Figura 60 - DESIGN DA APLICAÇÃO -----	40
Figura 61 - NAVEGAÇÃO DA APLICAÇÃO-----	40
Figura 62 - GESTÃO DE PEDIDOS -----	41
Figura 63 - INFORMAÇÃO PERTINENTE-----	41
Figura 64 - NAVEGAÇÃO PELAS PÁGINAS-----	42
Figura 65 - SERVIÇOS PRESTADOS PELO WEBSITE-----	42

INTRODUÇÃO

O presente trabalho é realizado no âmbito da unidade curricular anual Projeto Global da Licenciatura de Informática lecionado pelo Instituto Superior de Tecnologias Avançadas de Lisboa. O trabalho tem um conteúdo de relatório Técnico-Científico, sendo que a área científica ostentada tem como base as Ciências da Programação.

Com este trabalho é pretendido o desenvolvimento de uma aplicação na área da restauração, de modo a disponibilizar um ambiente de acesso de interação entre clientes e a empresa em questão. É ainda disponibilizada uma plataforma onde apenas o administrador é capaz de gerir os produtos.

Os objetivos específicos desta aplicação são a existência de um ambiente que permite à empresa uma maior proximidade com os clientes; disponibilizar ao cliente uma aplicação que permite um fácil e rápido pedido de uma refeição e permitir que a organização em causa consiga efetuar a consulta desses mesmos pedidos.

A escolha do tema para o presente trabalho surgiu devido à existência da necessidade de colmatar a falta de uma plataforma que vai ao encontro aos padrões de exigência que a empresa de restauração em questão colocou, de modo a contrapor as necessidades dos seus clientes.

A metodologia deste projeto compreende uma pesquisa bibliográfica sobre os temas da unidade científica em questão como *ASP.NET*, Gestão de Produtos, Gestão de Encomendas e Engenharia de *Software*. Esta pesquisa vai ser apresentada no capítulo referente ao Estado de Arte e, posteriormente apresentado o levantamento de necessidades por parte da empresa de restauração como por exemplo o *layout* pretendido.

O desenvolvimento deste trabalho teve como base a ferramenta *Microsoft Visual Studio*, ferramenta essa que foi base importante ao longo da Licenciatura. Este *Software* de desenvolvimento permite a construção da aplicação e ainda de forma prática, a inclusão de uma base de dados em *SQL Server*¹ suportando a construção da nossa aplicação.

¹ SQL Server - <http://www.microsoft.com/pt-pt/server-cloud/products/sql-server/>

Terminadas estas etapas procedeu-se à realização de um teste de usabilidade junto de possíveis clientes de forma a entender se a nossa aplicação vai de encontro às necessidades exigidas, cumprindo assim os objetivos a que este trabalho se propôs.

O trabalho está constituído em três etapas fundamentais. A primeira fase diz respeito à pesquisa bibliográfica responsável pelo Estado de Arte. Neste capítulo é descrito de forma precisa os conceitos que estão presentes ao longo do trabalho, como por exemplo *Website*, Engenharia de *Software*, *ASP.NET*, *SQL Server* e Gestão de Encomendas.

A segunda fase corresponde ao desenvolvimento do trabalho. Nesse capítulo serão apresentadas as ferramentas e o desenvolvimento do código fundamental para a conclusão do projeto prático.

A última fase corresponde ao teste de usabilidade que é imprescindível no desenvolvimento de uma aplicação, porque permite obter um *feedback* por parte de potenciais utilizadores da aplicação e também ter uma ideia do que se podem aprofundar ou aperfeiçoar no trabalho prático em questão, para que possa satisfazer a empresa de restauração e os seus clientes.

Na parte final deste trabalho, irá ser feita uma conclusão tendo como base a reflexão da execução do mesmo, e irá ser verificado se os objetivos propostos foram atingidos, e se as complicações que foram surgindo foram resolvidas.

ESTADO DE ARTE

Os avanços tecnológicos têm vindo a acompanhar a evolução do Ser Humano e para ir ao encontro das suas necessidades são diariamente criadas ferramentas para facilitar, tanto o dia-a-dia de pessoas ditas comuns como para profissionais da área da Informática. Hoje em dia, as Organizações estão cada vez mais dependentes da Informação Digital, cujo principal objetivo é gerir, divulgar ou manter o funcionamento das mesmas.

O trabalho apresentado tem como tema principal a conceção e desenvolvimento de uma aplicação em *ASP.NET* e por essa razão, esta parte do trabalho tem como objetivo pôr em evidência a criação de uma aplicação em *ASP.NET*. O presente capítulo vai evidenciar os elementos que são necessários para a realização desta tarefa e também definir conceitos relacionados com o tema do trabalho como por exemplo: Gestão de Produtos, Gestão de Encomendas, *ASP.NET* e Engenharia de *Software*.

Como já foi referido, um dos conceitos a ser abordado é a Gestão de Produtos. Mcnaughtan (2013) define Gestão de Produtos como uma disciplina dinâmica que precisa de uma combinação de visão de negócios, conhecimento tecnológico e orientação para o cliente. Segundo Yoskovitz (2009) existem fases na Gestão de Produtos que ajudam as Organizações a gerir diversos produtos e tomar decisões. Existe ainda quem acredita que uma boa Gestão de Produtos só pode ser sucesso se por detrás dela existir um bom Gestor de Produtos e Tiedman (2013), destacou seis características que marcam um bom Gestor que são: atitude, conhecimento, comunicação, clientes, gestão e poder de decisão. Relativamente à atitude Tiedman (2013) diz que esta é a característica que mais deve sobressair num Gestor de Produtos porque é ela que impulsiona as pessoas a alcançarem os seus objetivos ou a melhorar o seu desempenho, visto que vivemos numa sociedade bastante competitiva. A segunda característica é o conhecimento porque é a base para os outros atributos. A terceira característica é a comunicação e a capacidade de persuadir os outros porque um Gestor de Produtos tem de ser soberbo, ouvinte ativo e proactivo porque interage e estabelece um relacionamento com pessoas de diversas origens, classes sociais e cargos. A quarta característica é o cliente. O autor diz que os clientes são o ponto fulcral na Gestão de Produtos porque é devido a eles que existe ou não bons resultados dependendo da sua satisfação face a um determinado produto. A seguinte característica trata-se do gerenciamento que é uma característica que todos os gestores devem deter devido à obtenção de bons dados e quando não se obtém bons resultados, é necessário ter força de vontade para alcançar resultados positivos. Para isto, um Gestor de Produtos deve estabelecer metas apropriadas

aos seus objetivos e saber antecipar problemas e questões. A última característica que Tiedman (2013) refere, é o poder de decisão e é chamada como a essência de um Gestor de Produtos devido aos Gestores de Produtos enfrentarem uma progressão indeterminável de decisões que vão desde a compreensão de um problema a ser resolvido até ao desenvolvimento de estratégias para a resolução de problemas. Concluindo, a Gestão de Produtos é um mecanismo exigente porque requer liderança, organização e bons resultados. Mas no entanto, a obtenção de bons resultados traz consigo o aumento do sucesso da Organização assim como a satisfação de clientes.

Outro dos conceitos a ser tratado no presente trabalho é a Gestão de Encomendas e é este conceito que será discriminado em seguida. Para Menken (2007), os objetivos da Gestão de Encomendas são a oferta de um canal entre a empresa e o cliente onde o último pode requisitar e receber serviços, o fornecimento de informação a clientes e a possibilidade do cliente puder ser assistido. Bluemner (2013) põe em destaque o Software que Gestão de Encomendas tem, assim como as suas principais características como a criação de uma ordem que permite identificar o trabalho que deve ser feito.

Barry (2014) afirma que existem seis fases que podem ajudar uma empresa com a Gestão de Encomendas. A primeira fase, “*order management system assessment*”, melhora a informação da base de dados, da análise de *Marketing* e *Merchandasing* e ferramentas de BI. A segunda fase (“requisitos de documentação”) tem como objetivo os requisitos de negócios, a elaboração de relatórios e a base de dados de Marketing. A terceira fase ou a chamada “fase da avaliação de *Order Management System*” é aquela que avalia e completa cada sistema de Gestão de Encomendas e resolve os problemas que possam vir a existir. A quarta fase diz respeito à tomada de decisão através da revisão de custos iniciais. Nesta fase vai existir uma revisão de elementos com a finalidade de determinar os custos finais com um fornecedor. Na quinta fase vai existir uma negociação de contractos com o cliente onde este analisa o sistema de Gestão de Encomendas. Na última fase vai-se executar decisões que devem ser cumpridas e também é a fase que monitoriza o desenvolvimento do projeto.

No que toca à presença de Gestão de Encomendas em empresas de comércio, Panella (2014), diz que fornece uma nova alternativa de atendimento aos seus clientes sendo um dos componentes chave.

Para finalizar o conceito de Gestão de Encomendas, Burnson (2014) destaca dez Softwares que são considerados os que mais se distinguem neste conceito que são: *Fishbowl*, *Uroute*, *SNAP*, *RAMP*, *3PL Central*, *JDA*, *made4net*, *PathGuide*, *ViewPoint* e *Shippers Edge*. Embora estes dez Softwares tenham funções diferentes e se dirigem para diferentes clientes,

todos eles conseguem aumentar a eficiência do armazém com o seu conjunto de aplicativos, incluindo o atendimento de pedidos através de um rede *online*.

Como o presente trabalho tem como principal objetivo o desenvolvimento de um *Website*, é de extrema importância definir o conceito de *Website*.

Uma possível definição de *Website* é, segundo Griffith (2013), “um conjunto de páginas relacionadas e preenchidas com informação, anúncios e comércio eletrônico. Todos relacionados com um nome e domínio em comum.” Para o mesmo autor, o desenvolvimento de um *Website* pode ser complexo e poderá ser necessário contratar um profissional ou então executarmos nós próprios: “ Se for necessário um *Website* mais específico, poderá ter que recorrer à ajuda de profissionais, ou então pode tentar codificá-lo você mesmo.”

Depois de uma pequena introdução ao conceito de *Website*, é relevante para o presente trabalho, a introdução do conceito de Engenharia de *Software*. Este conceito revela-se de grande importância, uma vez que são descritas todas as etapas necessárias, até à implementação de um *Software*.

O desenvolvimento de um *Software* é representado por um conjunto de etapas que ocorrem desde a sua idealização, levantamento de requisitos, desenvolvimento, testes de usabilidade e implementação da aplicação.

Segundo Vernon (2008), a Engenharia de *Software* é o ramo da engenharia de sistemas que se foca no desenvolvimento de *Software* de grandes dimensões. É importante o Engenheiro de *Software* focar-se nos objetivos específicos do *Software* e aliá-los aos serviços e especificações requeridos pelo mundo real.

Para (Reading 2007), por vezes não é dada a devida importância à engenharia de *Software*, uma vez que este tipo de atividade não é apenas programação.

De acordo com o Institute of Electrical and Electronics Engineers (IEEE) (2006), um possível conceito para Engenharia de *Software* é o estudo e a aplicação de engenharia ao *Design*, desenvolvimento e manutenção de um *Software*. Existem vários modelos que permitem uma diversidade de escolha, uma vez que é possível utilizar um modelo mais adequado de acordo com a especificidade de cada projeto.

Segundo Munassar & Govardhan (2010), um modelo de desenvolvimento de *Software* é uma representação abstrata de um processo. Este mesmo processo pode ter diferentes pontos de vista, nomeadamente, no que diz respeito ao levantamento de requisitos, *design* e validação de *Software*. Alguns dos modelos de desenvolvimento de *Software* mais utilizados são o modelo em Cascata, o modelo de Protótipo, o modelo de Iteração, o modelo em Espiral e o Modelo de Desenvolvimento Rápido de Aplicações.

É possível então afirmar que a Engenharia de *Software* é multifacetada e é possível ser distinguida em três dimensões. Sustentando esta afirmação, Vernon (2008) indica que a primeira dimensão corresponde às ferramentas, técnicas e métodos necessários para o correto desenvolvimento de um *Software*. A segunda dimensão corresponde às técnicas de gestão que são necessárias para executar projetos de *Software* e aplicar boas práticas de desenvolvimento. Na terceira e última dimensão é representado se os atributos funcionais do *Software* foram alcançados, discriminando a segurança, portabilidade e fiabilidade.

Vernon (2007) reitera ainda que existe a regra dos três P's dentro da Engenharia de *Software*. Os mesmos são: Pessoas, Produtos e Processos. “As pessoas são o principal e mais importante componente, uma vez que são elas que desenvolvem, projetam e vão de encontro às funcionalidades do sistema. Relativamente aos produtos, o autor defende que existem dois tipos: os produtos genéricos e os produtos personalizados. Os primeiros têm como mercado alvo o mercado aberto, a quem o quiser comprar. Já os produtos personalizados destinam-se a clientes e organizações específicas.

O mesmo autor defende e explica que para além das dimensões anteriormente mencionadas, existem ainda “três componentes que são conhecidos como os três P's:

Pessoas, Produtos e Processos.” As pessoas são o principal componente neste tipo de engenharia porque “são um aspeto muito importante nos sistemas de Engenharia de *Software*. São elas que usam o sistema que está a ser desenvolvido, que projetam e constroem o sistema.” Relativamente ao produto, existem dois tipos, que são os produtos genéricos “(...) sistemas autónomos que são produzidos por um desenvolvimento de organização e vendido no mercado de modo aberto, a quem o quiser comprar.” e os produtos personalizados: “(...) são encomendados por um cliente específico e desenvolvidos especialmente para alguma empresa como forma a ir ao encontro de uma necessidade específica.”

Relativamente aos processos de *Software*, Vernon (2007) diz que se trata de um conjunto estruturado de atividades necessárias para desenvolver um sistema de *Software*.

Seguidamente vai ser abordado o conceito de *ASP.NET*, visto que foi a partir desta linguagem de programação que se desenvolveu o projeto.

Segundo Liberty (2002), *ASP.NET* pode ser considerada a ferramenta de maior sucesso para a construção de aplicações *Web Server-Side*, pois permite uma grande variedade de decisões, como por exemplo desenvolver aplicações orientadas para dados dinâmicos.

Para McPeak (2011), *ASP.NET* não deve ser utilizado para *Websites* de menor dimensão ou *Websites* pessoais. O autor afirma que a intenção da *Microsoft* é desenvolver uma plataforma cuja sua principal componente é a simplicidade. Devido a esse facto, a

Microsoft lançou a *WebMatrix*. Esta plataforma permite desenvolver um *Website* de forma mais simples e “*user-friendly*”, continuando a codificar o projeto em *ASP.NET*.

De acordo com Gladwin (2001), a utilização de *ASP* e de *JSP* são duas das tecnologias mais populares para gerar conteúdos dinâmicos num *Website*. A principal diferença entre estas duas tecnologias é o facto de que um conteúdo programado em *ASP* geralmente interage com um *Back-End* de ambiente construído com tecnologias *Microsoft*. Noutro espectro, *JSP* é equacionado para funcionar em ambiente baseado em *Java*.

CONTEXTUALIZAÇÃO

De acordo com um artigo de Silva (2014) é de salientar, na nossa sociedade, o crescimento cada vez maior da utilização da *Internet* e do uso de plataformas em que é possível encomendar um determinado produto virtualmente. Para ir de encontro a este fenómeno, algumas cadeias de Hipermercados investem cada vez mais em entregas de encomendas *online*.

Entidades comerciais como o Continente ou o Jumbo foram alvos deste artigo aonde se conclui que o Continente tem meio milhão de utilizadores da sua plataforma desde 2001, enquanto o Jumbo, conta com 250 mil registos desde 2007.

Tendo em conta este artigo, os Portugueses fazem mais encomendas em produtos alimentares mas tem havido uma crescente procura em produtos de bazar. Os nossos habitantes vão cada vez mais comprar *online*, o que vai fazer com que as encomendas *online* cresçam, fazendo com que as entidades invistam mais em suportes informáticos. Os utilizadores deste tipo de plataformas inserem-se em idades compreendidas entre os 25 e os 45 anos, incidindo mais em indivíduos do sexo feminino.

Embora haja quem ache que as plataformas de encomendas *online* sejam a melhor forma de efetuar compras, há quem ache que este método não é o mais acertado pois exige complexidade para os retalhistas, mas existe também a dúvida da qualidade dos produtos encomendados, especialmente de alimentos perecíveis.

DESENVOLVIMENTO

Nesta fase do trabalho são observados os procedimentos constituintes do desenvolvimento e da gestão da aplicação, desde o levantamento de necessidades até aos testes de usabilidade. Vai ainda ser apresentado alguns conceitos e informações necessários à resolução do projeto.

A ideia de desenvolver uma aplicação surgiu na sequência de uma conversa com um conhecido de uma empresa de restauração (n.d.r Restaurante Lucimar), em Lisboa, com o interesse de desenvolver uma aplicação de gestão de encomendas para o seu restaurante.

Como aluno finalista na Licenciatura em Informática, aceitei prontamente o desafio e adotei o mesmo ao desenvolvimento do Projeto Global.

Para a execução do Projeto, foram várias as fases necessárias até à sua conclusão. Os mesmos vão ser representados em seguida, por ordem cronológica.



FIGURA 1 – ETAPAS DE DESENVOLVIMENTO

Numa primeira fase foi necessário abordar o proprietário do restaurante em causa com o intuito de se fazer o levantamento de requisitos que eram necessários estar presentes na aplicação.

Seguidamente foi necessário decidir quais as ferramentas a utilizar na execução da aplicação. Nesta fase, e após fazer um enquadramento do Projeto com o Programa Curricular, decidi que o ambiente de desenvolvimento a utilizar é o *Microsoft Visual Studio 2013*. As linguagens de programação a utilizadas são: *C#*, *ASP.NET* e o *Transact Sql*, conhecido como *T-SQL*, uma vez que as tecnologias são suportadas pelo ambiente de desenvolvimento

utilizado.

A elaboração desta aplicação tem como objetivos práticos a implementação de uma plataforma que permite à empresa uma maior visibilidade junto dos clientes e também promover a funcionalidade de encomendar uma refeição através desta plataforma.

DESENVOLVIMENTO EMPÍRICO DO WEBSITE

Convém salientar que todas as figuras presentes no trabalho são da autoria do aluno. Dado por terminada esta primeira, irá ter início o desenvolvimento empírico do *Website*.

Para a realização deste *Website*, vão existir arquivos que contêm duas extensões diferentes, o *aspx* e *.aspx.cs*. As extensões *aspx* contêm as ferramentas do *asp.net*, como as caixas de texto e os botões. Já as extensões *aspx.cs* contem o código (C#) que define o comportamento dessas mesmas ferramentas.

Depois de aberta a aplicação *Microsoft Visual Studio 2013* procedeu-se à criação de um projeto que já vem definido com as ferramentas necessárias à criação da aplicação, isto é, já traz uma base de dados e vários ficheiros *aspx* e *aspx.cs*. A primeira fase foi criar uma tabela com o objetivo de inserir o menu que o restaurante tem para oferecer, e essa tabela tem a seguinte estrutura:

```
CREATE TABLE [dbo].[testeImage] (  
    [Id] INT IDENTITY (1, 1) NOT NULL,  
    [Name] VARCHAR (50) NULL,  
    [ContentType] VARCHAR (50) NULL,  
    [Data] VARBINARY (MAX) NULL,  
    [descricao] VARCHAR (50) NULL,  
    [preco] VARCHAR (50) NULL,  
    PRIMARY KEY CLUSTERED ([Id] ASC)  
);
```

FIGURA 2 – CRIAÇÃO DE UMA TABELA

Nesta tabela, os campos “*ContentType*” e “*Data*” são relativos à inserção de imagens, sendo que o “*ContentType*” contém este tipo de dados para permitir a utilização de imagens em formato *.jpeg*, *.gif* ou *.png*. O campo “*Data*” contém este tipo de dados para que os dados relativos à imagem sejam convertidos em binário. Relativamente aos campos “*descrição*” e “*preço*” são relativos à descrição da imagem e ao preço dos produtos a vender pelo restaurante. Como se pode verificar, o “*id*” (número de identificação do produto), vai incrementando-se em 1 consoante o número de imagens inseridas na nossa base de dados, referente ao Menu e para isso utilizamos a propriedade *IDENTITY(1,1)* do *transact-sql*.

²[https://msdn.microsoft.com/en-us/library/system.net.mime.contenttype\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/system.net.mime.contenttype(v=vs.110).aspx)

```

<connectionStrings>
  <add name="DefaultConnection" connectionString="Data Source=(LocalDb)\v11.0;AttachDbFilename=|DataDirectory|\aspnet-pjgb-20150916054014.mdf;
    Initial Catalog=aspnet-pjgb-20150916054014;Integrated Security=True" providerName="System.Data.SqlClient"/>
</connectionStrings>

```

FIGURA 3 – CONEXÃO À BASE DE DADOS

Esta instrução é definida no *Web.config*, que é o principal ficheiro de configuração de um *Website*, e vai permitir que a base de dados criada anteriormente se conecte ao *Website*. É utilizada a propriedade “*DefaultConnection*” para que mais tarde seja possível instanciar a conexão à base de dados noutros ficheiros de código necessários.

Em seguida, será necessário criar uma tabela para gerir os pedidos efetuados pelos clientes.

```

CREATE TABLE [dbo].[pedido] (
  [Id]          INT          IDENTITY (1, 1) NOT NULL,
  [Descricao]  VARCHAR (50) NULL,
  [email]      VARCHAR (50) NULL,
  PRIMARY KEY CLUSTERED ([Id] ASC)
);

```

FIGURA 4 – CRIAÇÃO DE UMA TABELA

Na figura acima representada, é criada uma tabela com o nome *pedido*, onde fica registado a descrição da encomenda e o correio eletrónico do cliente para potenciais esclarecimentos e confirmação do pedido.

```

<asp:FileUpload ID="FileUpload1" runat="server" />
<asp:Button ID="btnUpload" runat="server" Text="Upload"
  OnClick="btnUpload_Click" />

```

FIGURA 5 - FILEUPLOAD

No ficheiro *admin.aspx* foi introduzido o controlo *FileUpload*³ que vai permitir ao administrador fazer o carregamento de uma refeição no menu com as seguintes propriedades: imagem, descrição e preço. Por sua vez o botão denominado de “*btnUpload*” que é configurado no *admin.aspx.cs*, tem como objetivo inserir o produto na base de dados.

³ [https://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.fileupload\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.fileupload(v=vs.110).aspx)

```

<asp:Label ID="Label1" runat="server" Text="Descrição"></asp:Label>

<asp:TextBox ID="descricao" runat="server"></asp:TextBox>

<asp:Label ID="Label2" runat="server" Text="Preço"></asp:Label>

<asp:TextBox ID="preco" runat="server"></asp:TextBox>

<asp:TextBox ID="TextBox1" runat="server"></asp:TextBox>
<asp:TextBox ID="TextBox2" runat="server"></asp:TextBox>

```

FIGURA 6 – FORMULÁRIO A INTRODUIR NA BASE DE DADOS

Este código referente ao ficheiro `admin.aspx`, permite que as informações inseridas na base de dados referentes ao preço e descrição, sejam preenchidas de acordo com estas *textboxes*.

```

<asp:Label ID="lblMessage" runat="server" Text=""
Font-Names = "Arial"></asp:Label>

```

FIGURA 7 – INSERÇÃO NA BASE DE DADOS

O código acima mencionado é referente ao ficheiro `admin.aspx`, e permite que seja exibida uma mensagem a indicar se o produto inserido na base de dados foi executado com sucesso.

Após ser criada a tabela da base de dados a partir do *Web.config*, irá ser introduzido de seguida o processo referente à configuração da inserção dos produtos na base de dados no ficheiro *admin.aspx.cs*.

```

using System;
using System.IO;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Data.Sql;
using System.Data.SqlClient;

```

FIGURA 8 – LIBRARY 1

Tal como se pode verificar na figura acima mencionada, esta irá ser a *Library*⁴ utilizada para ser possível utilização de classes referentes à inserção de imagens na base de dados.

⁴ [https://msdn.microsoft.com/en-us/library/d11h6832\(v=vs.71\).aspx](https://msdn.microsoft.com/en-us/library/d11h6832(v=vs.71).aspx)

```

string filePath = FileUpload1.PostedFile.FileName;
string filename = Path.GetFileName(filePath);
string ext = Path.GetExtension(filename);
string contenttype = String.Empty;

```

FIGURA 9 - CONTROLO

As seguintes linhas de código reconhecem o caminho, nome e extensão do ficheiro, e converte para um *Array*⁵ do tipo *Byte*. Na primeira linha de código é identificado o caminho físico do ficheiro; Na segunda linha de código é identificado o nome do ficheiro; Na terceira linha de código é identificada a extensão do ficheiro; Na quarta linha de código é feito um controlo para saber se é apresentada alguma das extensões suportadas.

```

switch (ext)
{
    case ".doc":
        contenttype = "application/vnd.ms-word";
        break;
    case ".docx":
        contenttype = "application/vnd.ms-word";
        break;
    case ".xls":
        contenttype = "application/vnd.ms-excel";
        break;
    case ".xlsx":
        contenttype = "application/vnd.ms-excel";
        break;
    case ".jpg":
        contenttype = "image/jpeg";
        break;
    case ".png":
        contenttype = "image/png";
        break;
    case ".gif":
        contenttype = "image/gif";
        break;
    case ".pdf":
        contenttype = "application/pdf";
        break;
}

```

FIGURA 10 – LEITURA DA EXTENSÃO DE FICHEIROS

Nesta condição é identificada a extensão do ficheiro pelo *file upload* e definido o *ContentType* da base de dados.

```

}
if (contenttype != String.Empty)
{
    Stream fs = FileUpload1.PostedFile.InputStream;
    BinaryReader br = new BinaryReader(fs);
    Byte[] bytes = br.ReadBytes((Int32)fs.Length);
}

```

FIGURA 11 – CONDIÇÃO DE CONTROLO DA EXTENSÃO

Na imagem acima ilustrada, é efetuada uma condição de controlo para saber se o *ContentType* é preenchido com uma das extensões definidas na condição anteriormente descrita na figura 10.

⁵[https://msdn.microsoft.com/en-us/library/aa288453\(v=vs.71\).aspx](https://msdn.microsoft.com/en-us/library/aa288453(v=vs.71).aspx)

```
string strQuery = "insert into testeImage(Name, ContentType, Data,descricao,preco)" +
    " values (@Name, @ContentType, @Data , @descricao , @preco)";
```

FIGURA 12 – COMANDO SQL INSERT

Neste exemplo, o comando *insert* é uma função de inserção de dados. Neste caso em específico irá inserir a descrição, *ContentType* e preço na tabela *testeImage*.

```
SqlCommand cmd = new SqlCommand(strQuery);
```

FIGURA 13 – INSTANCIACÃO SQLCOMMAND

Definição da variável “*cmd*” que recebe o objecto *SqlCommand*⁶. Por sua vez este recebe como parâmetro o comando da base de dados presente na aplicação. Os vários tipos de comando podem ser o *insert*, *delete*⁷ ou *update*⁸.

```
cmd.Parameters.Add("@Name", SqlDbType.VarChar).Value = filename;
cmd.Parameters.Add("@ContentType", SqlDbType.VarChar).Value
    = contenttype;
cmd.Parameters.Add("@Data", SqlDbType.Binary).Value = bytes;
cmd.Parameters.Add("@descricao", SqlDbType.VarChar).Value
    = descricao.Text;
cmd.Parameters.Add("@preco", SqlDbType.Money).Value
    = preco.Text;
InsertUpdateData(cmd);
```

FIGURA 14 – PARAMETROS DEFINIDOS NO SQLCOMMAND

Uma vez que a variável *cmd* foi definida como um *SqlCommand*, irá ser obrigatória a inserção de uma imagem, para que seja também lida as *textboxes*⁹ correspondentes a essa mesma imagem. A última linha de código presente na imagem, chama uma função responsável por abrir a conexão ao *SQLServer* com o objetivo de inserir a informação.

```
lblMessage.ForeColor = System.Drawing.Color.Green;
lblMessage.Text = "File Uploaded Successfully";
}
else
{
    lblMessage.ForeColor = System.Drawing.Color.Red;
    lblMessage.Text = "File format not recognised." +
        " Upload Image/Word/PDF/Excel formats";
}
```

FIGURA 15 – RESPOSTA POSITIVA OU NEGATIVA

⁶[https://msdn.microsoft.com/en-us/library/system.data.sqlclient.sqlcommand\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/system.data.sqlclient.sqlcommand(v=vs.110).aspx)

⁷[https://msdn.microsoft.com/pt-br/library/ms189835\(v=sql.120\).aspx](https://msdn.microsoft.com/pt-br/library/ms189835(v=sql.120).aspx)

⁸<https://msdn.microsoft.com/en-us/library/ms177523.aspx>

⁹[https://msdn.microsoft.com/en-us/library/system.windows.controls.textbox\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/system.windows.controls.textbox(v=vs.110).aspx)

Após ser possível o carregamento da imagem através do objeto do *ASP.NET*, o *fileupload*, a instrução acima permite devolver uma resposta positiva caso a imagem for carregada com sucesso, ou uma resposta negativa caso o ficheiro carregado não seja do formato definido na figura 10.

```
private Boolean InsertUpdateData(SqlCommand cmd)
{
    String strConnString = System.Configuration.ConfigurationManager
        .ConnectionStrings["DefaultConnection"].ConnectionString;
    SqlConnection con = new SqlConnection(strConnString);
    cmd.CommandType = CommandType.Text;
    cmd.Connection = con;
}
```

FIGURA 16 – CONEXÃO À BASE DE DADOS

É estabelecida uma conexão ao *SQLServer*, através da *ConnectionString*, definida no *WebConfig* que se chama “DefaultConnection”.

```
try
{
    con.Open();
    cmd.ExecuteNonQuery();
    return true;
}
catch (Exception ex)
{
    Response.Write(ex.Message);
    return false;
}
```

FIGURA 17 – CONTROLO TRY OU CATCH

Se a aplicação executar com sucesso a ligação à base de dados, é executado o comando definido no objeto “*cmd*”. Caso contrário, é devolvida uma mensagem de erro.

```
finally
{
    con.Close();
    con.Dispose();
}
}
```

FIGURA 18 – FECHO DE CONEXÃO

No código referente a esta figura, depois de ser realizado as instruções de código presentes no “*try*” ou presentes no “*catch*”, é executado o “*finally*” onde está definido que a conexão irá ser fechada.

Seguidamente irá ser explicado detalhadamente todo o código referente ao default.aspx onde além de outras informações, é apresentado aos clientes as refeições disponíveis para encomenda.

```
<asp:GridView GridLines="Horizontal" ID="GridView1" runat="server" AutoGenerateColumns = "False"
Font-Names = "Arial" Height="560px" Width="823px" HorizontalAlign="Center" BackColor="White"
BorderColor="#CCCCCC" BorderStyle="None" BorderWidth="1px" CellPadding="4" CellSpacing="5" ForeColor="Black">
<Columns>
```

FIGURA 19 – GRID 1

No código acima mencionado, os dados são mostrados numa *grid*¹⁰. É ainda configurado que cada informação de um produto irá ter uma coluna própria.

```
<asp:BoundField DataField = "descricao" ItemStyle-HorizontalAlign="Center" HeaderText="" FooterStyle-HorizontalAlign="Center">
<ItemStyle HorizontalAlign="Center"></ItemStyle>
</asp:BoundField>
```

FIGURA 20 - BOUNDFIELD

O código acima apresentado, indica que a descrição é apresentada numa *BoundField*¹¹, ou seja, é mostrado o texto onde é efetuada a ligação à base de dados (*DataBinding*¹²).

```
<asp:ImageField DataImageUrlField = "ID"
DataImageUrlFormatString = "ImageCSharp.aspx?ImageID={0}"
ControlStyle-Width = "100" ControlStyle-Height = "100"
HeaderText = " " ItemStyle-HorizontalAlign="Center">
<ControlStyle Height="250px" Width="250px"></ControlStyle>
<ItemStyle HorizontalAlign="Center"></ItemStyle>
</asp:ImageField>
<asp:BoundField DataField = "preco" ItemStyle-HorizontalAlign="Center" >
<ItemStyle HorizontalAlign="Center"></ItemStyle>
</asp:BoundField>
</Columns>
<FooterStyle BackColor="#CCCC99" ForeColor="Black" />
<HeaderStyle BackColor="#333333" Font-Bold="True" ForeColor="White" />
<PagerStyle ForeColor="Black" HorizontalAlign="Center" BackColor="White" />
<RowStyle BackColor="#FFF7E7" ForeColor="Gray" />
<SelectedRowStyle BackColor="#CC3333" Font-Bold="True" ForeColor="White" />
<SortedAscendingCellStyle BackColor="#F7F7F7" />
<SortedAscendingHeaderStyle BackColor="#4B4B4B" />
<SortedDescendingCellStyle BackColor="#E5E5E5" />
<SortedDescendingHeaderStyle BackColor="#242121" />
</asp:GridView>
```

FIGURA 21 - IMAGEFIELD

¹⁰ <https://msdn.microsoft.com/en-us/library/aa479342.aspx>

¹¹ [https://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.boundfield\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/system.web.ui.webcontrols.boundfield(v=vs.110).aspx)

¹² [https://msdn.microsoft.com/en-us/library/bb546190\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/bb546190(v=vs.110).aspx)

Neste código, é apresentado o campo imagem num *Imagefield*. O *Imagefield* mostra a imagem no qual é efetuada a ligação de dados e através do URL de um *generichandler*¹³ vai ser possível seleccionar a imagem pretendida.

Seguidamente irá ser apresentado o código referente ao processo lógico de carregamento do *Website*, no ficheiro *menu.aspx.cs*.

```
using System;
using System.Collections.Generic;
using System.Data;
using System.Data.Sql;
using System.Data.SqlClient;
using System.IO;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
```

FIGURA 22 – LIBRARY 2

Esta vai ser a *library* presente no ficheiro *menu.aspx.cs*, e permite incorporar nesta página uma ligação à base de dados com a inclusão de *using System.Data* e os seus derivados nas seguintes linhas.

```
protected void Page_Load(object sender, EventArgs e)
{
    DataTable dt = new DataTable();
```

FIGURA 23 - DATATABLE

A imagem acima apresentada, indica o código referente à instância de um objeto *DataTable*¹⁴ como *dt*. A *DataTable* é uma tabela onde todos os dados referentes ao menu da aplicação ficam armazenados.

```
String strConnString = System.Configuration.ConfigurationManager.
    ConnectionStrings["DefaultConnection"].ConnectionString;
```

FIGURA 24 - CONNECTION STRING

Neste código é configurada uma *string*¹⁵ *strConnString*, sendo que a *DefaultConnection* é definida no *web.config*.

¹³ [https://msdn.microsoft.com/en-us/library/bb398986\(v=vs.140\).aspx](https://msdn.microsoft.com/en-us/library/bb398986(v=vs.140).aspx)

¹⁴ <https://datatables.net/development/server-side/sql>

¹⁵ [https://msdn.microsoft.com/en-us/library/system.string\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/system.string(v=vs.110).aspx)

```
string strQuery = "select ID, Name , descricao , preco from testeImage order by ID";
```

FIGURA 25 - COMANDO SELECT

O comando *select* permite selecionar o que é pretendido na tabela da nossa base de dados “*testeImage*”. Neste caso é pretendido o ID, *Name*, a descrição e o preço. Convém referir que as refeições vão ser ordenadas segundo o seu ID devido à cláusula “*order by ID*”.

```
SqlCommand cmd = new SqlCommand(strQuery);
```

FIGURA 26 - SQLCOMMAND

Neste excerto de código, à classe *SqlCommand* é atribuída à variável *cmd* para receber a *Query*.

```
SqlConnection con = new SqlConnection(strConnString);
```

FIGURA 27 - SQL CONNECTION STRING

Nesta linha de código, à classe *SqlConnection* é atribuída à variável *con* para receber a *ConnectionString* anteriormente definida.

```
SqlDataAdapter sda = new SqlDataAdapter();
```

FIGURA 28 - SQL DATA ADAPTER

Neste excerto de código, é definida a classe *SqlDataAdapter*, que é um conjunto de comandos.

```
cmd.CommandType = CommandType.Text;
```

FIGURA 29 - COMMAND TYPE

Aqui, a variável *cmd* tem um objecto *CommandType*¹⁶ que vai receber apenas texto.

```
cmd.Connection = con;
```

FIGURA 30 - COMANDOS INJETADOS NA BASE DE DADOS

O comando *SQL* vai ser injetado na conexão da base de dados.

```
try  
{  
    con.Open();
```

FIGURA 31 - CONTROLO TRY

¹⁶[https://msdn.microsoft.com/en-us/library/system.data.sqlclient.sqlcommand.commandtype\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/system.data.sqlclient.sqlcommand.commandtype(v=vs.110).aspx)

Esta imagem descreve uma condição *try*. A primeira linha de código apresentada vai permitir abrir a conexão da base de dados. Irão ser apresentadas as restantes linhas de código presentes neste controlo.

```
sda.SelectCommand = cmd;
```

FIGURA 32 - SELECT COMMAND

A segunda linha de código da condição *try*, permite que o *DataAdapter* receba o comando *Select* definido.

```
sda.Fill(dt);
```

FIGURA 33 - PREENCHER A TABELA COM DADOS

A terceira linha de código referente à condição *try* indica que o *DataAdapter* preencha a tabela definida com dados.

```
GridView1.DataSource = dt;
```

FIGURA 34 - ID GRIDVIEW 1

A quarta linha de código da condição *try* indica que a tabela definida no *datatable*, vai estar atribuída à *grid*, uma vez que já foi definido com o ID *GridView1*.

```
GridView1.DataBind();  
}
```

FIGURA 35 - DADOS DA TABELA

Na quinta e última linha de código pertencente à condição *try*, a *GridView1* preenche os dados da tabela utilizando o método *databinding*, que permite efetuar a ligação entre a tabela e a *gridview* definida.

```
catch (Exception ex)  
{  
    Response.Write(ex.Message);
```

FIGURA 36 - CATCH

Se a condição descrita anteriormente como *Try* não for executada, vai ser executado o código referente à condição *catch*¹⁷.

¹⁷<https://msdn.microsoft.com/pt-br/library/0yd65esw.aspx>

```

        finally
        {
            con.Close();
            sda.Dispose();
            con.Dispose();
            dt.Dispose();
        }
    }
}

```

FIGURA 37 - FECHAR A CONEXÃO

Após concluir as operações efetuadas anteriormente, a conexão entre a base de dados e a *grid* vai ser terminada.

Em seguida irá ser mostrado o código referente ao ficheiro `site.master.aspx` que apresenta a estrutura da aplicação.

```

<ul class="nav navbar-nav main-nav clear navbar-right ">
  <li><a runat="server" id="adminLink" visible="false"
href="~/admin/admin.aspx" class="color_animation">Admin</a></li>
  <li><a class="active color_animation" href="Default.aspx">WELCOME</a></li>
  <li><a class="color_animation" href="Encomendas.aspx">PEDIR</a></li>
  <li><a class="color_animation" href="#beer">MENU</a></li>

```

FIGURA 38 - LINKS

Nesta imagem está representado o código referente às hiperligações que cada página da aplicação vai ter. Como podemos verificar a página “*admin*” está inicialmente oculta através da declaração “*visible = false*”, mas mais tarde irá sofrer uma alteração devido à configuração de roles.

```

<div class="description_body margin-right">
  <h1>Sobre nós</h1>
  <div class="fa fa-cutlery fa-2x"></div>
  <p>Somos conhecidos pelas nossas francesinhas , visite-nos e dê nos a sua opinião .</p>
</div>

```

FIGURA 39 - DESCRIÇÃO DA EMPRESA

Na imagem acima é indicada a presença de uma classe, em que vai ser exibida a descrição da empresa de restauração.

```

<div class="or_call">
  <span class="social_heading">OR DIAL</span>
  <span class="social_info"><a class="color_animation" href="tel:217974689">217974689</a></span>
</div>

```

FIGURA 40 - CONTACTOS

Na imagem acima é fornecido ao cliente o número de contacto para o caso de existir alguma dúvida, ou para efetuar um esclarecimento.

Em seguida irá ser mostrado o código referente ao ficheiro *default.aspx.cs*.

```
using System;
using System.Collections.Generic;
using System.IO;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
```

FIGURA 41 - LIBRARY 3

Esta é a *library* presente no ficheiro *default.aspx.cs*, e que permite a ligação à base de dados anteriormente criada.

```
public partial class _Default : System.Web.UI.Page
{
```

FIGURA 42 - EXECUTAR A LIBRARY

Este código permite a leitura e execução da estrutura da *library* acima definida.

```
protected void Page_Load(object sender, EventArgs e)
{
}
```

FIGURA 43 - PAGE LOAD

A figura acima representada indica o processo lógico que permite ao mesmo tempo que a página está a ser carregada, as instruções de código vão sendo executadas.

Em seguida vai ser mostrado o código referente ao ficheiro *encomendas.aspx*.

```
<div class="description_body margin-right">
  <h1>Encomende aqui</h1>
  <div class="fa fa-cutlery fa-2x"></div>

  <div class="form">

    <label runat="server" >Prato </label> <asp:DropDownList ID="DropDownList1" runat="server"
      DataSourceID="SqlDataSource3" DataTextField="descricao"
      DataValueField="descricao"></asp:DropDownList>
    <asp:SqlDataSource ID="SqlDataSource3" runat="server"
      ConnectionString="<%= ConnectionStrings.DefaultConnection %>"
      SelectCommand="SELECT [descricao] FROM [testeImage]"></asp:SqlDataSource>
    <asp:SqlDataSource ID="SqlDataSource2" runat="server"
      ConnectionString="<%= ConnectionStrings.DefaultConnection %>"
      SelectCommand="SELECT [descricao] FROM [testeImage]"></asp:SqlDataSource>

    <asp:Label ID="Label1" runat="server" Text="email"></asp:Label>
    <asp:TextBox ID="email" runat="server"></asp:TextBox>
    <br />
    <br />
    <asp:Button ID="Button1" runat="server" Text="Encomendar" OnClick="Button1_Click" />
  </div>
</div>
```

FIGURA 44 - ENCOMENDA WEBSITE

No código apresentado na figura 44, vai ser definido o espaço onde se vai realizar a encomenda no *Website*, nos campos 'descrição' e 'Email'. O campo descrição é um controlo do tipo *dropdownlist* que vai executar um *select* à coluna descrição da tabela *testeImage*, para que o cliente possa só encomendar os produtos disponíveis. Em seguida, é definido um botão com o ID de 'Button1', que vai permitir executar esta operação.

Em seguida vai ser apresentado todo o código referente ao ficheiro *encomendas.aspx.cs*.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Web;
using System.Web.UI;
using System.Web.UI.WebControls;
using System.Data;
using System.Data.SqlClient;
using System.Data.Sql;
```

FIGURA 45 - LIBRARY 4

A imagem da figura 45 indica a *library* existente no ficheiro *encomendas.aspx.cs*.

```
String strConnString = System.Configuration.ConfigurationManager.
  ConnectionStrings["DefaultConnection"].ConnectionString;
```

FIGURA 46 - CONNECTION STRING

Estas duas linhas de código permitem iniciar uma nova ligação à base de dados.

```
string query = "INSERT INTO pedido (Descricao,email) VALUES ('" + DropDownList1.SelectedItem.Text + "', '" + email.Text + "')";  
SqlCommand command = new SqlCommand(query, connection);
```

FIGURA 47 - COMANDO INSERT

O código apresentado na figura 47, faz com que seja inserido na tabela os campos descrição lendo o valor selecionado pelo cliente numa *dropdownlist* e o endereço electrónico de quem faz a encomenda introduzido pelo cliente na *textbox*.

A primeira funcionalidade presente no *Website* que vai ser mencionada é a *Master Page*.

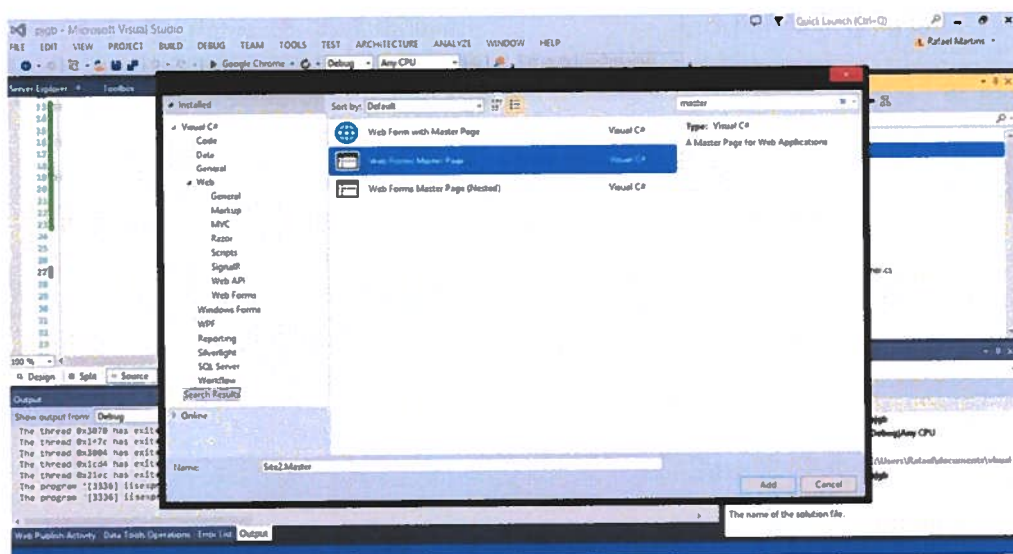


FIGURA 48 - MASTERPAGE

Uma *Master Page* é uma forma de construir um *layout*, uma estrutura, para todas as páginas *aspx* terem o mesmo formato, não sendo diferenciado umas das outras.

A segunda funcionalidade presente no *Website* que vai ser mencionada é o *Theme*.

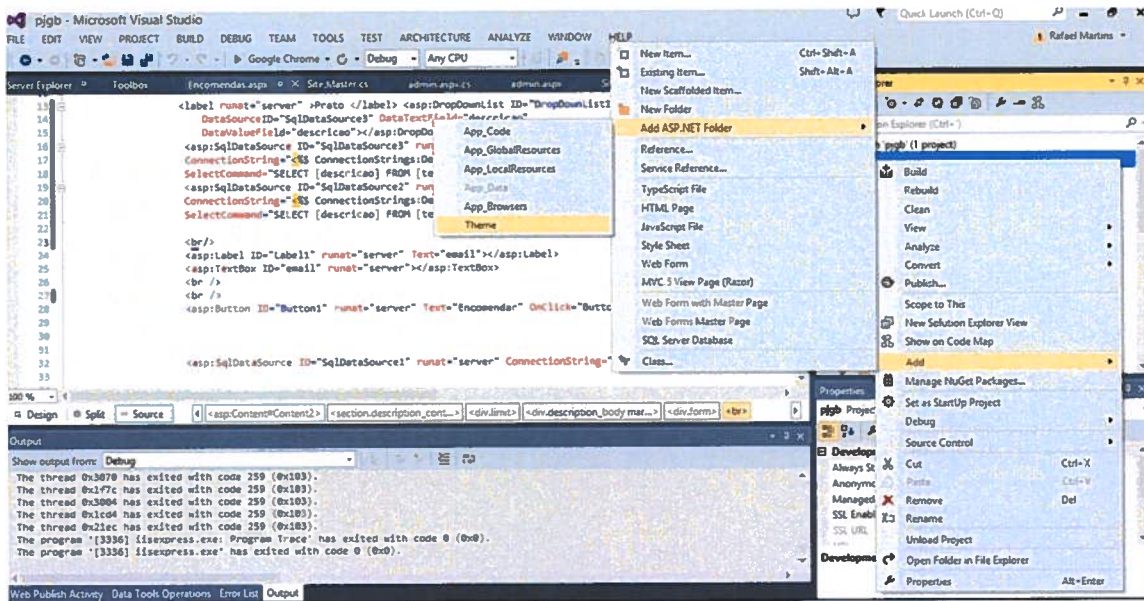


FIGURA 49 - THEME

O *Theme* fornece ao programador um mecanismo que permite definir o visual ao longo das nossas *webforms* através da existência de uma *Skin*, que por sua vez é ligada ao *Theme*, através de `theme="theme1"` para que todos os controlos de página possuam o aspeto visual a respeitar pela *Skin*.

Em seguida vai ser apresentada a configuração da página *admin.aspx*, onde só poderá ser acedido por um utilizador registado na página como sendo parte da role “*canEdit*”.

Para ser possível, teremos que criar uma classe na pasta “*Logic*” que terá o nome “*RoleAction*”. Esta classe irá definir um utilizador para uma role.

```
Models.ApplicationDbContext context = new ApplicationDbContext();
IdentityResult idRoleResult;
IdentityResult idUserResult;
```

FIGURA 50 - ACESSO AO CONTEXTO DA APLICAÇÃO

Começamos por instanciar um objeto *ApplicationDbContext*. Esta classe é responsável por administrar os objetos de identidade durante o tempo de execução. Nas seguintes linha de código instanciamos dois objetos *idRoleResult* e *idUserResult* que representam o resultado de uma operação de identidade.

```
var roleStore = new RoleStore<IdentityRole>(context);
```

FIGURA 51 - INSTANCIÇÃO DE ROLESTORE

De seguida instanciamos um objeto do tipo *RoleStore* para agrupar vários utilizadores na variável do tipo *role store* que se chama *roleStore*.

```
var roleMgr = new RoleManager<IdentityRole>(roleStore);
```

FIGURA 52 - INSTANCIÇÃO DE ROLEMANAGER

De seguida instanciamos o objeto do tipo *Role Manager* que é responsável pela gestão das *roles*, neste caso irá gerir a *role* definida anteriormente.

```
if (!roleMgr.RoleExists("canEdit"))
{
    IdRoleResult = roleMgr.Create(new IdentityRole { Name = "canEdit" });
}
```

FIGURA 53 - CRIAÇÃO DE ROLE

Após instanciarmos o *rolemanager* e o *rolestore*, iremos criar a *role* "canEdit". Cria-se uma condição de que se a *role* não existir, então é iniciado o processo de criação da *role*.

```
var userMgr = new UserManager<ApplicationUser>(new UserStore<ApplicationUser>(context));
var appUser = new ApplicationUser
{
    UserName = "rafa_martins@sapo.pt",
    Email = "rafa_martins@sapo.pt"
};
IdUserResult = userMgr.Create(appUser, "Martins24@");

// Se o utilizador da role edit for criado com sucesso
//adiciona o utilizador da canedit à can edit role
if (!userMgr.IsInRole(userMgr.FindByEmail("rafa_martins@sapo.pt").Id, "canEdit"))
{
    IdUserResult = userMgr.AddToRole(userMgr.FindByEmail("rafa_martins@sapo.pt").Id, "canEdit");
}
```

FIGURA 54 - CRIAÇÃO DO UTILIZADOR PARA A ROLE CAN EDIT

Neste processo criamos o utilizador utilizando a classe *ApplicationUser* que recebe dois parâmetros, o *userName* e o *e-mail* a que esse *userName* está associado. De seguida definimos a palavra passe para o utilizador criado na variável *appuser*. De seguida é testado a condição de que se o utilizador não estiver na *role*, atribui o utilizador definido pelo *e-mail* à *role canEdit*.



FIGURA 55 - ORGANIZAÇÃO DA PASTA ADMIN

Na última fase da criação da *role canEdit* para a página *admin.aspx* temos que inicialmente criar uma pasta onde terá as páginas *admin* e *.webconfig* – ficheiro de configuração – para que só possa ser acedida pelos utilizadores da *role canEdit*.

```

<?xml version="1.0"?>
<configuration>
  <system.web>
    <authorization>
      <allow roles="canEdit"/>
      <deny users="*/>
    </authorization>
  </system.web>
</configuration>

```

FIGURA 56 - CONFIGURAÇÃO DA PASTA ADMIN

Na imagem acima podemos ver que o ficheiro webconfig na página *admin* só vai permitir utilizadores que pertençam à *role canEdit*.

Como visto anteriormente, foi criado um *link* para a página *admin* que estaria invisível a qualquer cliente que visitasse a página. Para tornarmos o *link* visível para os utilizadores da *role canEdit*, foi editado o ficheiro *site.Master.cs*.

```

protected void Page_Load(object sender, EventArgs e)
{
  if (HttpContext.Current.User.IsInRole("canEdit"))
  {
    adminLink.Visible = true;
  }
}

```

FIGURA 57 - TORNAR A PÁGINA ADMIN VISIVIEL AOS UTILIZADORES DA ROLE CAN EDIT

A instrução de código acima permite que caso um utilizador que esteja numa sessão no nosso *website* e pertença à *role canEdit*, o *link* torna-se imediatamente visível para que possa fazer a administração da nossa aplicação.

TESTES DE USABILIDADE

Nesta etapa do trabalho irá ser mostrado o questionário colocado a trinta e cinco inquiridos escolhidos de forma aleatória que testaram a nossa aplicação de encomendas *online*. Depois serão expostos e analisados os resultados obtidos a partir desse questionário.

Os testes de usabilidade permitem compreender a interação entre a aplicação e o cliente. Assim, foi dado a oportunidade aos nossos inquiridos a experimentarem a nossa aplicação e o serviço de encomendas *online* onde responderam a perguntas baseando-se na experiência que tiveram.

O questionário tem como parâmetros ser anônimo e voluntario tendo somente informações pessoais dos inquiridos, o sexo e a sua faixa etária de modo a ser permitido a realização da amostra no tratamento estatístico aos dados recolhidos.

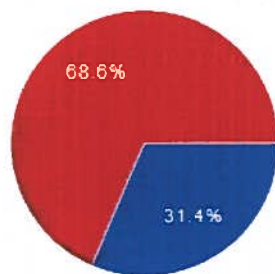
O questionário é constituído por oito questões sendo a última opcional. No início do questionário são colocadas apenas duas questões relativamente aos dados pessoais dos inquiridos que são o seu género e a sua faixa etária. A faixa etária compreende os seguintes intervalos: menores de dezasseis anos, dos dezasseis anos aos trinta anos, dos trinta e um anos aos cinquenta anos e maiores de cinquenta anos.

Na primeira questão é pedido ao inquirido uma classificação do *design* do *Website* numa escala de 1 a 5, sendo 1 para fraco e 5 para excelente. Na segunda questão questiona-se se o *Website* é de fácil navegação colocando para isso duas respostas possíveis: sim ou não, caso a resposta seja não os inquiridos podem juntar uma sugestão ou explicar porque pensam que a aplicação não é de fácil navegação. A terceira questão é pedida uma classificação, da Gestão de Pedidos, numa escala de 1 a 5, sendo 1 fraco e 5 excelente. A quarta questão pede-se uma classificação relativa à navegação pelas várias páginas sendo que 1 é fraco e 5 excelente. A quinta questão questiona se a informação prestada é pertinente, sendo uma resposta com duas opções: sim ou não. A sexta questão interroga se os serviços prestados pela aplicação são suficientes, caso seja positivo o inquirido deverá responder que sim, caso seja negativo a resposta será não.

ANÁLISE DOS DADOS RECOLHIDOS

Para este inquérito foram recolhidas trinta e cinco respostas sobre a experiência da utilização da aplicação com a funcionalidade de fazer encomendas *online*. Verificou-se que 31.4% dos inquiridos são do sexo feminino e 68.6% são do sexo masculino (gráfico 1) com idades compreendidas entre os menores de 16 anos e os a maiores de 50 anos, havendo uma maior incidência no grupo dos 16 aos 30 anos com 57.1% como se pode verificar no gráfico 2.

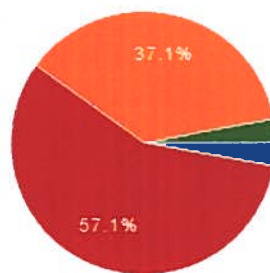
Sexo do inquirido



Feminino	11	31.4%
Masculino	24	68.6%

FIGURA 58 - GÉNERO DOS INQUIRIDOS

Faixa etária

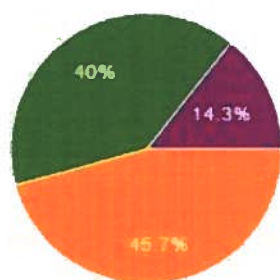


<16	1	2.9%
16>30	20	57.1%
31>50	13	37.1%
>50	1	2.9%

FIGURA 59 - IDADE DOS INQUIRIDOS

Em relação à primeira questão, que corresponde ao *design* do *Website*, 14.3% dos inquiridos classificou o aspeto visual do *Website* como 'excelente', 40% classificou o *Website* como 'bom', 45.7% como 'satisfatório' (ver gráfico 3). Com base nas respostas recolhida podemos concluir que o *design* do *Website* deve ser melhorado uma vez que queremos proporcionar uma melhor experiência aos utilizadores da plataforma.

Como classifica o design do WebSite?

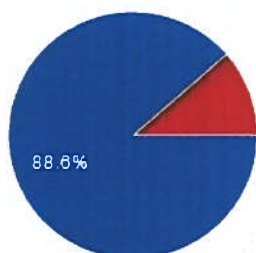


1	0	0%
2	0	0%
3	16	45.7%
4	14	40%
5	5	14.3%

FIGURA 60 - DESIGN DA APLICAÇÃO

A segunda questão pede ao inquirido que avalie a navegação na aplicação. Depois de feita a análise a esta pergunta, 88.6% dos inquiridos classificou a navegação como fácil e rápida e 11.4% não considera fácil e rápida a navegação na aplicação (gráfico 4). Conclui-se que nesta vertente, a aplicação vai ao encontro dos nossos inquiridos.

Considera fácil e rápida a navegação no website?

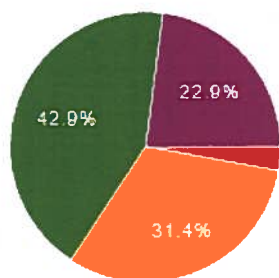


Sim	31	88.6%
Não	4	11.4%

FIGURA 61 - NAVEGAÇÃO DA APLICAÇÃO

Na terceira pergunta que é relativa ao modo de como a gestão dos pedidos são classificadas, 2.9% classificou como 2 ('insatisfatória'), 31.4% classificou como 3 ('satisfatória'), 42.9% classificou como 4 ('muito satisfatória') e 22.9% classifica como ('excelente'). Conclui-se que a gestão dos pedidos tem de ser melhorada para que seja fácil a gestão das encomendas.

Como classifica a gestão de pedidos?

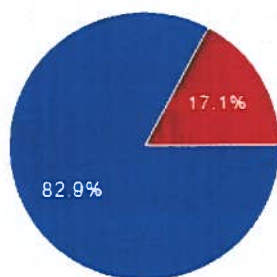


1	0	0%
2	1	2.9%
3	11	31.4%
4	15	42.9%
5	8	22.9%

FIGURA 62 - GESTÃO DE PEDIDOS

Na quarta pergunta é pedido aos inquiridos, que respondam com ‘sim’ ou com ‘não’ se consideram que o *Website* apresentado fornece de forma clara e pertinente a informação da empresa. 82.9% dos inquiridos respondeu ‘sim’ e apenas 17.1% respondeu ‘não’ (gráfico 6). Podemos concluir que embora haja um grande número de respostas positivas, existem aspetos mínimos que se podem melhorar para que haja a totalidade de respostas positivas.

Considera pertinente a informação prestada pelo website?

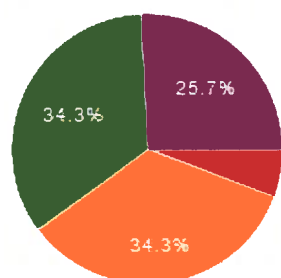


Sim	29	82.9%
Não	6	17.1%

FIGURA 63 - INFORMAÇÃO PERTINENTE

A quinta pergunta é referente à navegação entre as páginas da aplicação. Nove dos inquiridos (25.7%) respondeu como 5 (‘excelente’), doze dos inquiridos (34.3%) respondeu com 4 (‘muito satisfatória’), doze dos inquiridos (34.3%) responderam com 3 (‘satisfatória’), dois dos inquiridos respondeu com 2 (‘insatisfatória’) como se pode verificar no gráfico 7. Podemos concluir que talvez haja uma necessidade de melhorar a navegação entre páginas.

Classifique a navegação pelas várias páginas .

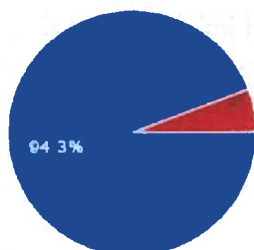


1	0	0%
2	2	5.7%
3	12	34.3%
4	12	34.3%
5	9	25.7%

FIGURA 64 - NAVEGAÇÃO PELAS PÁGINAS

A pergunta número seis é relativa aos serviços disponibilizados pela aplicação. Foi solicitado nesta pergunta aos inquiridos que respondessem ‘sim’ ou ‘não’ se consideravam a plataforma suficiente. 94.3% dos inquiridos respondeu de forma positiva e 5.7% de forma negativa (gráfico 8). Conclui-se que a maior parte dos inquiridos considera que os serviços disponibilizados são os necessários e suficientes para tornar esta aplicação rentável.

Considera suficiente os serviços prestados pelo website?



Sim	33	94.3%
Não	2	5.7%

FIGURA 65 - SERVIÇOS PRESTADOS PELO WEBSITE

Concluída a análise dos inquéritos e terminada a demonstração dos resultados, podemos afirmar que a aplicação e a plataforma de encomendas *online*, embora tenham tido resultados e apreciações positivas pela maior parte dos inquiridos, teremos que trabalhar no aspeto visual da aplicação, mas o objetivo foi atingido: o de fornecer aos clientes da empresa de restauração serviços práticos e cómodos na sua utilização. Os inquiridos consideram que tanto o aspeto visual como a navegação do *Website* podem ser melhorados podendo se tornar mais rica em termos de visualização da aplicação.

CONCLUSÃO

Neste capítulo serão apresentadas as conclusões do presente trabalho. O seu objetivo foi a criação de uma aplicação *ASP.net* para uma empresa de restauração e para isso foram abordados temas como a Gestão de Pedidos, Engenharia de *Software*, *SQL Server* e o *ASP.net*.

Com o desenvolvimento desta aplicação, a empresa em causa pode agora fornecer aos seus clientes uma solução de encomendas de serviços *online*. Para isso, a aplicação contém as informações acerca da empresa como a sua localização, os menus proporcionados aos seus clientes. Com a plataforma de encomendas *online* a empresa pode gerir a informação numa página da aplicação criada para o efeito, reduzindo assim o tempo de espera dos seus clientes e rentabilizar o seu tempo.

Este projeto levou cerca de onze meses a ser concretizado, desde o seu conceito inicial até ao fim do desenvolvimento do código. O projeto contou com a consulta de várias fontes, como livros, *sites*, artigos científicos e conhecimentos académicos.

Foi delineado um plano para abordar conceitos essenciais à sua realização, sendo feito uma filtragem daqueles pouco relevantes, abordando assim os fundamentais. Foi executado um tratamento dos textos onde foram selecionadas as ideias essenciais relativamente ao tema do presente projeto. Depois de algum estudo destas ideias avançou-se para a realização do Estado de Arte que contou com a ajuda de professores e profissionais na área das Tecnologias Informáticas.

Findo o Estado de Arte, deu-se início ao desenvolvimento do trabalho com a criação da aplicação. Estas duas tarefas realizaram-se em paralelo pelo facto de o desenvolvimento ser a demonstração e explicação daquilo que é o código da aplicação e que permite que esta funcione.

Concluído o desenvolvimento do trabalho, aprimorou-se a construção da aplicação graficamente. O aspeto de uma aplicação é fundamental, uma vez que não compreende apenas a componente visual, mas também a sua componente funcional, como a navegação entre as páginas ou a construção de ferramentas da plataforma *ASP.NET* necessárias.

A última fase deste projeto é referente aos testes de usabilidade sob a forma de questionários feitos de forma *online*, usando a plataforma Google realizados a inquiridos escolhidos de forma aleatória compreendendo todas as faixas etárias. O seu principal objetivo é adquirir alguma perceção confrontando os resultados obtidos, com o aprimoramento da aplicação para se tornar mais completa e mais eficiente para os utilizadores.

A análise dos inquéritos permitiu ter uma noção clara de que o aspeto visual da aplicação teria de ser trabalhado com o objetivo de se tornar agradável à vista dos utilizadores, mas por outro lado percebeu-se que a vertente funcional e a gestão da aplicação vai ao encontro dos inquiridos.

Concluída a discriminação das etapas realizadas neste projeto, pode ser afirmado que todos os objetivos propostos inicialmente foram atingidos assim como as etapas propostas.

A metodologia presente neste trabalho foi a mais adequada devido ao rigor que requer. A etapa mais complicada que surgiu na aplicação foi o aspeto visual da aplicação, uma vez que, foi realizado por um estudante de Informática e não de Engenharia Multimédia. A nível de programação e de execução de código foi bastante desafiante uma vez que com as ferramentas necessárias podemos executar projetos ao nível daqueles que os profissionais na área executam.

REFERÊNCIAS BIBLIOGRÁFICAS

Addy, Rod (2007). *Effective IT Service Management - To ITIL and Beyond!*. Heidelberg: Springer.

Barry, F. Curtis (2014). *Order Management System/Order Management Software*. F. Curtis Barry & Company, <http://www.fcaco.com/services/order-management-systems>

Bluemner, Adam (2013). *Work Order Management Software: Solutions for Performance Optimization*. Find Accounting Software, <http://findaccountingsoftware.com/focus-on-function-ality/work-order-management-software-solutions-for-performance-optimization>

Burnson, Forrest (2014). *Compare Order Fulfillment Software*. Software Advice, <http://www.softwareadvice.com/scm/order-fulfillment-software-comparison/>.

Burnson, Forrest (2014). *Top 10 Most Reviewed Order Fulfillment Software Systems*. Software Advice, <http://www.softwareadvice.com/scm/order-fulfillment-software-comparison/>

Gladwin, Lee Copeland (2001). *How-To Active Server Pages in Computerworld* <http://www.computerworld.com/article/2591795/app-development/active-server-pages.html>

Griffith, Eric (2013). *How to build a Website*. PCMAG, <http://www.pcmag.com/article/print/297782>

Griffith, Eric (2013) . *How to build a Website*, <http://www.pcmag.com/article/print/297782>

Guinn, Justin (2014). *Compare Field Service Software*. Software Advice, <http://www.softwareadvice.com/field-service/>

Liberty, Jesse & Hurwitz, Dan (2002). *Programming ASP.NET 3.5*. Estados Unidos: O'Reilly, 4ª Edição

McCown, Sean (2014). *Review: SQL Server 2014 pushes the pedal to the metal*. Infoworld, <http://www.infoworld.com/article/2610600/database/review--sql-server-2014-pushes-the-pedal-to-the-metal.html>

McGould, James (2013). *Microsoft SQL Server*. Microsoft, <https://msdn.microsoft.com/en-us/library/bb545450.aspx>

Mcnaughtan, Alan (2013). *Q&A. Do Technology Startups Need Product Managers?* <http://timreview.ca/article/695>

Mcpeak, Jeremy (2011). *Why Many Developers Hate ASP.NET...and Why They're Wrong*. Tutos+, <http://code.tutsplus.com/articles/why-many-developers-hate-aspnet-and-why-theyre-wrong--net-22975>

Menken, Ivanka (2007). *The IT Service Management ITIL V3 Benchmark Checklist*. The Art of Stress-free IT ServiceManagement, 62-110, <ftp://ftp.psu.ac.th/pub/itil/The%20Art%20of%20Stress%20free%20IT%20ser->

Panella, Jon (2014). *Order Management: Overview and Solution Options*. In SapientNitro
Tiedman, Mark J. (2013). *The Exceptional Product Manager: What is the Right Stuff?* <http://pragmaticmarketing.com/resources/the-exceptional-product-manager-what-is-the-right-stuff>

Silva, Ana Rute (2014). *Portugueses nunca compraram tanta comida online*. Público. Disponível em www.publico.pt

Vernon, David (2008). *A Review of Software Engineerin*. In Khalifa University. Software Engineering

ANEXOS

ANEXO I – QUESTIONARIO

Inquérito para estudo da performance do website de restauração

*Obrigatório

Sexo do inquirido *

- Feminino
- Masculino

Faixa etária *

- <16
- 16>30
- 31>50
- >50

Como classifica o design do WebSite? *

1 - para fraco 5- para excelente

- 1
- 2
- 3
- 4
- 5

Considera fácil e rápida a navegação no website? *

- Sim
- Não

Se respondeu não , de que forma podemos melhorar?

Como classifica a gestão de pedidos? *

1 - para fraco 5- para excelente

- 1
- 2
- 3
- 4
- 5

Considera pertinente a informação prestada pelo website? *

- Sim
- Não

Considera suficiente os serviços prestados pelo website? *

- Sim
- Não

Se respondeu não , de que forma podemos melhorar?

Caso tenha alguma ideia , para que possamos melhorar o website , por favor não hesite em mostrá-lo junto de nós.