



SOLUÇÃO DE LANDING PAGES BASEADA EM QR CODES

LEANDRO MIGUEL ROCHA MENDES

Junho de 2020

SOLUÇÃO DE LANDING PAGES BASEADA EM QR CODES

Leandro Miguel da Rocha Mendes

**Dissertação para obtenção do Grau de Mestre em
Engenharia Informática, Área de Especialização em
Sistemas Computacionais**

Orientador: Alexandre Bragança

Supervisor: Armando Rocha

Porto, julho 2020

Dedicatória

Ao meu avô, que apesar de não estar presente sempre desejou ver o meu sucesso.

Resumo

Nos dias de hoje é de grande importância obter mais informações sobre um produto ou serviço, quer o mesmo seja um bem adquirido ou exposto de alguma forma. A evolução tecnológica presente nos tempos de hoje traz novas necessidades para obter essa informação, pois implica que a mesma seja de fácil de exposição para quem a quer transmitir e de fácil e rápido acesso para a quem a pretende visualizar. À empresa Ifthen surgiu a possibilidade de desenvolver uma solução do género que permita a qualquer entidade pessoal ou empresarial visualizar e disponibilizar informação sobre produtos e serviços, de forma similar ao que acontece para as referências de multibanco que a empresa já disponibiliza.

A combinação da criação de uma página personalizável com a rapidez e facilidade de acesso através do uso de códigos QR, também eles totalmente personalizáveis, é algo que, tanto quanto foi possível pesquisar não existia no mercado.

Esta dissertação tem como objetivos disponibilizar uma solução do tipo Web que permita que os seus clientes diretos disponibilizem aos seus utilizadores finais (i.e., *clientes dos clientes*) o acesso a informação dinâmica e personalizada sobre produtos ou serviços. Este acesso deve ocorrer de forma automatizada recorrendo à digitalização dos códigos QR. Aos clientes diretos deve também ser disponibilizado um espaço onde possam gerir as páginas e códigos QR criados para os seus produtos ou serviços. A solução deve ainda ser facilmente utilizada por pessoas sem qualquer tipo de conhecimentos técnicos.

Para os objetivos serem alcançados foi necessário realizar um estudo sobre a indústria atual, a criação de páginas e da tecnologia de códigos QR onde se evidenciaram diversas características, funcionalidades e tipos. Através de um desenho detalhado e completo, foi possível implementar uma solução. Esta solução preconizada através das diversas etapas da dissertação cumpre a grande maioria dos objetivos delineados anteriormente. Os resultados obtidos e avaliação efetuada sobre a solução comprovam que o software desenvolvido é uma solução classificada como boa, onde pode ser melhorada a acessibilidade e facilidade da solução. A empresa requerente poderá oferecer um serviço de informação personalizado aos seus clientes, tal como desejava inicialmente, com a possibilidade de melhoria no futuro. Os clientes irão também poder informar e melhorar o acesso a informação que pretendem divulgar.

Palavras-chave: Páginas, Códigos QR, Produtos, Serviços, Informação

Abstract

Nowadays it's of great importance to obtain more information about any product or service, whether it's an acquired product or exposed product in some way. The technological evolution present nowadays brings new needs to obtain that information, because it implies that the information is of easy exposure for those who want to transmit it and of easy and fast access to those who want to view it. To the Ifthen company arose the possibility of developing a solution of this kind, one that allows any personal or business entity to view or provide any information about products or services, similarly to what happens with the ATM references that the company already offers.

The combination of creating a customizable page with the speed and ease of access using QR codes, which are also fully customizable, is something that, as far as it was possible to search, did not exist in the market.

This dissertation aims to provide a Web-type solution that allows its direct customers to provide their end users (i.e. customers' customers) with access to dynamic and personalized information about products or services. This access must occur in an automated way using the scanning of QR codes. Direct customers should also be provided with a space where they can manage the pages and QR codes created for their products or services. The solution must also be easily used by people without any technical knowledge.

For the objectives to be achieved it was necessary to carry out a study on the current industry, the creation of pages and the technology of QR codes where several characteristics, functionalities and types were evidenced. Through a detailed and complete design, it was possible to implement a solution. This solution recommended through the various stages of the dissertation fulfills most of the objectives outlined previously. The results obtained and the evaluation carried out on the solution prove that the software developed is a solution classified as good, where the accessibility and ease of the solution can be improved. The requesting company will be able to offer a personalized information service to its customers, as it initially wanted, with the possibility of improvement in the future. Customers will also be able to inform and improve access to information they wish to disclose.

Keywords: Pages, QR Code, Products, Services, Information

Agradecimentos

De forma geral, queria agradecer a todos que me ajudaram diretamente e indiretamente na realização desta dissertação. O apoio recebido nesta fase é imprescindível para a realização da mesma, portanto muito obrigado a todos.

Em primeiro, gostaria de agradecer ao Professor Doutor Alexandre Bragança, pois sempre se mostrou disponível para qualquer dúvida que tive. O seu apoio e críticas construtivas foram sem dúvida essenciais para atingir o nível necessário nesta dissertação.

Gostaria de agradecer a Iftthen, por me terem proporcionado esta oportunidade de trabalhar com eles e desenvolver esta solução. Queria agradecer em especial ao Filipe Moura pela confiança depositada em mim para este projeto. Queria agradecer também ao Armando Rocha, que supervisionou o meu desenvolvimento durante todo o tempo que tive presente na empresa e me ajudou a ultrapassar obstáculos no desenvolvimento. Agradeço também aos restantes colaboradores por fornecerem um excelente ambiente de trabalho dentro da organização.

A todos os meus amigos e colegas que me deram apoio. Pois foram uma ajuda sem igual durante todo o Mestrado e em dúvidas complexas durante a escrita da dissertação.

Agradeço a toda a minha família. A família para mim é das coisas mais importantes que se pode ter enquanto ser humano. O apoio incondicional que me foi dado durante todos estes anos irão ficar sempre marcados em mim. Tenho que agradecer do fundo do coração a todos, principalmente aos meus pais e ao meu irmão que me tiveram de aturar todos os dias durante a escrita desta dissertação e nos seus momentos menos bons.

Por fim, queria agradecer a minha namorada Sara. A Sara foi um apoio incondicional durante todo este processo e foi sem dúvida o meu pilar. Nos momentos em que mais ninguém me conseguia aturar era ela que me dava apoio, era a ela que pedia para ler o que tinha escrito e o que pedia opinião, era ela que chegava do trabalho a 23:30 da noite e me ouvia a falar do assunto quando o que mais queria era descansar. Só tenho a dizer, obrigado e amo-te muito, espero estar contigo para o resto das nossas vidas.

Índice

1	Introdução	1
1.1	Contexto	1
1.2	Apresentação da empresa	1
1.3	Problema	2
1.4	Objetivos	2
1.5	Abordagem	3
1.6	Estrutura do documento	4
2	Estado da Arte	7
2.1	Indústria 4.0	7
2.1.1	Conceitos e motivos da Indústria 4.0	8
2.1.2	Sistemas ciber-físicos (Cyber-Physical System)	8
2.2	Landing Pages	10
2.2.1	Conceitos e definição	10
2.2.2	Custos de uma Landing Page/Website	10
2.2.3	Abordagens para a criação de Landing Pages	11
2.3	Plataformas de criação de Landing Pages	12
2.3.1	Cardd	12
2.3.2	Wix	13
2.3.3	MailChimp	13
2.3.4	Análise e comparação das plataformas de criação de Landing Pages	14
2.4	Códigos QR (QR CODES)	15
2.4.1	Conceitos e definição de código QR	15
2.4.2	Principais Funcionalidades/Características do código QR	16
2.4.3	Correção de erros (<i>Error Correction</i>)	17
2.4.4	Estrutura do código QR	18
2.4.5	Versões do código QR	19
2.5	Tipos de códigos QR	20
2.5.1	Código QR (comum) Modelo 1 e 2 (<i>Model 1 and 2</i>)	20
2.5.2	Código QR micro (<i>Micro QR Code</i>)	21
2.5.3	Código QR com moldura (<i>Frame QR</i>)	21
2.6	Casos de uso de códigos QR	22
2.6.1	McDonald's	22
2.6.2	Amazon	23
2.6.3	Super Bock	24
2.6.4	Medicamentos (Laboratórios)	25
2.7	Plataformas/software de Códigos QR	26
2.7.1	The QR Code Generator	27
2.7.2	QR Code Generator	27
2.7.3	QR Code Monkey	28

2.7.4	QR Code Studio	28
2.7.5	Visualead.....	29
2.7.6	Análise e comparação das plataformas.....	29
2.8	Alternativas de Frameworks ao desenvolvimento	31
2.8.1	Blazor.....	32
2.8.2	React	33
2.8.3	Angular	33
2.9	Visão da Solução	34
3	Análise de valor	37
3.1	Análise de valor - O que é e para que serve?.....	37
3.2	Fuzzy Front of Innovation e Modelo NCD.....	37
3.3	Proposta de valor.....	41
3.4	Modelo de Negócio CANVAS.....	42
3.5	Analytic Hierarchy Process (AHP).....	45
3.5.1	Fase 1 - Construção da árvore hierárquica de decisão	45
3.5.2	Fase 2 - Comparação das alternativas e dos critérios escolhidos.....	46
3.5.3	Fase 3 - Prioridade relativa de cada critério	47
3.5.4	Fase 4 - Avaliar a consistência das prioridades relativas.....	49
3.5.5	Fase 5 - Construção da matriz de comparação paritária para cada critério.....	50
3.5.6	Fase 6 - Obter a prioridade composta para as alternativas.....	52
3.5.7	Fase 7 - Escolha da alternativa	53
3.6	Quality Function Deployment (QFD)	53
4	Análise e Desenho	55
4.1	Requisitos Funcionais	55
4.2	Requisitos Não Funcionais.....	57
4.3	Modelo de domínio	57
4.4	Casos de uso	59
4.4.1	Descrição formal dos casos de uso	60
4.4.2	Descrição detalhada do caso de uso principal: Criar uma <i>landing page</i>	65
4.5	Alternativas de Arquitetura.....	67
4.5.1	Alternativa 1	68
4.5.2	Alternativa 2	69
4.5.3	Alternativa 3	70
4.5.4	Alternativa 4	70
4.5.5	Escolha da alternativa	71
4.6	Desenho detalhado	71
4.6.1	Criar Landing Page	72
4.6.2	Visualizar Landing Page	73
4.6.3	Criar código QR	74
5	Implementação.....	77

5.1	Metodologia adotada para gestão do desenvolvimento.....	77
5.2	Modelo de dados.....	78
5.3	Estrutura da solução	79
5.4	Autenticação e autorização	80
5.5	Criação das Landing Pages.....	81
5.5.1	Configurações gerais (<i>Header e Footer</i>)	81
5.5.2	Componentes do corpo da página	81
5.5.3	Gravar e pré-visualizar páginas.....	84
5.5.4	Diagrama de sequência.....	85
5.5.5	Diagrama de classes	87
5.6	Criar código QR	88
5.6.1	Biblioteca de códigos QR selecionada (<i>QRCoder</i>)	88
5.6.2	Diagrama de sequência.....	90
5.6.3	Diagrama de classes	92
5.7	Visualizar Landing Pages	93
5.7.1	Pedidos externos/remotos.....	93
5.7.2	Diagrama de sequência.....	95
5.7.3	Diagrama de classes	96
6	Experimentação e Avaliação.....	99
6.1	Hipótese.....	99
6.1.1	1º Hipótese	99
6.1.2	2º Hipótese	100
6.2	Grandezas.....	100
6.2.1	Satisfação.....	100
6.2.2	Eficácia.....	100
6.2.3	Eficiência	100
6.2.4	Desempenho.....	100
6.3	Metodologia de Avaliação.....	101
6.3.1	Questionários para os requerentes e utilizadores da solução	101
6.3.2	Formulário de avaliação aos requisitos funcionais	104
6.3.3	Testes de desempenho	104
6.4	Resultados	105
6.4.1	Resultados Questionários	105
6.4.2	Avaliação da pontuação total do questionário SUS.....	110
6.4.3	Avaliação por heurísticas (Questionário SUS)	111
6.4.4	Avaliação das repostas à questão aberta	113
6.4.5	Resultado e avaliação do formulário de avaliação requisitos	114
6.4.6	Resultado dos testes de desempenho.....	114
6.4.7	Avaliação dos resultados dos testes de desempenho	115
7	Conclusão	117
7.1	Objetivos alcançados	117
7.2	Limitações encontradas e trabalho futuro	119

7.3	Apreciação final e pessoal	120
-----	----------------------------------	-----

Lista de Figuras

Figura 1 – Comparação da codificação entre um código de barras e um código QR [29]	15
Figura 2 – Estrutura completa de um código QR (Editada e adaptada) [29][38]	19
Figura 3 - Comparação entre a versão 1 e a versão 40 de um Código QR [39]	20
Figura 4 – Comparação entre o código micro (Micro QR Code) e o QR Code “normal” [55]	21
Figura 5 – Exemplos de códigos QR do tipo Frame (imagem editada e adaptada) [57][56]	22
Figura 6 - Exemplo de um produto McDonald's	23
Figura 7 – Imagem da publicidade em cacifos, pacotes de entrega e aplicação de SmileCodes[60][61]	24
Figura 8 - Imagem de uma cerveja Super Bock com um Código QR	25
Figura 9 - Medicamentos Generis a esquerda, a direita medicamentos de outros laboratórios	26
Figura 10 – Modelo de inovação [80]	38
Figura 11 - Modelo New Concept Development [82]	39
Figura 12 – Modelo Canvas da proposta de valor	42
Figura 13 - Modelo de negócio Canvas	44
Figura 14 – Árvore hierárquica de decisão	45
Figura 15 - Árvore hierárquica de decisão com os valores correspondentes	52
Figura 16 – Desdobrável QFD Preenchido	54
Figura 17 - Modelo de domínio da solução	58
Figura 18 - Diagrama de casos de uso da solução	59
Figura 19 – <i>Mockup</i> do estado inicial da página de criação de <i>landing pages</i>	66
Figura 20 – <i>Mockup</i> resumido exemplo de um componente adicionado	67
Figura 21 - Alternativa de arquitetura 1	68
Figura 22 - Alternativa de arquitetura 2	69
Figura 23 - Alternativa de arquitetura 3	70
Figura 24 - Alternativa de arquitetura 4	71
Figura 25 – Diagrama de sequência do sistema (Criar uma Landing Page)	73
Figura 26 - Diagrama de sequência do sistema (Visualizar Landing Page)	74
Figura 27 - Diagrama de sequência do sistema (Criar código QR)	75
Figura 28 – Documento de configurações e componentes disponíveis	78
Figura 29 - Extrato do código para implementar Identity com MongoDB (Startup.cs)	80
Figura 30 - Extrato código da injeção do utilizador nas páginas <i>Identity</i>	81
Figura 31 - Extrato de código da alteração necessária a injeção do utilizador	81
Figura 32 - Exemplo de interoperabilidade	82
Figura 33 - Tratamento do documento obtido da base de dados para obter dados para o componente selecionado	82
Figura 34 - Imagem de código de adição dinâmica de componentes	83
Figura 35 - Exemplo de inserção de documento na base de dados	84
Figura 36 - Trecho de código da inserção de um documento utilizando a funcionalidade de <i>TTL</i>	85

Figura 37 - Diagrama de sequência (Criar uma Landing Page).....	87
Figura 38 – Diagrama de classe do caso de uso “Criar Landing Page”	88
Figura 39 - Código para gerar um código QR básico	89
Figura 40 - Código exemplo de todas as opções de geração de código selecionadas.	89
Figura 41 - Diagrama de sequência (Criar código QR)	92
Figura 42 - Diagrama de classe do caso de uso (Criar código QR)	93
Figura 43 - Código do método de ciclo de vida de renderização modificado	94
Figura 44 - Trecho de código de pedidos externos.....	94
Figura 45 - Diagrama de sequência (Visualizar Landing Page)	96
Figura 46 - Diagrama de classe do caso de uso “Visualizar Landing Page”	97
Figura 47 – Classificação do valor obtido no SUS [109]	103
Figura 48 – Q1: Eu acho que gostaria de utilizar este sistema frequentemente	106
Figura 49 – Q2: Eu achei que o sistema era desnecessariamente complexo.....	106
Figura 50 – Q3: Eu achei que o sistema era fácil de utilizar.....	107
Figura 51 – Q4: Eu achei que iria necessitar de ajuda de um técnico de forma a conseguir utilizar este sistema.....	107
Figura 52 – Q5: Achei que as várias funcionalidades deste sistema estavam bem integradas	107
Figura 53 – Q6: Achei que as várias funcionalidades deste sistema estavam bem integradas	108
Figura 54 – Q7: Eu imagino que a maioria das pessoas aprenderia a utilizar rapidamente este sistema	108
Figura 55 – Q8: Eu achei o sistema muito complicado de utilizar	108
Figura 56 – Q9: Eu senti-me muito confiante a utilizar este sistema.....	109
Figura 57 - Q10: Tive necessidade de aprender muitas coisas antes de conseguir entender/utilizar este sistema	109
Figura 58 – Resultados obtidos com cálculo da pontuação SUS	111

Lista de Tabelas

Tabela 1 – Tabela comparativa de plataforma de criação de Landing Pages	14
Tabela 2- Capacidade de correção de erros de um código QR [32][33]	17
Tabela 3 - Tabela Comparativa de softwares de códigos QR.....	30
Tabela 4 – A escala Fundamental de Thomas L Saaty [88][89].....	47
Tabela 5 – Matriz de comparação de critérios	47
Tabela 6 – Matriz de comparação dos critérios do Segundo Nível	48
Tabela 7 – Matriz normalizada dos critérios do Segundo Nível.....	48
Tabela 8 – Peso dos valores/Prioridade relativa (Vetor de prioridades)	48
Tabela 9 – Tabela de Índices aleatórios de ordem n.	50
Tabela 10 – Matriz de comparação paritária do critério Funcionalidade	50
Tabela 11 – Matriz de comparação paritária do critério Funcionalidade normalizada	51
Tabela 12 – Matriz de comparação paritária do critério Facilidade de Utilização	51
Tabela 13 – Matriz de comparação paritária do critério Facilidade de Utilização normalizada	51
Tabela 14 - Matriz de comparação paritária do critério Inovação.....	51
Tabela 15 - Matriz de comparação paritária do critério Inovação.....	52
Tabela 16 – Requisitos não funcionais de acordo com o modelo FURPS+	57
Tabela 17 – UC01: Fazer login/Autenticação.....	60
Tabela 18 – UC02: Criar uma <i>landing page</i>	61
Tabela 19 – UC03: Criar código QR.....	62
Tabela 20 – UC04: Gestão de páginas e códigos	63
Tabela 21 – UC05: Estatísticas analíticas	64
Tabela 22 – UC06: Visualizar <i>landing page</i>	65
Tabela 23 – Escala de Likert com respostas correspondentes.....	103
Tabela 24 - Pontuação de cada resposta perante a escala	103
Tabela 25 – Resultados obtidos através da mediana das amostras utilizadas	115
Tabela 26 – Cálculo coeficiente de correlação	116

Acrónimos e Símbolos

Lista de Acrónimos

AHP	<i>Analytic Hierarchy Process</i>
CPS	<i>Cyber Physical System</i>
CSS	<i>Cascading Style Sheets</i>
DOM	<i>Document Object Model</i>
FEI	<i>Front End Of Innovation</i>
FFE	<i>Fuzzy Front End</i>
HTML	<i>HyperText Markup Language</i>
JIS	<i>Japanese Industrial Standards</i>
NCD	<i>New Concept Development</i>
NPD	<i>New Product Development</i>
NPPD	<i>New Product and Process Development</i>
POS	<i>Point Of Sale</i>
QR	<i>Quick Response</i>
SQRC	<i>Secure Quick Response Code</i>
SUS	<i>System Usability Scale</i>
URL	<i>Uniform Resource Locator</i>
2D	Duas dimensões

1 Introdução

Neste capítulo é introduzida a dissertação a desenvolver no âmbito do Mestrado em Engenharia Informática. Inicialmente, é apresentado o contexto, uma pequena apresentação da empresa onde irá ser desenvolvida a solução, o problema a resolver, os principais objetivos, a abordagem a seguir e por último, a estrutura do documento.

1.1 Contexto

A IFTHEN é uma empresa que atualmente fornece a empresas diversos serviços, no entanto, tem como atividade principal fornecer serviços de pagamentos por referências multibanco, MBWAY e PAYSHOP. A empresa pretende agora que seja desenvolvido um serviço totalmente novo, uma solução de uso comercial, que pretende permitir a qualquer entidade pessoal ou empresarial visualizar e disponibilizar informação sobre produtos e serviços. O acesso a informação deverá ocorrer com base em referências ou com outra tecnologia do género, de forma similar ao que acontece para as referências de multibanco que a empresa já disponibiliza.

1.2 Apresentação da empresa

A Ifthen é uma empresa portuguesa, fundada em 1998. Um dos objetivos principais da empresa é fornecer soluções informáticas aos seus clientes.

A empresa atualmente contém várias soluções com funcionalidades e objetivos diferentes. A solução Ifthenpay é uma *gateway* de pagamentos que permite geração de referências para empresas. Estas referências podem ser de multibanco, MBWAY e PAYSHOP. Qualquer empresa que queria gerar referências só tem de aderir ao serviço que a Ifthen proporciona.

Além desta solução, a empresa tem soluções como o IFPROTEC - uma aplicação informática que está vocacionada para as Proteções Civis Portuguesas e outras entidades que intervêm na

prevenção e coordenação de meios e recursos. O IFFIRE é uma aplicação desenvolvida exclusivamente para as Corporações de Bombeiros. É uma solução modular que cobre toda a área operacional e administrativa e que permite às Corporações de Bombeiros uma gestão mais moderna, mais organizada e mais eficiente. A empresa possui ainda módulos de integração e diversos outros softwares desenvolvidos.

1.3 Problema

Atualmente, no mercado, existem diversos produtos que utilizam as etiquetas tradicionais (etiquetas com informação impressa diretamente). As etiquetas tradicionais são muito limitadas em espaço de impressão para informação.

No entanto, já existem no mercado algumas soluções que fazem uso de códigos QR para mitigar a situação anterior. Estas soluções são também elas limitadas em diversas características visuais e funcionais deste tipo de código. Além disso estes serviços/software apenas permitem gerar códigos que redirecionam o utilizador para urls estáticos.

Uma loja de roupa pode querer disponibilizar mais informações acerca dos seus produtos e utilizar uma das soluções existentes no mercado. Porém, uma loja de roupa vende diversos produtos com características diferentes e ao utilizar uma destas soluções implicaria já terem previamente desenvolvido algo tipo software ou página. Uma loja pode pretender uma solução que lhe permita criar para cada um dos seus produtos uma página personalizável em que se pode adaptar componentes/campos de acordo com as características do produto (Ex: tamanho, cores). A combinação da criação de uma página personalizável com a rapidez e facilidade de acesso através do uso de códigos QR, também eles totalmente personalizáveis, é algo que, tanto quanto foi possível pesquisar (e é relatado no capítulo 2), não existe atualmente no mercado.

Os vendedores e lojas ao utilizarem um possível software deste género irão permitir ao cliente final ter sempre disponível informação do produto e permitir a pessoas que nunca tenham adquirido nenhum produto do género descobrir a marca e o vendedor. Um simples código impresso numa etiqueta irá servir como método de informação e promoção.

1.4 Objetivos

O objetivo desta dissertação é o desenvolvimento de uma prova de conceito de uma solução de serviço no âmbito do problema descrito anteriormente.

A visão para este problema/solução é possibilitar obter informações de todo o tipo (imagens, vídeo, etc.) sobre um produto/serviço (ex.: fornecedores) e atualizados em tempo útil (ex.: stocks), através de dados guardados nos próprios códigos QR e pela sua simples leitura usando qualquer aplicação/digitalizador de códigos QR.

A solução deve ainda inovar por permitir elevada personalização por parte do cliente de como o conteúdo guardado no código é posteriormente apresentado. O *display* de informação personalizado poderá ser feito através do uso de *landing pages*. As *landing pages* são páginas web, que como referido, deverão permitir aos clientes criar a sua própria estrutura de como os dados vão ser apresentados. Ou seja, devem permitir a capacidade de adicionar, editar ou remover campos/componentes.

Deverá também permitir personalizar o aspeto do código em si (ex: tipos de código, inversão de cores) e oferecer capacidades de recuperação/correção para a informação guardada nos códigos (ex: permitir ao cliente escolher um nível de correção do código para a informação guardada).

Este tipo de serviço já foi solicitado por muitos parceiros e clientes da Ifthen. São objetivos da Ifthen, como dono do projeto, e, por consequência, objetivos deste projeto, atingir os seguintes resultados:

- Disponibilizar uma solução do tipo Web que permita que os seus clientes diretos disponibilizem aos seus utilizadores finais (i.e., clientes dos clientes) o acesso a informação dinâmica e personalizada sobre produtos ou serviços.
- A solução deverá ocorrer de forma automatizada, sem introdução manual de dados por parte dos utilizadores finais, recorrendo à digitalização dos códigos QR.
- Aos clientes diretos deve também ser disponibilizado um espaço onde possam gerir as páginas e códigos QR criados para os seus produtos ou serviços.
- Permitir que a solução possa ser facilmente utilizada por pessoas sem qualquer tipo de conhecimentos técnicos.

1.5 Abordagem

A abordagem que foi adotada na realização deste projeto visa ter em conta principalmente a empresa que requereu esta solução, pois são eles os maiores interessados. De seguida é evidenciada uma abordagem passo a passo para dar resposta as necessidades dos mesmos.

- Primeiro é efetuada uma reunião com a empresa de forma a perceber o problema a resolver e objetivos iniciais.
- Depois de um primeiro contacto é efetuado um estudo das possíveis soluções, tecnologias e abordagens a utilizar. O capítulo Estado da Arte é o resultado desse estudo, onde é possível retirar novas informações e melhorar a visão anterior da solução.

- É importante após o estudo efetuado voltar a reunir com as partes interessadas de forma a expor aquilo que foi encontrado e adquirido com o estudo. Desta forma são expostas novas informações, o que poderá fazer mudar o caminho a seguir bem como a adoção de novas estratégias.
- Após definida a visão da solução é importante analisar se a mesma tem valor e realizar possíveis desenhos da solução a conceber.
- Estas análises devem de novo ser apresentadas às partes interessadas para se proceder a implementação concreta do software.
- A partir deste momento é iniciado o desenvolvimento do software. O desenvolvimento é feito utilizando uma abordagem ágil *scrum*. O desenvolvimento é realizado em pequenos sprints, ao fim de cada sprint é apresentada ao supervisor e às restantes partes interessadas o estado do software e as novas tarefas a realizar. Estas iterações repetem-se até a uma data limite estipulada ou o desenvolvimento se encontrar finalizado.
- Por fim, são efetuadas um conjunto de avaliações à solução preconizada e são apresentados os resultados as partes interessadas.

1.6 Estrutura do documento

Este documento encontra-se dividido em 7 capítulos seguidos pelas referências utilizadas.

O primeiro (e atual) capítulo (1. Introdução) é constituído pelo contexto, uma pequena apresentação da empresa que requisitou e introduziu a necessidade de resolver um problema, o problema que levou a escrita e desenvolvimento desta dissertação, os objetivos a alcançar com a mesma e por fim a presente estrutura do documento para uma melhor localização e explicação de algum tipo de informação presente nesta dissertação.

No segundo capítulo (2. Estado da Arte) apresenta-se, como o nome indica, o estado da arte das áreas tecnológicas/científicas relativas à tese. Inicialmente, é efetuada uma apresentação do estado da indústria, apresentação de conceitos e plataformas de *Landing Pages*, da tecnologia de códigos qr em si, uma pequena apresentação de plataformas e softwares de códigos qr no mercado e por fim são também apresentadas algumas alternativas de *Frameworks* que poderão ser utilizados para o desenvolvimento do software.

No terceiro capítulo (3. Análise de valor) efetua-se uma análise de valor à solução a ser desenvolvida. Aqui serão aplicados vários modelos de análise e construído um modelo de negócio.

No quarto capítulo (4. Análise e Desenho) apresenta-se uma análise e desenho da solução a desenvolver através de requisitos, casos de usos, alternativas e um desenho detalhado.

No quinto capítulo (5. Implementação) evidencia-se algumas metodologias utilizadas para o desenvolvimento e pontos principais de implementação como lógica e código utilizado.

No sexto capítulo (6. Experimentação e Avaliação) avalia-se a solução através dos métodos de avaliação como inquéritos a solução concebida.

No sétimo capítulo (7. Conclusão) apresenta-se a conclusão desta dissertação onde é possível perceber os objetivos cumpridos, limitações e uma apreciação global do trabalho realizado.

Existem por fim uma seção não numerada, a secção (Referências) onde estão as referências utilizadas ao longo deste documento.

2 Estado da Arte

Neste capítulo é efetuado um estado da arte das áreas tecnológicas/científicas relativas à tese, através da descrição de alguns conceitos, métodos e abordagens existentes mais úteis para a resolução do problema apresentado.

O capítulo é composto por uma análise da Indústria 4.0 (onde é evidenciado o seu estado atual), uma análise aos conceitos, abordagens e ferramentas existentes na criação de *Landing Pages*, uma análise similar ao tópico anterior acerca da criação de códigos QR. São também apresentadas algumas alternativas de *Frameworks* a utilizar no desenvolvimento da solução. Por fim, é efetuado uma visão global perante toda a informação recolhida nos tópicos anteriores.

2.1 Indústria 4.0

A indústria é caracterizada por ser uma das partes da economia que produz bens materiais, que normalmente contém elevados níveis de automatização e mecanização. No passado, sempre que existia um grande passo de evolução em alguns componentes dava-se o nome de revolução industrial. Ao longo dos tempos existiram algumas: a primeira revolução que se caracterizou pela evolução mecânica, a segunda que introduziu o uso intenso da eletricidade e a terceira que espalhou o uso da digitalização. Com o aparecimento da digitalização e o crescimento das tecnologias da internet deu-se uma mudança de paradigma. A partir desse momento a visão do futuro foi a produção de sistemas modulares e eficientes capazes de controlar o seu próprio processo de fabrico. Este avanço é feito a pensar num produto singular, mas tendo em conta a aplicação do conceito de produção em massa. Esta nova revolução denomina-se Indústria 4.0 (Industry 4.0) sendo o termo uma maneira de homenagear as versões de software [1].

2.1.1 Conceitos e motivos da Indústria 4.0

Ao longo dos tempos foram definidos alguns termos ou conceitos para a chamada Indústria 4.0. Seguem-se agora alguns desses conceitos que se podem encontrar através de pesquisas no google académico, sendo que alguns destes vêm de artigos, livros ou revistas científicas. São aqui evidenciados alguns destes conceitos que irão ajudar a transparecer e a entender melhor do que se trata.

A definição de Hermann et al define que a indústria 4.0 é um “termo coletivo para tecnologias e conceitos da cadeia de valor de uma organização” [2]. Existe outra definição que diz que a indústria é “a integração de maquinaria física complexa e dispositivos com sensores e software em rede, usado para prever, controlar e planear melhores resultados empresariais e sociais ” definida pelo Consortium II [3]. Henning e Johannes por outro lado definem esta indústria referindo que é “um novo nível da cadeia de valor de uma organização e de gestão através do ciclo de vida dos produtos.” [4].

Como já foi referido, estas definições são apenas algumas daquelas possíveis de encontrar, sendo uma pequena amostra de uma vasta pesquisa de conceitos. Com base nos três conceitos evidenciados anteriormente, pode-se observar que não existe uma definição absoluta e adotada para Indústria 4.0, e que este termo é algo que é definido de maneira diferente dependendo das perspetivas e da área científica da pessoa que a define. Não existir uma definição absoluta é algo que acontece para muitos outros termos científicos.

A indústria 4.0 é marcada pelo aparecimento de tecnologias de software web e mobile, capazes de responder às necessidades dos consumidores. A evolução acontece quando é dado ao consumidor facilidade de acesso a informação e a aquisição de produtos ou serviços. O facto de tudo isto estar à distância de um clique, sem necessidade de efetuar qualquer tipo de deslocação física e instalação, é uma mais valia. Uma mais valia que é cada vez mais uma obrigatoriedade que uma necessidade, pois se não for aplicada os consumidores irão optar por outras alternativas que o façam.

É importante ter em mente o estado atual da indústria, para que a solução a desenvolver no âmbito desta dissertação vá em conta o paradigma existente e seja adotada e escolhida pelos consumidores no mercado. Inclusive, o facto desta solução poder vir a ser utilizada por diferentes setores da indústria faz este estudo ganhar ainda mais relevo.

2.1.2 Sistemas ciber-físicos (Cyber-Physical System)

O aparecimento desta indústria deu-se a fatores de diferentes teores como a necessidade da redução do tempo de desenvolvimento de qualquer tipo de software, a necessidade de individualizar sistemas de acordo com as necessidades dos clientes, a flexibilidade, a descentralização e por fim a eficiência dos recursos, pois hoje em dia é importante pensar no impacto ecológico de forma a manter a sustentabilidade [5].

Os fatores anteriores fizeram a indústria ser revolucionada, sendo que a mesma se fundamentou com os seguintes princípios: a garantia de assegurar a virtualização dos sistemas, a descentralização, a interoperabilidade, a disponibilização de serviços em tempo real e forma flexível, a garantia de modularidade e de sistemas orientados a serviços. Foi neste sentido e sobre estes princípios que se deu o aparecimento dos sistemas ciber-físicos [6].

Estes sistemas procuram reduzir o papel da mecânica e hardware substituindo os mesmos por aplicações de software, dando então a estas um papel muito mais relevante [7]. Como já foi referido anteriormente, o aparecimento destes sistemas só é possível devido a diversos avanços tecnológicos, tendo como base a evolução de ferramentas e aplicações em combinação com outros sistemas avançados com capacidades extraordinárias de armazenamento e velocidade de transmissão de dados.

Estes tipos de sistemas ciber-físicos são considerados uma conexão entre o mundo “virtual” (componente digital) e o mundo “real” (componente físico) [7]. Segundo vários estudos efetuados, é possível identificar que estes sistemas podem ser caracterizados em 5 dimensões e a interligação destas dimensões fornecem a este tipo de sistemas CPS uma maior abertura e aumento de inteligência. As 5 dimensões podem ser identificadas de seguida [8]:

- A conexão ou interligação entre os dois mundos (Físico e virtual)
- Sistemas com limites de capacidade adaptáveis (dinâmicos), de forma a poderem escalar
- Adaptação consoante o contexto efetuada de forma própria (autónoma) e com controlo disponibilizado em tempo útil
- Marcada pela forte cooperação entre o ser humano e o próprio sistema
- Um sistema que permite distribuição e mudanças de controlo

Uma solução que utilize códigos QR poderá ser considerada um sistema ciber-físico. É possível verificar a afirmação anterior através do simples caso de um vendedor de camisas. Um vendedor de camisas que forneça o seu produto com um código QR irá criar uma ponte entre o mundo real e virtual.

No mundo real/físico iremos ter o código QR impresso. Através de uma câmara de um smartphone ou qualquer aplicação de leitura de códigos QR ou de digitalização consegue-se digitalizar o código QR fornecendo assim a ponte e conexão da parte real com a virtual. Por fim, temos a parte virtual que irá ser a informação apresentada após a digitalização que poderá ser informação de outro tipo de camisas, tamanhos e cores disponíveis do vendedor.

2.2 Landing Pages

Nesta secção são apresentadas as chamadas *Landing Pages*. Inicialmente, irá ser evidenciado um pouco o que são estas páginas e para que servem. São apresentadas algumas das metodologias utilizadas, o público alvo e o custo de produção de uma página do género.

2.2.1 Conceitos e definição

Uma *Landing Page* é uma página web, que normalmente, oferece um recurso ao visitante em troca de informação de contacto do mesmo. Uma *Landing Page* deve-se focar num fluxo específico, o do visitante. Estas páginas contém informações específicas sobre o produto ou serviço em questão, uma vez que as pessoas que irão parar a estas *Landing Pages*, terão já à partida um grande interesse no produto pois de alguma forma foram parar à página em questão. O fator de interesse dos visitantes combinado com informação específica do produto irá proporcionar que os visitantes preencham campos na página de forma a que se possa obter mais informação sobre o mesmo e converter essas pessoas em clientes [9].

Existem dois subtipos de *Landing Pages*. Algumas *Landing Pages* têm como função principal preencher ou recolher informação dos visitantes em troca de algo (Ex: promoções, ofertas). Outras *Landing Pages* denominadas de “*click-through*” servem para dar informação de um produto ou serviço, para que se possa ser redirecionada para uma página de aquisição do mesmo [10].

2.2.2 Custos de uma Landing Page/Website

A construção de um website personalizado normalmente é algo bastante dispendioso, cujo valor aumenta exponencialmente consoante as funcionalidades presentes. Existem diversos fatores que afetam o preço de um website e que por sua vez fazem alterar o preço. Um *freelancer* pode cobrar para construir um website entre 10€ a 50€ por hora, empresas mais profissionais e maiores podem cobrar entre a 60€ a 200€ por hora. Em média, um website pode custar entre 250€ a 10000€ ou ainda mais, tudo irá depender do tempo perdido e das funcionalidades embutidas [11].

De referir, que o valor abordado anteriormente apenas remete para o desenvolvimento e construção pois, como já foi referido, é necessário ter atenção outros fatores como a necessidade de registar um domínio, um servidor para hospedagem do website e realizar a manutenção do website para que se encontrem erros e problemas e se mantenha o website funcional. Uma alternativa para poupar custos e aumentar a rapidez com que um website fica disponível é a utilização de ferramentas de criação de websites e *Landing Pages*. Desta maneira por apenas o preço de uma de subscrição o utilizador tem na sua mão as ferramentas necessárias [12].

2.2.3 Abordagens para a criação de Landing Pages

Existem algumas metodologias diferentes para criar *Landing Pages*, algo que irá ser evidenciado mais à frente. No entanto, é necessário ter em conta o público-alvo em questão. A maioria das plataformas de criação de Websites e *Landing Pages* são efetuadas com a ideia de que o utilizador já tem algum tipo de conhecimento informático ou conhece alguns termos ou metodologias a utilizar. Serão apresentadas algumas das metodologias existentes.

2.2.3.1 Markdown

Nas páginas web é utilizado como linguagem principal de formatação o *HTML*. O *HTML* utiliza um conjunto de *tags* envolventes ao texto para a formatação. Quando queremos colocar um texto em negrito em *HTML*, é necessário envolver o texto de *tags* do género “ este texto está a negrito ”. O *HTML* pode-se tornar rapidamente complexo devido a quantidade de *tags* utilizadas, o que não apresentara qualquer tipo de problema para os sistemas de computação atuais. No entanto, para um utilizador comum, o *HTML* não é algo que é propriamente amigo do utilizador o que torna por vezes difícil ler um simples texto [13]. Uma das alternativas existentes é o *markdown*.

O *markdown* é uma linguagem de marcação leve que tem como principal funcionalidade aplicar formatação em texto ou em documentos somente de texto. A formatação utilizada pela *markdown* funciona de forma diferente de como se aplica formatações, por exemplo, num documento word. Num documento word a formatação é aplicada através de cliques e acesso a definições específicas. No *markdown* a formatação é efetuado no texto em si, ou seja, quando queremos que um determinado texto apareça com um formato, tamanho ou cor diferente essa indicação é efetuado no próprio texto [14].

Como referido, o *markdown* é de fácil interpretação. De seguida apresentam-se alguns exemplos. Para se colocar um texto em negrito no *markdown* é necessário colocar a palavra envolvida em dois asteriscos de cada lado. Para se colocar a palavra “livro” em negrito escreve-se a palavra da seguinte maneira “**livro**”. Uma outra enorme vantagem do *markdown* é a sua conversão fácil para *HTML*. A própria linguagem tem em si embutida esta funcionalidade [15].

Para a criação de *Landing Pages*, onde a mesma é efetuada normalmente por pessoas como pouco experiência informática, o *markdown* é uma das abordagens possíveis. Desta maneira o utilizador pode, de uma forma mais perceptível para si mesmo, aplicar estilos e formatações, que mais tarde são convertidas pelo sistema no *HTML* correspondente. Desta forma é possível apresentar aos utilizadores páginas ricas e personalizáveis, de forma simples e eficaz.

2.2.3.2 Drag & Drop (Arrastar e Soltar)

Outra metodologia e abordagem utilizada na criação de páginas e websites é o *Drag & Drop* (arrastar e soltar). O *Drag & Drop* é o método de selecionar algo (componentes, elementos) com o rato ou dedo. Esse elemento é depois arrastado até uma área onde é possível largar (drop) o mesmo. Atualmente existem diversas bibliotecas que permitem arrastar conteúdos

HTML (Ex: *interact.js*, *draggable*) e até o próprio *HTML* inclui de forma nativa uma *API* de *Drag & Drop* que permite, de forma simples, arrastar conteúdo entre duas tags “<div>” [16][17]. Este método é especialmente eficaz na criação de páginas pois permite, na maior parte das vezes, apresentar conteúdo de forma visual ao utilizador. No entanto, este método sozinho só permite o arrastar de conteúdos de *HTML* sendo que necessita de ser combinado com outro tipo de abordagem para permitir configurações adicionais como tamanhos e cores, entre outros. O *markdown* pode surgir como uma abordagem de apoio para se conseguir configurar os componentes arrastados. Cada componente ou campo inserido numa página através de (*Drag & Drop*) seria posteriormente personalizado através de *markdown*.

2.3 Plataformas de criação de Landing Pages

Existem no mercado algumas plataformas que permitem a criação deste tipo de páginas, variando em funcionalidade, características e preço. No âmbito deste projeto foi necessário estudar algumas das soluções disponíveis, enumerando alguns dos seus pontos fortes e fracos. Essa análise permite perceber onde a solução a desenvolver se diferencia e de que forma se adapta ao contexto, problema e objetivos delineados inicialmente nesta dissertação.

2.3.1 Cardd

Esta plataforma apresenta uma página principal bastante ilustrativa e simples, o que torna apelativo o seu uso. Inicialmente, como indicada pela plataforma, são proporcionados diversos templates base [18].

O utilizador pode escolher iniciar a edição por qualquer uma destas opções oferecidas. As opções estão devidamente organizadas por categorias sendo que muitas são de uso gratuito e outras apenas acessíveis através de uma subscrição. Para além de todas estas opções com conteúdo pré-definido, o utilizador pode optar por selecionar uma página/tela vazia [19].

Após a escolha inicial é iniciada a edição da página. Aqui é possível adicionar diversos tipos de campo, sendo eles, entre outros, campos de texto, listas de texto, imagens, botões, links, vídeo e tabelas. De referir que algumas opções estão bloqueadas e existe a necessidade de obter a versão paga como, por exemplo, a adição de formulários e adição de código externo. Esta última opção é bastante interessante pois permite adição de *HTML*, *CSS* e *JavaScript* externo [19].

A ordem dos campos pode ser trocada através de *drag and drop*. O utilizador consegue arrastar o campo para aparecer antes ou depois de outro selecionado. Cada campo tem ainda configurações extra como o comprimento, altura, cores e fontes de letra [19].

A publicação da página acontece com um domínio do próprio site na versão gratuita e com um domínio personalizado na versão paga. Esta plataforma não tem qualquer tipo de estatísticas sobre as páginas criadas [19].

2.3.2 Wix

O Wix é uma plataforma que bastante complexa que permite a criação de sites, sendo a maior parte das funcionalidades gratuitas. A plataforma permite escolher entre vários *templates* gratuitos como base inicial, sendo que estes se encontram organizados por categorias e áreas de negócio. Após a seleção de um *template* é iniciada a edição da página. A plataforma permite ainda que seja criado um site/página automático com base num conjunto de questões colocadas ao utilizador [20][21].

A plataforma permite criar várias páginas interligadas entre si, ou seja, um site completo. No entanto, é possível ao utilizador optar por configurar apenas uma página. Nesta opção a plataforma permite mudar o fundo de cada página e acrescentar novos campos como caixas de texto, imagens, botões, listas e vídeos. A plataforma oferece uma grande variedade de opções de campos e muitos destes permitem a conexão a serviços externos para obter informação.

É possível também adicionar diversas aplicações internas disponíveis no *Wix App Market*. Este contém aplicações gratuitas e pagas com diversas funcionalidades. Estas aplicações são desenvolvidas pela própria Wix ou por outros membros da comunidade e podem ser integradas facilmente no site ou página a criar. Às páginas é ainda possível adicionar um blog e uma área de marcações [22].

Por fim, é possível publicar a página no domínio da Wix ou, por um custo adicional, num domínio personalizado. Após a publicação são escolhidas algumas palavras para que o site seja mais facilmente encontrado numa pesquisa no google.

A plataforma oferece um painel de controlo para que seja possível gerir os sites/páginas criadas. De referir, que na parte de edição de páginas, devido à plataforma oferecer diversas funcionalidades e utilizar por defeito um *template* escolhido, o conteúdo da página não aparece de imediato. Pode-se ainda tornar confuso trabalhar com a plataforma devido a sua elevada complexidade [23].

2.3.3 MailChimp

Nesta plataforma para se criar uma *Landing Page* é necessário primeiro escolher um nome para a página e o seu público alvo [24].

De seguida, é necessário começar a selecionar o conteúdo da página. Inicialmente é necessário escolher um modelo de *Landing Page*. São proporcionados três modelos (*templates*) básicos: um para aceitar pagamentos, um com objetivo de aumentar a lista de clientes com um formulário embutido e outro para promover produtos. Para além destes modelos base, a plataforma proporciona ainda modelos temáticos, que já incluem por defeito imagens de fundo e espaço de texto pré-definidos e estruturados [25].

Existe a possibilidade de adicionar mais conteúdos e campos. É possível adicionar blocos de conteúdo como formulários e também conteúdos simples como botões, vídeos e códigos promocionais, assim como modificar o *header* e o *footer*. Na criação da página é possível clicar num botão de pré-visualização para visualizar os conteúdos adicionados e se estão de acordo com as necessidades pretendidas [26][27].

Por fim, é possível adicionar um título à *Landing Page* que irá aparecer na barra do navegador e personalizar o *URL*. A personalização do *URL* varia consoante a versão a utilizar. Em algumas versões é sempre necessário ter presente um domínio do mailchimp e noutras versões pagas é possível adicionar um domínio de propriedade do utilizador que cria a página. A ferramenta permite ainda rastrear a página através de serviços de integração como o Pixel do Facebook e do Google Analytics. As páginas podem ainda ser posteriormente editadas ou eliminadas [24].

2.3.4 Análise e comparação das plataformas de criação de Landing Pages

Perante a amostra de plataformas abordadas, é possível efetuar uma pequena análise através de uma tabela comparativa (Tabela 1). Nesta é possível observar pontos de destaque, de forma a que seja possível apoiar a solução no âmbito desta dissertação. Os pontos de comparação utilizados na tabela são um cruzamento dos destaques das aplicações e alguns dos objetivos da solução a conceber (Ex: intuitivo).

Tabela 1 – Tabela comparativa de plataforma de criação de Landing Pages

Descrição	Cardd	Wix	MailChimp
Intuitivo (fácil utilização)	✓	✗	✓
Templates	✓	✓	✓ / ✗
Ligação com aplicações/informação externa	✓	✓	✓
Espaço de gestão de páginas	✗	✓	✓
Aplicações/Widgets internos para aplicação na página	✗	✓	✗
Estatísticas Analíticas	✗	✓	✓
Versão gratuita	✗	✓	✗
Boa performance (tempo de carregamento da página baixo)	✓	✗	✓

É possível observar que todas as plataformas evidenciadas têm os seus pontos fortes e fracos. Em geral, a aplicação que parece ser mais completa das três observadas será o Wix, no entanto o site tem um desempenho baixo dado a enorme quantidade de funcionalidades. A plataforma torna-se até um pouco confusa com demasiados menus e opções. Por outro lado, o Carrd é uma plataforma menos completa, mas muito mais amigável do utilizador, os menus são

simples e de fácil compreensão. O Mailchimp encontra-se a meio termo perante as duas, não é tão completo quanto o Wix, mas é mais que o Carrd, proporcionando uma experiência satisfatória nas suas interfaces.

2.4 Códigos QR (QR CODES)

Nesta secção é efetuado um estudo aprofundado sobre códigos QR. Inicialmente, é evidenciada um pouco a história e origem destes códigos. De seguida descrevem-se os tipos de códigos, quais as suas principais funcionalidades (evidenciando-se o que as distingue de outras tecnologias existentes) bem como funcionalidades avançadas de maior complexidade. São ainda apresentadas as várias versões existentes de códigos QR e a sua estrutura. Por fim, é explicado como determinar a versão de um código QR.

2.4.1 Conceitos e definição de código QR

O aparecimento do código QR deu-se devido a fatores económicos e sociais [28]. Um Código QR acrónimo para (*Quick Response*) “rápida resposta” é um código de duas dimensões inventado pela companhia Denso Wave [29]. Este código foi inventado no ano 1994 e foi denominado desta forma “QR” pois tal como o nome indica, tinha como principal objetivo do seu criador, a leitura de conteúdo a velocidades muito elevadas [30].

Este tipo de código QR difere dos códigos de barras normais, pois permite a codificação de dados quer de forma vertical quer de forma horizontal. Assim, permite mais capacidade de armazenamento que um código 2D normal, sendo que consegue armazenar cerca de 100 vezes mais informação. A Figura 1 ilustra, de uma maneira visual, essa diferença de codificação entre o código 2D normal (à esquerda) e o código QR (à direita) [29].



Figura 1 – Comparação da codificação entre um código de barras e um código QR [29]

A informação contida nestes tipos de código pode ser obtida/lida através da câmara fotográfica de um simples smartphone e processada por um leitor denominado frequentemente de digitalizador. Este tipo de códigos tem muitas outras características pelas

quais são reconhecidos e definidos. Essas características são evidenciadas ao longo do documento e incluem, por exemplo, diferentes tipos, versões e padrões.

2.4.2 Principais Funcionalidades/Características do código QR

Como a maior parte das tecnologias, os códigos QR tem diferentes funcionalidades/características. Nesta secção são apresentadas aquelas que são consideradas as principais e são expostas pelos próprios criadores desta tecnologia, a Denso Wave. Existem funcionalidades básicas e outras que são bastante complexas e requerem um entendimento mais profundo da estrutura, modelo e tipo do código QR.

Tamanho de impressão muito baixo - O código possui um tamanho muito baixo quando impresso devido a um facto apontado anteriormente, a capacidade de codificação horizontal e vertical (quando comparado com um código de barras normal que apenas tem horizontal). Para a mesma quantidade de dados, o tamanho de impressão de um código QR vai ser sempre menor que um código de barras [31].

Elevada capacidade de codificação – Como referido anteriormente, os códigos QR diferem dos códigos normais existentes que normalmente apenas podem conter cerca de 20 dígitos. Os códigos QR têm capacidade bastante mais elevada de armazenamento, sendo capazes de trabalhar com todo o tipo de dados, como por exemplo, números, caracteres, binários e todas as três diferentes formas de escrita japonesas (Kanji, Kana, Hiragana). O número máximo de caracteres num único código QR é atualmente de 7,089 (utilizando a versão 40) [31].

Resistência a danos e sujidade- Os códigos QR tem capacidades de correção de erro, sendo que é possível restaurar informação de um código que esteja parcialmente sujo ou danificado (como um pequeno rasgão). Um código QR tem um nível de correção de cerca de 30 %, ou seja, se o dano ou a sujidade forem superiores nem toda a informação pode ser recuperada. Mais à frente, na secção 2.4.3, é abordada a correção de dados, tratando-se de uma funcionalidade um pouco mais avançada.

Leitura a partir de qualquer direção – Os códigos QR são capazes de ser lidos de todas as direções, ou seja, 360°. Isto é possível dado a estrutura do mesmo, este contém nos três cantos padrões que permitem detetar a posição, sendo que são assim também garantidas leituras de alta velocidade. Mais à frente, na secção 2.4.4, irão ser evidenciados estes três pontos e a estrutura do código [31].

Estrutura de ligação entre códigos QR - Um único código QR pode ser dividido em muitos (informação espalhada), em contraste muitos códigos QR diferentes podem ser reconstruídos para criar um único. De referir que esta divisão pode ser feita até um total de 16 códigos. Esta funcionalidade é útil por exemplo quando existe necessidade de fazer impressão do próprio código QR numa secção estreita, desta maneira a divisão iria permitir partir a informação em códigos diferentes que cabem nessa secção [31].

2.4.3 Correção de erros (*Error Correction*)

Apesar de ser também uma funcionalidade do código QR, esta é aquela que envolve mais complexidade e, portanto, uma observação de funcionamento e comportamento mais ao detalhe. Como já referido anteriormente, o código QR consegue restaurar informação quando o mesmo se encontra sujo ou, de alguma maneira, danificado. Nesta secção apresentam-se os processos envolvidos nessa correção e os níveis de correção.

Existem 4 níveis diferentes de correção disponíveis, sendo eles o nível L, M, Q e H (Tabela 2). Quanto maior for o nível escolhido maior será a capacidade de correção, no entanto será também maior a informação a guardar no código QR. A escolha do tipo de correção irá depender muito do seu utilizador e do tipo de ambiente em que o mesmo estará inserido. Se o mesmo estiver num local onde se poderá sujar facilmente possivelmente deverá optar por um nível Q ou H, enquanto que num ambiente mais limpo e pouco propício a sujidade o nível mínimo L poderá ser escolhido [32].

Na Tabela 2, podem-se observar os níveis e a percentagem, aproximada, de correção correspondente.

Tabela 2- Capacidade de correção de erros de um código QR [32][33]

Nível de correção	Capacidade em %
L	Cerca de 7%
M	Cerca de 15%
Q	Cerca de 25%
H	Cerca de 30%

É necessário entender e rever alguns conceitos para perceber como funciona esta correção. Este tipo de correção é implementado através do uso do código de Reed-Solomon e de palavras de código (*codewords*) na informação a guardar. Palavras de código (*codewords*) são códigos de informação e correção de erros que são guardados em partes de 8 bits, nas suas respetivas secções, sendo elas a secção de dados e de correção [34].

A capacidade de correção depende da quantidade de informação a ser corrigida, portanto se for necessário codificar 100 palavras de código (*codewords*), sendo que 50 precisam de correção, irão ser então necessárias 100 palavras de código de Reed-Solomon. Podemos observar então que este código de Reed necessita sempre do dobro da quantidade de palavras de código a ser corrigidas. Neste caso específico iremos então ter no total 200 palavras de código, sendo que 50 podem ser corrigidas. Se for efetuado o enquadramento com aquilo que foi observado anteriormente consegue-se perceber que é necessário um rácio de percentagem de correção de 25% que corresponde ao nível Q [32].

2.4.4 Estrutura do código QR

Um código QR é símbolo de matriz que consiste num vetor organizado de pequenos módulos/secções quadrada, que por sua vez se encontram dentro de um padrão em formato quadrado [35]. Existem vários artigos que demonstram como se denominam, onde se encontram e como estão organizados estes módulos/secções, mas nenhum deles o faz de forma completa. De seguida são evidenciadas definições recolhidas de várias dessas fontes pois não existe nenhuma fonte totalmente completa. Em adição, e para uma melhor identificação e visualização dessas secções, uma adaptação de várias imagens encontradas foi efetuada (Figura 2). É também importante referir que esta estrutura só é viável a partir da versão 2, pois só partir daqui é que alguns módulos/secções foram introduzidos. A adaptação (Figura 2), encontra-se abaixo da descrição e numerada a par dos elementos, para que se possa identificar facilmente cada componente na imagem da sua estrutura.

1. O primeiro componente da estrutura é o padrão de localização (**Finder Pattern**). Através deste padrão específico, localizado em três dos quatros cantos que um código QR possui, é possível facilitar e melhorar a deteção do posicionamento do código QR. Deste maneira irá ser muito fácil detetar a posição, o angulo do qual se está a digitalizar o código e o tamanho do mesmo [36].
2. Em segundo aparecem os separadores (**Separators**). Estes espaços em branco separadores ocupam cerca de 1 pixel de largura e tem como funcionalidade melhorar o padrão de localização pois separam os mesmos da informação propriamente dita [29].
3. Em terceiro ilustra-se o padrão de tempo (**Timing Pattern**) Este padrão tem como funcionalidade determinar a coordenada central de cada célula através de padrões pretos e brancos devidamente agrupados de forma alternada. Está organizado nas direções horizontais e verticais e quando existe um símbolo distorcido o mesmo consegue proceder à correção dessa mesma coordenada central da célula de dados [37].
4. Outro componente da estrutura é o padrão de alinhamento (**Alignment Pattern**). Este padrão, que apenas se encontra disponível a partir da versão 2, e acima do código QR (tema abordado na seção 2.4.5), é usado para restaurar distorções presentes no código. Através de uma pequena célula preta e independente presente neste padrão é possível identificar qualquer tipo de distorção não linear [36].
5. O quinto componente refere-se ao formato de informação (**Format Information**). É nesta secção, que se encontra mesmo ao lado dos separadores, que é guardada toda a informação acerca do nível de correção e o tipo de padrão de *masking* (processo de escolha da cor dos módulos, de forma a tornar o código perceptível) do código QR. Esta secção consiste em 15 bits [29].

6. Este componente consiste na secção de dados (**Data Area**). É nesta área que toda informação do código QR vai ser guardada. A informação utilizada neste código é codificada através números binários (0 e 1) tendo como base a codificação de códigos de Reed-Solomon (RS) [37].
7. Esta secção é denominada de correção de erros (**Error Correction**). A mesma é muito similar à secção de dados/informação e é aqui que os códigos de correção de erros são guardados em palavras de código (**codewords**) de 8 bits de comprimento [29].
8. Temos a secção de bits restantes (**Remainder Bits**). Consiste numa secção vazia de bits, se os bits dos dados e da correção de erro não forem possíveis de dividir em palavras de código (**codewords**) de 8 bits sem resto [29].
9. Existe por fim uma secção externa ao código QR, mas que de certa maneira faz parte do mesmo, denominada de zona sem ruído ou zona de sossego (**Quiet Zone**). Esta é uma simples margem de espaço “vazio” à volta do código QR de forma a facilitar a sua leitura. Esta zona possibilita uma melhor deteção do símbolo através da imagem lida pelos sensores, normalmente são necessárias 4 ou mais células para se fazer a leitura de forma efetiva e correta [37].

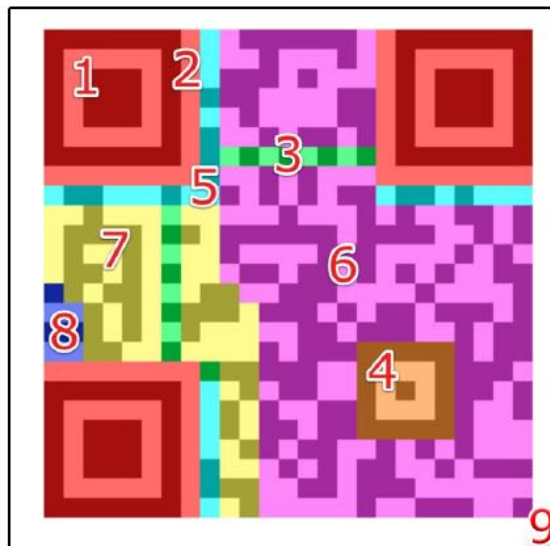


Figura 2 – Estrutura completa de um código QR (Editada e adaptada) [29][38]

2.4.5 Versões do código QR

Existem diferentes versões de códigos QR. O fator principal que as difere é a capacidade de guardar informação. Pode-se então afirmar que a capacidade de guardar dados, algo já referido, esta diretamente ligada à versão do código, ao seu nível de correção e ao tipo dos dados. A versão do código QR é então definida num intervalo de 1 a 40. Em cada incremento de versão a capacidade de guardar dados vai aumentando exponencialmente. Sabe-se então que cada versão tem um número e uma configuração diferente dos seus módulos (módulos

são os pontos brancos e pretos que verdadeiramente definem o código em si). A versão 1 do código QR é composta por módulos de 21x21, onde se consegue codificar cerca de 25 caracteres alfanuméricos, a versão 2 já é composta por módulos de 25x25, conseguindo codificar até 47 caracteres alfanuméricos. A última versão disponível, a versão 40, consiste em módulos de 177x177, conseguindo-se codificar até 4296 caracteres alfanuméricos.

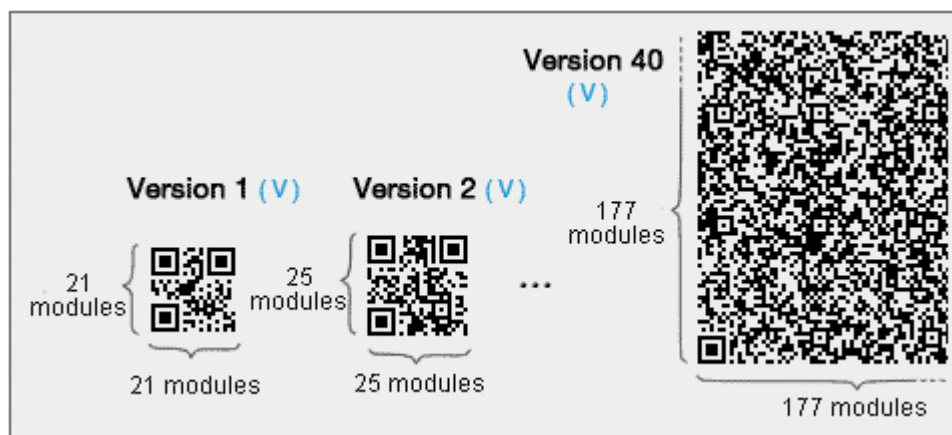


Figura 3 - Comparação entre a versão 1 e a versão 40 de um Código QR [39]

Pode-se concluir que se a versão incrementar em 1, o número de módulos aumenta em 4x4 e que para cada versão (1-40) existem, como já abordado anteriormente, os 4 tipos de correção. Ao se escolher a versão é importante escolher o nível de correção pretendido, pois quanto maior for o nível de correção menos capacidade de codificação irá ter o código QR. Um limite prático de tamanho de um código QR irá ser sempre dependente da câmara usada para captar/digitalizar o código, pois as resoluções e capacidades de focagem irão influenciar essa leitura [40][41][42]. É possível determinar ainda a versão de um código a ser utilizada [43].

2.5 Tipos de códigos QR

Os códigos QR podem ser de diversos tipos. Existem desde os códigos tradicionais (que são aqueles mais comuns) constituídos por uma forma retangular de “pontos” brancos e pretos até aos códigos com imagens embutidas. Nesta secção são evidenciados alguns dos tipos de códigos mais importantes, explicando-se as suas diferenças, vantagens e desvantagens. No entanto existem outros tipos de código [44][45][46][47][48][49].

2.5.1 Código QR (comum) Modelo 1 e 2 (*Model 1 and 2*)

Os códigos QR de modelo 1, muitas vezes denominados de originais, são diferentes pois continham uma capacidade máxima de codificação baixa (versão 1-14) e porque, principalmente em comparação com os outros modelos, não continham qualquer tipo de padrão de alinhamento na sua estrutura. Os códigos QR de modelo 2 já possuem padrões de

alinhamento e representam a estrutura completa do código QR (presente na seção 2.4.4). Os códigos do modelo 2 apresentam melhorias na suavização da leitura, aumento de capacidade de codificação e um maior número de versões disponíveis (1-40) [50][51].

2.5.2 Código QR micro (*Micro QR Code*)

Os códigos QR micro (em comparação com a versão comum do código QR) caracterizam-se por serem mais pequenos, terem um tamanho de impressão mais baixo e por conterem apenas um padrão de deteção de orientação (Figura 4) [52].

Este código possui 4 variantes/versões que vão desde o M1 até ao M4 com capacidade de codificação também variante. A versão M4, a maior de todas as anteriores, possui como capacidade máxima 35 valores numéricos e é composta por módulos de 17 por 17 [53]. Este tipo de código QR obteve também o seu padrão atribuído e aprovado em novembro de 2004 como um padrão JIS (JIS X 0510) [54].

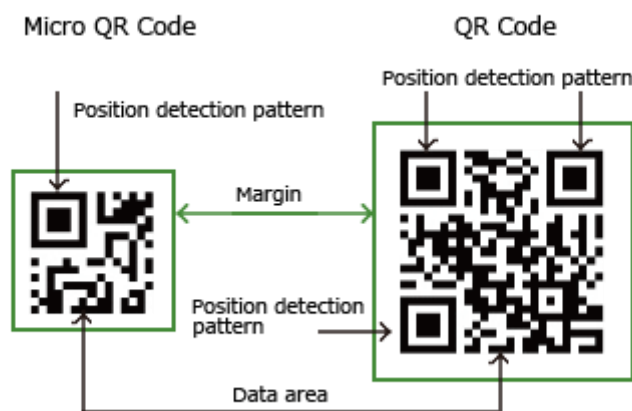


Figura 4 – Comparação entre o código micro (Micro QR Code) e o QR Code “normal” [55]

De referir que a margem envolvente deste código conhecida como **Quiet Zone** (zona sem ruído ou zona de sossego) é composta por apenas dois módulos de largura em comparação aos quatro da versão comum do código QR [55].

2.5.3 Código QR com moldura (*Frame QR*)

Este tipo de código destaca-se da versão comum por possuir uma área onde é possível colocar uma imagem, gráficos e outro tipo de informação. A informação encontra-se centrada no código e pode ter diversas formas e cores permitindo uma identificação visual mais fácil (Figura 5). O conteúdo adicionado ao centro não altera a quantidade de informação a guardar mas pode interferir na leitura se tiver um tamanho elevado [56].

Este tipo de código é muito utilizado para fins promocionais e publicitários pela sua fácil identificação visual perante a versão comum, apesar de poder ter outro tipo de fins.



Figura 5 – Exemplos de códigos QR do tipo Frame (imagem editada e adaptada) [57][56]

2.6 Casos de uso de códigos QR

Os códigos QR tem cada vez mais um papel fundamental no dia-a-dia, os mesmos contribuem para uma maior facilidade em algumas tarefas a realizar dentro e fora de casa. Atualmente, os códigos QR podem ser utilizados em cartões, folhetos, sistemas de pagamento e entre outros. São também utilizados muito em fábricas quer para promoção/informação dos seus produtos mas também em operações internas de logística [58].

De seguida, irão ser evidenciados alguns casos de uso destes códigos em diversas áreas e tipos de negócio. Em muitos dos casos evidenciados apenas se tem acesso a informação externa a empresa, portanto apenas serão identificadas as diversas características dos códigos usados e que funcionalidade/propósito servem os mesmos.

2.6.1 McDonald's

Os produtos vendidos pela McDonald's na sua grande maioria possuem códigos QR. No entanto, os códigos impressos são todos do tipo comum. Estes apenas possuem cores diferentes (inversão de cores) (Figura 6).



Figura 6 - Exemplo de um produto McDonald's

Quanto a funcionalidade muitos dos códigos que foram digitalizados redirecionavam para um URL de erro, outros para uma página do site da McDonald's ("<https://www.mcdonalds.pt/produtos>") onde são apresentados todos os produtos da marca.

A marca utiliza também códigos QR sem qualquer tipo de personalização na sua aplicação móvel como forma de autenticação no sistema de pontos MLovers, um sistema que atribui pontos por cada compra ou visita a um estabelecimento da marca.

2.6.2 Amazon

A Amazon é conhecida por ser uma das maiores retalhistas online em todo o mundo. A Amazon utiliza códigos QR e em 2018 inventou o seu próprio estilo de código QR. A companhia lançou a sua própria versão destes códigos denominada SmileCodes, facilitando o acesso dos seus consumidores a presentes e ofertas especiais. Ao contrário dos códigos QR normais estes contêm um "sorriso" posicionado no centro. Esta customização permite que os consumidores associem rapidamente o código a marca. Para além do design único, estes códigos especiais apenas podem ser lidos pela própria aplicação da Amazon (Figura 7). Esta oferece segurança redobrada e exclusividade aos clientes da marca e faz com que as pessoas efetuem o download da aplicação [59].

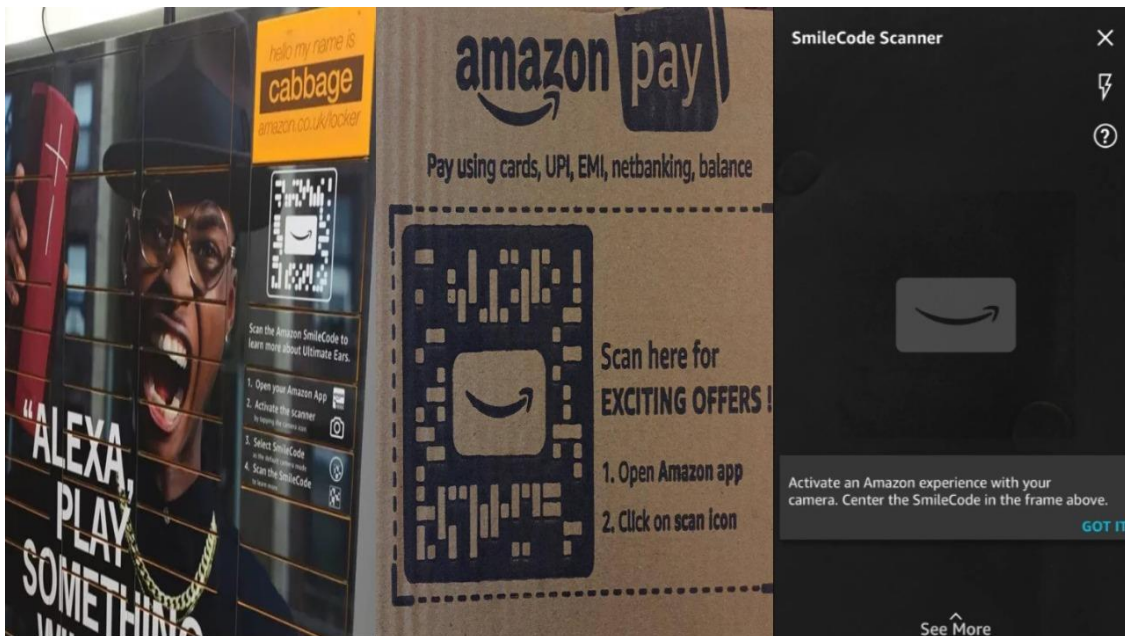


Figura 7 – Imagem da publicidade em cacifos, pacotes de entrega e aplicação de SmileCodes[60][61]

De acordo com a empresa estes códigos demonstram benefícios na localização onde são encontrados. Se uma pessoa passar por um cacifo da Amazon poderá encontrar no mesmo um SmileCode, que irá permitir abrir o cacifo e obter algum tipo de presente ou desconto promocional. Estes códigos começam também a ser aplicados nos pacotes de distribuição da empresa [60][61].

2.6.3 Super Bock

A Super Bock é uma marca de cerveja portuguesa. A marca apareceu em 1927 pela empresa Unicer e destaca-se pela sua variedade e qualidade dos produtos. A mesma evidencia-se ainda pela sua atenção dedicada as necessidades dos seus consumidores, optando por grandes estratégias de comunicação [62].

A Super Bock utiliza códigos QR como uma forma de comunicação e marketing. Se for observada uma cerveja da marca, é possível identificar que a mesma contém impresso no rótulo traseiro um código QR (Figura 8). Assim, é possível identificar um conjunto de características dos códigos utilizados pela marca, sendo elas: a utilização de código do tipo frame (pois o mesmo contém o logotipo da marca centrado); permitem também a modificação da forma dos módulos interiores (sendo que os códigos criados substituem o formato retangular normal, por módulos em formato oval e oval em forma de estrela; por fim é possível observar que os três módulos utilizados para o padrão de localização utilizam um formato sem ser o “normal”, substituindo os quadrados “normais” por quadrados com as pontas em formato mais oval.



Figura 8 - Imagem de uma cerveja Super Bock com um Código QR

Quanto a funcionalidade, a marca utiliza os códigos não para informar mais acerca do produto, mas sim para promover um festival realizado pela marca, neste caso o Super Bock Super Rock. Ao ser efetuado uma digitalização ao código obtemos um URL para uma página estática ("<https://www.sbsr.fm/playlists/>"), que redireciona o utilizador para uma playlist de músicas dos artistas/músicos que irão estar presentes no festival. Ou seja, a empresa utiliza os códigos nas suas cervejas como uma estratégia de promoção de marketing.

2.6.4 Medicamentos (Laboratórios)

Os medicamentos que se utilizam no dia a dia, ou em situações de emergência, fazem também utilização destes códigos QR. A utilização ou não destes códigos é decidida pelos laboratórios que fabricam cada medicamento.

No entanto, existe algo em comum em todos os medicamentos analisados. Todos fazem utilização de códigos do tipo iQR. A utilização deste tipo de código é especialmente útil pois estes podem ter tamanhos ainda mais pequenos e ter um nível de correção maior que o código QR micro. Estas características são especialmente úteis em caixas de medicamento, pois o espaço nas caixas é reduzido e estão sujeitos a rasgões na sua abertura.

Um exemplo de laboratório que utiliza códigos é a Generis, conhecida por fabricar medicamentos genéricos de marcas originais. A marca adotou utilizar este código recentemente. Numa amostra de 4 caixas de medicamentos apenas uma não continha código, pois tinha sido fabricado há mais tempo. De referir, dos medicamentos observados de outros laboratórios, que um dos medicamentos fazia uso da opção de formato retangular do código iQR, como pode ser observado na Figura 9.



Figura 9 - Medicamentos Generis a esquerda, a direita medicamentos de outros laboratórios

Quanto a funcionalidade dos códigos quando digitalizados, todas as amostras demonstravam textos idênticos. São apresentados alguns exemplos desses códigos digitalizados: 0105099151904454212W9C46H2MK714532191417220228108088642, e 010358291009132517220930109A01021100A25GG013KVC7145475090. Observa-se que os códigos contêm caracteres que não conseguiram ser lidos corretamente e são compostos por um conjunto de números/códigos incompreensíveis ao ser humano. Ou seja, estes códigos são provavelmente usados internamente nos laboratórios ou uso interno nas farmácias, onde o conjunto de números deverá ser interpretado pelos seus sistemas. Pode-se concluir que estes códigos, à primeira vista, não terão qualquer utilidade para o utilizador final.

2.7 Plataformas/software de Códigos QR

Um dos aspetos mais importantes de qualquer tipo de software/plataforma é perceber a utilidade do mesmo e se faz sentido o seu desenvolvimento. É necessário fazer um pequeno reconhecimento daquilo que existe e não existe no mercado. Nesta secção é efetuada uma apresentação e análise das características das soluções apresentadas no contexto, problema e objetivos da dissertação.

Antes de demonstrar tecnologias/plataformas específicas é importante referir os resultados obtidos quando se efetua uma pesquisa no google por softwares de geração de códigos QR. Foram efetuadas duas pesquisas diferentes com as seguintes frases: “qr code generator” e “qr code”. Em ambas as pesquisas efetuadas os resultados das primeiras páginas foram idênticos e evidenciam que a maior parte dos softwares encontrados são gratuitos e que servem para uma rápida criação de um código QR comum. Pode observar-se também que, muitas vezes,

estes softwares são cópias idênticas uns dos outros. Importa referir que esta pesquisa foi efetuada no último dia de 2019.

A abundância de software gratuito e idêntico já existia em 2014, onde Sousa efetuou um estudo de aplicações existentes em duas das maiores lojas móveis da atualidade: a AppStore e a GooglePlay [63]. Neste estudo evidencia-se o facto de existirem cerca de 242 aplicações gratuitas contra 8 pagas na GooglePlay. Na AppStore existiam cerca de 11 aplicações gratuitas para 34 pagas. No geral, e como já referido, este é um cenário onde a maioria das aplicações é de teor gratuito e apenas contém funcionalidades básicas [63].

2.7.1 The QR Code Generator

Uma aplicação/plataforma de teor gratuito muito básica que apenas permite gerar códigos QR, onde até o próprio nome é genérico. No geral, esta aplicação apenas permite gerar códigos QR básicos. A mesma contém poucas opções acerca da carga/tipo de informação (Ex: texto, contacto, url e sms). Este software é minimalista contendo poucas opções, sendo que tem um único ponto forte e diferenciador[64]:

- Permite digitalizar códigos através da webcam

2.7.2 QR Code Generator

Esta aplicação contém duas vertentes, uma versão gratuita e uma versão paga, e serve para gerar códigos. Ao contrário da anterior, já contém mais algumas opções e funcionalidades. Nesta aplicação também são apresentadas algumas opções para a carga/tipo de informação a ser guardada oferecendo um leque mais vasto em comparação com a opção anterior. Abaixo são evidenciados pontos fortes deste software[65]:

- Diferentes opções de carga/tipo informação (URL, SMS, vídeos de youtube, wifi, mp3, pdf e entre outros)
- Possibilidade de criar códigos dinâmicos (mudar informação guardada sem criar novo código). De referir que estas funcionalidades necessitam sempre do login por parte do utilizador
- Mudar formato dos módulos interiores do código QR
- Possibilidade de adicionar um logotipo (FrameQR)
- Permite modificar/inversão de cores. Ou seja, permite mudar da cor preto para, por exemplo, verde. Tem disponíveis apenas 6 cores diferentes.

- Versão paga com pacotes diversos contendo funcionalidades extra, como estatísticas detalhadas e mais opções de cores.

2.7.3 QR Code Monkey

De todos os softwares gratuitos considera-se este o mais completo. O QR Code Monkey oferece a maior parte das funcionalidades anteriores e ainda algumas únicas (não disponíveis noutras soluções). Tal como nas tecnologias e softwares anteriores apresentam-se, de seguida, as principais características e pontos fortes deste software[66]:

- Opções de codificação do tipo de informação variadas e algumas únicas para um software gratuito (Facebook, bitcoin e entre outros), chegando a 17 tipos diferentes.
- Permite mudar o formato dos módulos interiores do código QR, sendo dos softwares com vertente gratuita apresentados aquele que mais opções contém (cerca de 22 opções diferentes).
- Permite mudar o formato dos limites do código, oferecendo diversas opções (Ex: redondo, oval, em bolas e entre outros).
- Permite mudar o formato específico dos padrões de deteção alinhamento.
- Possibilidade de adicionar um logotipo (FrameQR).
- Permite modificar/inversão de cores.
- Possibilidade de redimensionar os códigos QR.
- Possibilidade de escolher, dentro de um determinado alcance, a qualidade da imagem gerada pela plataforma.

2.7.4 QR Code Studio

Esta é a versão premium do software QR Code Monkey. Comparativamente à versão anterior, esta apenas adiciona a capacidade de gerar códigos dinâmicos (“alterados depois de ser gerados”) e um espaço de gestão dos códigos.

Um código diz-se dinâmico quando é gravado no mesmo um URL. O conteúdo de um URL pode ser mudado a qualquer momento, no entanto o URL gravado no código irá ser sempre o mesmo.

2.7.5 Visualead

Este software contém duas vertentes sendo que dá mais ênfase a sua versão paga, apesar de existir uma versão gratuita extremamente limitada, mas que promete aos seus utilizadores criar códigos QR em 4 passos simples [67].

De uma forma geral, é possível observar que o mesmo tem como maior força a promoção de geração de códigos QR com incorporação de imagens visuais no mesmo (FrameQR). A empresa promove a solução como a escolha ideal para marcas de negócio. Tal como nas tecnologias e softwares anteriores apresentam-se, de seguida, as principais características e pontos fortes deste software [67]:

- Criação de códigos do tipo FrameQR. Este tipo de código é visualmente mais apelativo e permite uma fácil identificação perante vários códigos existentes.
- Permite escolher a imagem partir de alguns *templates* existentes e criados pela comunidade ou então permite ao utilizador fazer *upload* da sua própria imagem.
- Permite modificar/inversão de cores.
- Permite rodar/inverter o padrão de deteção de alinhamento. A ferramenta permite rodar estes “quadrados” da estrutura do código QR perante a imagem escolhida.
- Permite colocar os padrões de deteção de alinhamento de forma arredondada e mudar a forma entre 3 à escolha (pontos, quadrados, estrelas) de cada módulo constituinte do código QR.
- Permite estatísticas analíticas (Ex: quantas pessoas digitalizaram um código)
- A informação contida pode ser dinâmica
- Espaço próprio para gerir códigos criados

2.7.6 Análise e comparação das plataformas

Após serem referidas as características principais que constituem cada solução, é importante colocar os mesmos em comparação entre si e em comparação com os objetivos da solução a desenvolver. Desta maneira, é perceptível aquilo que os distingue e aquilo em que são iguais. Foi elaborada então uma tabela de comparação (Tabela 3) para uma melhor visualização destes pontos.

Tabela 3 - Tabela Comparativa de softwares de códigos QR

Plataforma/ Funcionalidades	The QR Code Generator	QR Code Generator	QR Code Monkey	QR Code Studio	Visualead	Software a desenvolver
Criar códigos QR	✓	✓	✓	✓	✓	✓
Digitalizar códigos QR	✓	✗	✗	✗	✗	✗
Tipo de codificação *	✓	✓	✓	✓	✓	✓
Tipos de código	✗	✓	✓	✓	✓	✓
Escolha de colocar/ retirar <i>Quiet Zone</i> (Zona sem ruído)	✗	✗	✗	✗	✗	✓
Inversão de cores do código	✗	✓ / ✗	✓	✓	✓ / ✗	✓
Redimensionar códigos	✗	✗	✓	✓	✓	✓
Códigos dinâmicos	✗	✗	✗	✓	✓	✓
Forma dos módulos interiores	✗	✗	✓	✓	✓	✓
Inversão de posição do código	✗	✗	✗	✗	✓	✓
Espaço de gestão de códigos	✗	✗	✗	✓	✓	✓
Diferentes formatos de download de ficheiro	✗	✓	✓	✓	✓	✓
Estatísticas analíticas	✗	✓	✓	✓	✓	✓
Aviso do limite máximo de carateres	✗	✗	✗	✗	✗	✓
Landing pages personalizadas	✗	✗	✗	✗	✗	✓
Escolha de nível de correção	✗	✗	✗	✗	✗	✓
Versão do código a ser utilizada	✗	✗	✗	✗	✗	✓

* De referir que só se pode gravar informação nos códigos QR em forma de texto. A maior parte das opções de codificação/gravação apresentadas nos softwares anteriores, são apenas “mascaras” e truques usados para desencadear um conjunto de ações quando um utilizador digitaliza um código QR. Para ser mais perceptível o comentário anterior é possível ilustrar com um exemplo. Quando se escolhe nestes websites a opção de codificação através de URL, o mesmo apenas adiciona ao texto existente um prefixo http ou https. Depois desta ação o tipo de dados continua a ser texto.

Observando a tabela é possível retirar algumas conclusões. Primeiro o facto de existirem diversas funcionalidades fulcrais que deveriam estar presentes numa solução deste género e que não se encontram disponíveis em nenhum. Por exemplo, nenhuma solução permite escolher entre diferentes níveis de correção de um código QR, nem informam se a escolha existe. Não existe indicação da versão do código a ser utilizado (diferentes quantidades de espaço e níveis de correção ocupam espaços diferentes). Identificou-se ainda a falta de informação do limite/quantidade de dados possível de armazenar, o que poderá resultar num código em que é impossível efetuar a leitura.

De referir que muitas destas soluções são gratuitas, mas também muito limitadas. No entanto, quase todas as encontradas oferecem vertentes pagas que também em si limitam muito a utilização pois impõem limites do número de códigos QR a ser gerados e também limitações ao número de leituras.

Nenhuma das soluções anteriores inclui *Landing Pages* personalizáveis que são dos pontos fulcrais e de grande valor acrescentado da solução a desenvolver. A solução a desenvolver pretende-se mais completa em termos de personalização de códigos QR em comparação com as soluções existentes. Neste aspeto, um dos objetivos é permitir a configuração da forma como é efetuado o display da informação guardada no código. Quando a digitalização do código é efetuada, na maioria dos casos evidenciados e outras aplicações do mercado, a informação não recebe qualquer tipo de tratamento, sendo apenas redirecionado o utilizador para *urls* estáticos previamente definidos ou até apresentando apenas texto simples.

Estes são apenas alguns dos pontos que necessitam de ser incorporados e pensados para desenvolver uma solução melhor do que as já existentes no mercado. De referir que estas soluções já existentes são colocadas em comparação no capítulo de análise valor utilizando a metodologia QFD, para que seja possível ter um melhor entendimento das necessidades dos utilizadores, com as funcionalidades já existentes e aquelas que necessitam de ser implementadas.

2.8 Alternativas de Frameworks ao desenvolvimento

Antes de se proceder ao desenvolvimento de uma solução é necessário fazer uma análise das ferramentas a ser utilizadas. Nesta secção faz-se uma apresentação de 3 alternativas em termos de *Frameworks* base para suportar o desenvolvimento da solução. Estas foram as

alternativas que foram sugeridas e apresentadas pela empresa na discussão inicial de ideias sobre o desenvolvimento da solução. A ferramenta deverá permitir desenvolver uma solução que: consista numa aplicação web, permita interligação a base de dados e sistemas externos, possibilite gerar e manipular códigos QR e Landing Pages.

No capítulo de Análise de valor (seção 3), através do método AHP, irá ser determinada qual a ferramenta mais adequada a utilizar para o desenvolvimento da solução a preconizar.

2.8.1 Blazor

O Blazor é uma *framework* em C# com o objetivo principal de desenvolver e construir aplicações web para o cliente [68]. O mesmo possui uma série de funcionalidades que são descritas de seguida:

- Permite o desenvolvimento de interfaces web utilizando C#, HTML e CSS [69]. É uma alternativa à maior parte das *frameworks* atualmente no mercado que utilizam JavaScript.
- Uma das melhores vantagens é que nesta *framework* poderemos partilhar a lógica entre o cliente e o servidor algo que não acontece com a maior parte das *frameworks* existente no mercado atual, onde necessitamos sempre de uma tecnologia/*framework* extra, para a parte do servidor, que pode ser numa linguagem diferente daquela a ser utilizada no cliente. Aqui temos a vantagem de ter tanto o cliente como o servidor em .NET [68][70].
- Possibilidade de correr o código do cliente diretamente no browser através de WebAssembly usando o modelo cliente-servidor (**Blazor WebAssembly**), tendo sempre a possibilidade de correr o código no servidor (**Blazor Server**) como é normalmente efetuado [69].
- O mesmo permite fazer interpolação (troca de informação entre linguagens) com o JavaScript. Apesar de umas das suas maiores vantagens ser unificar todo o código numa única linguagem, esta *framework* permite fazer chamadas a código em JavaScript [68][70]. Se já tivermos algo desenvolvido nesta linguagem temos a vantagem de não descartar por completo aquilo que já foi desenvolvido, no entanto, é importante referir que esta interpolação é algo complexo e deve ser evitada quando possível.

Uma das grandes desvantagens desta *framework* é que no momento da escrita desta tese, o modelo de servidor (**Blazor Server**), saiu de modo *preview* à relativamente pouco tempo e o modelo cliente-servidor (**Blazor WebAssembly**) encontra-se ainda em *preview* para a versão 3.1 do ASP.NET CORE [71]. Isto pode levar ao encontro de vários bugs e que certas funcionalidades não estejam presentes ou a funcionar corretamente, podendo também existir muitas coisas que ainda estejam parcialmente documentadas ou sem documentação de todo.

2.8.2 React

Apesar de muitas pessoas considerarem o React uma *framework*, o mesmo é na verdade uma biblioteca de JavaScript com o principal propósito de construir interfaces para os seus utilizadores. O mesmo possui diversas características e vantagens aqui enumeradas[72]:

- O mesmo funciona de maneira declarativa, ou seja, o código desenvolvido irá ser de fácil interpretação e irá ser mais fácil detetar bugs e erros.
- Utiliza componentes com o seu próprio estado, toda a lógica é escrita em JavaScript permitindo assim passar grandes quantidades de informação e deixar o estado fora do DOM.
- O mesmo permite utilizar o código quer no cliente quer no servidor, sendo que do lado do servidor é utilizado o node.

No entanto, o mesmo possui uma grande limitação apenas se foca na camada *View*, ou seja, de acordo com o modelo *MVC* necessitamos sempre de outras tecnologias para efetuar um desenvolvimento mais complexo (*Full Stack*) [73]. Para o *backend* pode ser utilizado simples código *c#* ou outra *framework* mais complexa.

2.8.3 Angular

Angular é uma *framework* de desenvolvimento que atualmente se encontra na versão 8, esta é utilizada na criação de aplicações HTML e JavaScript, a mesma é escrita/criada em TypeScript [74]. Para o *backend* pode ser utilizado simples código *c#* ou outra *framework* mais complexa, tal como na biblioteca anterior. Esta *framework* possui algumas características e vantagens aqui enumeradas[74][75][76]:

- Permite o desenvolvimento multiplataforma. Ou seja, permite desenvolvimento web, mobile web, mobile nativo e desktop nativo.
- Cada aplicação desenvolvida em Angular contém então um módulo (*root*), este módulo proporciona um mecanismo de *bootstrap* para lançar a aplicação. De referir que os módulos no Angular importam funcionalidades de outros módulos.
- Utiliza componentes que são definidos como classes normais e que contêm meta dados específicos para a sua identificação. Cada aplicação desenvolvida em Angular tem de ter no mínimo um componente (*root*) que conecta a hierarquia à DOM. Um componente, como referido anteriormente, é uma classe que irá conter os dados e a lógica.
- Utiliza *templates* para combinar HTML com a marcação Angular de forma a poder modificar elementos HTML antes de os demonstrar. Este *templates* proporcionam a

lógica e permitem ligar os conceitos de marcação com a informação e a DOM da aplicação.

- Permite a criação de serviços para lógica ou informação não associada com a *View*. Utiliza injeção de dependências para facilitar o processo de utilizar qualquer serviço criado em qualquer componente.
- O roteamento é efetuado através de um módulo existente (*NgModule*) por defeito, de forma a permitir ao utilizador fácil navegação através dos diferentes estados e vistas da aplicação.

2.9 Visão da Solução

Após efetuado o estado da arte foram identificadas diversas características dos códigos QR, bem como casos de uso e soluções existentes. Os objetivos inicialmente definidos foram interpretados no contexto dos conhecimentos obtidos no estado da arte e refinados de forma a alcançar a visão pretendida com esta solução. Conseguiu-se assim uma melhor reflexão sobre as necessidades de desenvolver uma solução deste género.

A solução a desenvolver poderá então ser considerada um sistema ciber-físico, pois enquadra-se em todas as dimensões que um sistema CPS contém. A solução irá ter uma enorme cooperação com o ser humano, aliás sem esta cooperação a mesma não consegue prosseguir e deixa de ter sentido, pois o ato de digitalizar é efetuado pelo mesmo. Outra dimensão que importa considerar é o facto de a plataforma ser dinâmica e ter campos/informações que se atualizam em tempo útil para o seu utilizador. É possível fazer esta observação após o estudo efetuado na secção Indústria 4.0, onde foram identificados sistemas do género.

Esta solução deverá inovar no sentido em que será uma solução de uso comercial cujo objetivo é permitir que o utilizador obtenha informações de produtos desde a produção até à entrega, ou para a promover outros produtos existentes da marca/empresa.

Em termos de dados, a solução deve permitir conter o máximo de informação possível no próprio código QR, de forma a aliviar a carga e custo de armazenamento em base de dados para a empresa. No entanto, deve sempre informar o cliente dos limites da tecnologia (Ex: capacidade máxima de dados a gravar). Deve permitir ao cliente estar informado ao detalhe das circunstâncias da tecnologia que esta a usar (Ex: versão do código gerado). O estudo que permitiu verificar que os pontos anteriores são de valor para a solução foi efetuado na secção Códigos QR (QR CODES).

Em termos de personalização, a solução deverá permitir personalização total da apresentação final do conteúdo presente no código QR, utilizando *landing pages* (por exemplo determinar o número de campos para apresentar a informação ou a ordem com que a mesma aparece na página, através das abordagens e plataformas estudadas na secção 2.2). Deve também

possibilitar a personalização do próprio código QR, como por exemplo: fornecer diferentes tipos de códigos (Ex: FrameQR); permitir inversão de cores (mudar cor dos módulos); permitir inversão dos formatos dos módulos constituintes dos códigos QR; permitir adição/remoção da *Quiet Zone*; permitir o redimensionamento e inversão da posição dos códigos. Tal como nos casos anteriores, estes fatores de personalização foram identificados através do estudo efetuado nas secções de Códigos QR (QR CODES) e Tipos de códigos QR.

A solução deve ter em mente o cliente. O cliente deverá poder escolher o nível de correção para a informação guardada em cada código e ter ainda um espaço de gestão das *landing pages* e códigos criados, permitindo a observação de estatísticas analíticas sobre a utilização dos códigos e serviço. O estudo aos “Casos de uso” e das “Plataformas/software” existentes no mercado, permitiu perceber que os pontos evidenciados anteriormente iriam trazer um grande valor acrescentado a solução.

Por fim, a análise às alternativas de *frameworks* que foram sugeridas e apresentadas pela empresa permitiu identificar pontos fortes e fracos das mesmas. Através desta análise tornou-se mais fácil escolher aquela que mais se adequava para o desenvolvimento a solução.

3 Análise de valor

Neste capítulo irá ser abordada a análise valor evidenciado alguns conceitos acerca da mesma, demonstrando também alguns modelos existentes e aplicando algumas técnicas de análise. Todas os métodos selecionados são aqueles que são mais adequadas ao tema e ao software/plataforma a ser desenvolvido no âmbito desta tese. Por fim é efetuado um modelo de negócio com todas as componentes importantes e relevantes para este software/plataforma, incluindo a construção na forma de modelo Canvas.

3.1 Análise de valor - O que é e para que serve?

Quando se questiona alguém se um produto faz o que é suposto durante o tempo que é suposto, diz-se que o produto tem boa qualidade. Quando dizemos que algo faz, dura e custa o que é suposto então dizemos que o mesmo tem qualidade. Foram estas duas questões que o pioneiro da análise de valor Lawrence D. Miles colocou de forma a confirmar as técnicas de análise de valor e que se colocasse em perspectiva novas técnicas, onde se reduzem os custos e mantem-se a qualidade. O mesmo define a análise de valor como “uma filosofia que implementa um conjunto de técnicas, um corpo de conhecimento e um conjunto de habilidades adquiridas, sendo que tem como principal objetivo identificar de forma eficiente custos desnecessários (Ex: não proporciona qualidade, uso e etc.)” [77].

3.2 Fuzzy Front of Innovation e Modelo NCD

Inicialmente o modelo de inovação era conhecido por estar dividido em três partes, sendo elas o de *the fuzzy front end (FFE)*, o processo *the new product development (NPD)*, e a fase comercialização. Sendo que foi mais tarde atualizado nos termos e processos utilizados por especialistas na área [78][79].

Segundo estes autores especializados no tema, um processo de inovação encontra-se dividido por três partes. A primeira parte designa-se de Front End of Innovation (FEI), em segundo temos o New Product and Process Development (NPPD) e por fim temos as fases de comercialização. Inicialmente descrito por *fuzzy front end (FFE)*, este termo foi então renomeado para *Front End of Innovation (FEI)*, sendo que os autores acreditam que o primeiro termo fazia transparecer que o chamado *FFE* fosse caracterizado principalmente por fatores incontrolláveis e desconhecidos, dando uma perspectiva que a esta primeira fase deste modelo de inovação fosse impossível incutir algum tipo de gestão [80].

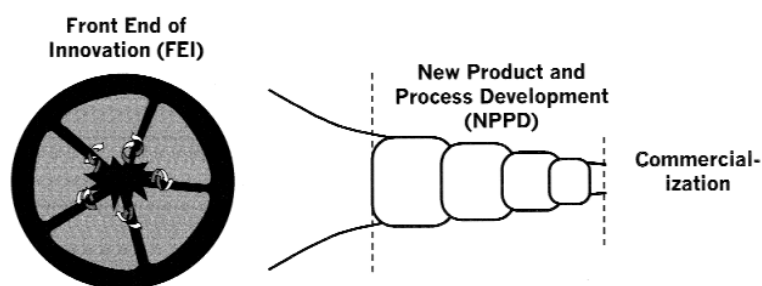


Figura 10 – Modelo de inovação [80]

Cada uma das partes evidenciadas anteriormente serve funções específicas. A primeira fase é onde é marcado por um pré trabalho que serve para descoberta de novas oportunidades. Na segunda fase é onde se efetua uma análise do marketing e de todos os pormenores técnicos que o produto poderá ter. Por fim, na última fase é construído um modelo de negócios onde se tem em conta todas as propriedades e trabalho efetuado nas fases anteriores [81].

Anteriormente no primeiro modelo era quase impossível comparar uma empresa de alguém com outra, pois não existia qualquer tipo de definição ou linguagem geral para definir todos os componentes do *front end*. Foi através desta necessidade que foi criado o modelo definido como *New Concept Development (NCD)*, promovendo aquilo que outrora faltava, uma linguagem e visão universal deste processo [80].

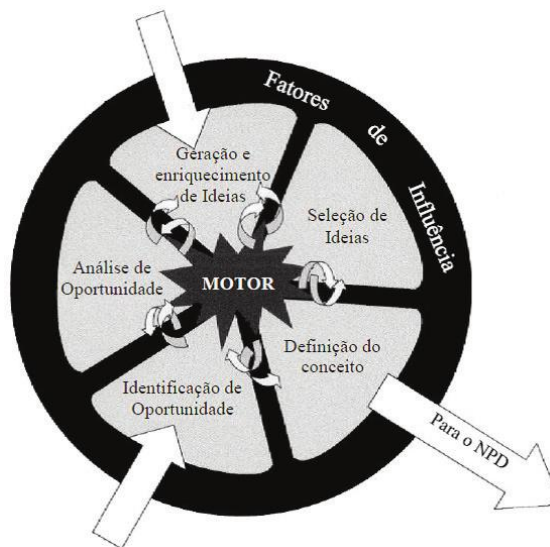


Figura 11 - Modelo New Concept Development [82]

Este modelo *NCD* consiste em três partes chaves[83]:

- A parte interior que consiste nos cinco elementos que compõem o *Front End of Innovation (FEI)*, elementos esses evidenciados na imagem anterior.
- O elemento central conhecido como motor que ira fazer os 5 elementos funcionar, este motor é alimentado pela cultura e liderança presente dentro do seio organizacional.
- O terceiro elemento consiste em fatores ambientais externos que iram influenciar o motor e por sua vez alterar os 5 elementos interiores.

Os autores deste modelo evidenciam, como já referido anteriormente na Figura 11, 5 elementos fundamentais. De seguida, vai se aplicar este modelo *NCD* e desenvolver os 5 elementos em relação produto/serviço a ser desenvolvido no âmbito desta dissertação.

Identificação de oportunidade – A falta de um software completo e personalizável de códigos QR, foi identificada pela empresa Ifthen e pelos seus clientes. A Ifthen ficou ciente desta lacuna no mercado após algumas conversações e trocas de ideias entre o seu grande número de clientes, que se queixavam da inexistência de softwares completos do tipo. Foi então efetuado um estudo ao mercado pela Ifthen e mais tarde um estudo pessoal para verificar em mais detalhe a situação

Análise de oportunidade – No estudo do mercado efetuado pela empresa e pelo estudo pessoal, foi detetado que a maior parte das grandes companhias que fazem uso de códigos QR, fazem-no num sistema fechado, ou seja, utilizam software privados e não divulgados para o público.

Os restantes softwares existentes na sua grande maioria são websites de teor gratuito, que apenas oferecem opções limitadas de customização e muito semelhantes entre todos. A grande lacuna identificada no mercado foi a falta de um software que oferecesse uma combinação completa de customização dos códigos QR, que fornecesse aos seus clientes “*landing pages*” personalizadas e visualização de dados analíticos

Geração e Enriquecimento de Ideias – Após a observação do mercado, foram discutidas e analisadas várias possibilidades de preencher as necessidades identificadas e encontradas anteriormente. Foram discutidas a possibilidade de existirem e serem desenvolvidas duas soluções diferentes mobile e web, mas ficou desde cedo bem definido que o software a ser desenvolvido seria de teor web, mas totalmente preparada para ser acedida por mobile.

A grande dúvida na geração e discussão de ideias seria que framework seria utilizada para o desenvolvimento. Foram identificadas e discutidas a possibilidade de usar entre três frameworks apresentadas anteriormente (Blazor, React, Angular).

Foi também evidenciado a ideia de estabelecer o software desenvolvido entre duas opções: um software de compra única ou através de um serviço de subscrição.

Seleção de Ideias – Depois de geradas as ideias, foi necessária selecionar aquelas que melhor irão cumprir com os objetivos definidos. Primeiro, para escolher entre as diversas frameworks propostas na geração de ideia foi utilizado uma metodologia de decisão multicritério discreta. A metodologia selecionada foi o método de análise hierárquica (*AHP- Analytic Hierarchy Process*), esta análise foi efetuada de maneira a entender qual das frameworks iria trazer mais valor a empresa. Nesse sentido, foram deliberados em conjunto com a empresa um conjunto de critérios para utilizar nesta metodologia de forma a efetuar a melhor escolha. Os critérios selecionados foram a funcionalidade, facilidade de utilização e a inovação.

Foi também definido que este software seria então disponibilizado através de subscrição.

Definição de Conceito - O objetivo deste projeto será então construir e desenvolver um software web disponibilizado para os clientes como um serviço. Os clientes poderão utilizar o serviço para gerar “*landing pages*” personalizadas através de códigos QR, oferecendo assim aos seus clientes páginas diferentes e devidamente personalizadas de acordo com o produto em questão. Para cada produto poderão optar por exemplo por códigos de cores diferentes, bem como outras opções. Serão também disponibilizadas aos clientes dados analíticos e estatísticos sobre a digitalização de um código. (Ex: número de scans de cada código, número de scans total)

A Ifthen terá então assim clientes diretos (aqueles que aderem ao serviço e vão gerar códigos para os seus produtos) e clientes indiretos (os clientes dos seus clientes, que irão ser todas as pessoas que irão digitalizar os códigos gerados pelos clientes diretos).

3.3 Proposta de valor

Uma proposta de valor é normalmente vista como todos os benefícios que uma organização consegue entregar a todos os elementos externos. Osterwalder e Pigneur defendem que uma proposta de valor é a forma como são agrupados, um conjunto de itens de valor (como produtos ou serviços), de maneira a que possa ser possível satisfazer as necessidades dos clientes. Uma proposta de valor serve para oferecer uma percepção aos clientes de como os mesmo irão conseguir retirar valor de um produto ou serviço disponibilizado (*perceived value*), esta é efetuado de forma diferente para cada um dos segmentos de clientes e é o fator de diferenciação entre os competidores existentes no mercado [84].

Para se obter uma boa proposta de valor, foi utilizado o Canvas de proposta de valor idealizado por Osterwalder (Figura 12). Este Canvas é constituído por 3 partes diferentes, primeiro temos o *Customer Profile* (perfil do cliente), onde é clarificado aquilo que é entendido ou percebido do cliente. A segunda parte, o *Value Map* (mapa de valor) é usado para descrever como se pretender criar valor para o cliente. Na parte central temos o *Fit* (encaixe), que simboliza o encontro das ambes as partes anteriores [85]. Aqui conseguimos também obter informações mais específicas sobre os segmentos de clientes escolhidos.

O *Customer Profile* (perfil do cliente) é constituído por os seguintes componentes:

- **Customer Jobs (Trabalhos dos clientes)** – Promover ou informar acerca dos seus produtos. Permitir acesso a informação de produtos em serviços de distribuição.
- **Customer Pains (Dores dos clientes)** – Dificuldade na entrega da informação de forma rápida, eficiente e eficaz. Custos elevados em desenvolvimento de softwares específico, parcerias, serviços de gestão de redes sociais usadas na promoção do produto. Falta de controlo do aspeto visual e a da apresentação da informação transmitida. Diferentes estruturas de dados para cada produto (visto que cada um tem diferentes características).
- **Customer Gains (Ganhos dos clientes)** – Ganhar mais clientes. Poupança de custos. Segurança. Diferentes tipos de autorização de acesso a informação. Dados analíticos do acesso à informação.

O *Value Map* (mapa de valor) é constituído por os seguintes componentes:

- **Products and Services (Produtos e Serviços)** – Serviço/Software de criação de códigos QR com landing pages, sendo ambos totalmente personalizáveis.
- **Pain Relievers (Analgésicos)** – Uso de códigos QR para facilitar o acesso, rapidez e entrega da informação (através de uma simples digitalização do código). Landing pages totalmente personalizáveis (permitindo remoção e adição de campos e diferente ordem dos mesmos). Fácil usabilidade (permitindo ao utilizador criar

códigos e páginas sem necessidade de fazer contratações extra). Customização do aspeto do código QR (oferecendo diferentes tipos, cores e entre outras opções).

- **Gain Creators (Ganhos criados)** - Escolha do nível de correção dos códigos QR (garantindo ao utilizador que a informação não é perdida caso algo acontece ao código). Dados analíticos (como por exemplo número de digitalizações). Diferentes níveis de acesso (sendo que informação sensível apenas será visível com login). Serviço de subscrição único e flexível (o mesmo serviço para a geração de códigos QR e criação de páginas, possuindo diferentes pacotes de adesão do tipo mensal, trimestral e anual).

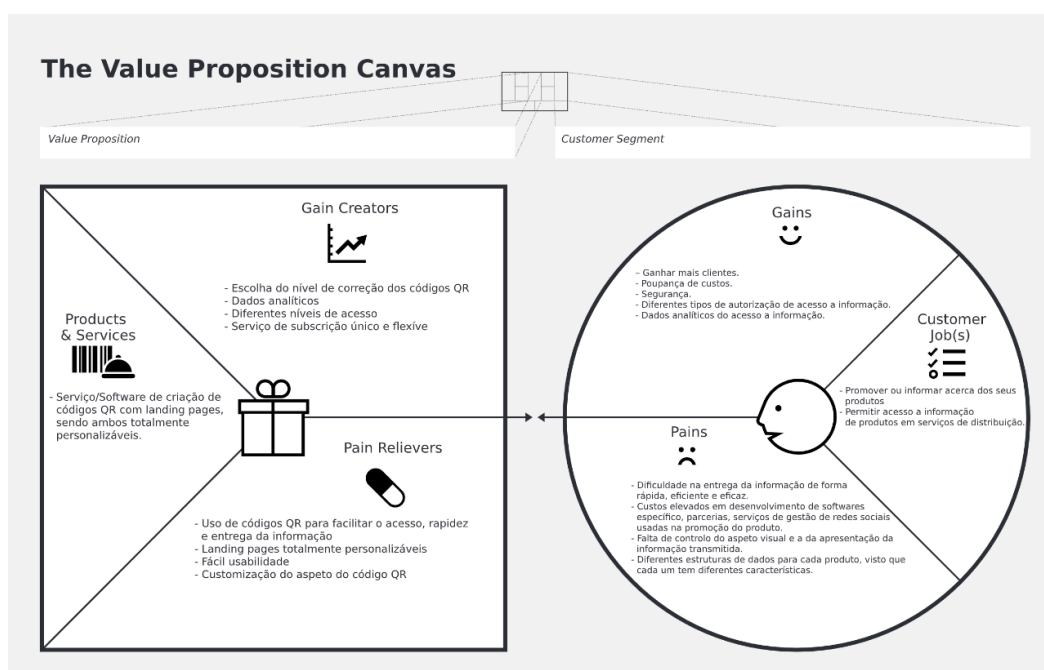


Figura 12 – Modelo Canvas da proposta de valor

A partir deste modelo foi possível identificar os segmentos do cliente e criar uma proposta de valor credível. No próximo capítulo, que aborda o Canvas do modelo de negócio irão ser evidenciadas ambos.

3.4 Modelo de Negócio CANVAS

O conceito conhecido por modelo de negócio apareceu originalmente por Peter Drucker em 1955, o mesmo defendia que um bom modelo de negócio é aquele que permite descobrir quem é o cliente e o que o mesmo valoriza. O modelo deverá ainda permitir responder a algumas questões fundamentais sendo elas: Como se obtém dinheiro/lucro através do negócio criado ou gerido e indicar a lógica económica que permite fazer a ligação da entrega de um produto com valor e custo apropriado ao cliente [86].

Apesar de o conceito ser algo antigo, a investigação acerca deste tema só ganhou relevo mais recentemente, sendo que atualmente não existe uma definição absoluta pois não há uma concordância dos investigadores quando a definem. No entanto umas das melhores definições referentes a atualidade foi feita por Osterwalder e Pigneur em 2010, criadores do primeiro modelo Canvas de negócio. Os mesmos definiram um modelo de negócio como uma lógica que uma organização necessita de obter em ordem para criar, entregar e capturar algum tipo de valor. O modelo proposto é dividido em 9 partes interconectadas (Proposta de Valor, Parcerias Chave, Atividades Chave, Recursos chave, Estrutura de Custos, Relacionamento com Clientes, Segmento de Clientes, Canais e Fontes de Receitas) e é aquele que continua a ser utilizado atualmente [87].

Este modelo e os respetivos componentes referentes a este software a ser desenvolvido de QR Codes irão ser apresentados de seguida, de uma forma mais completa para finalmente ser elaborado um modelo Canvas mais resumido e conciso:

- **Proposta de Valor** – As suas páginas e códigos num só lugar. Crie páginas personalizadas para os seus produtos e serviços através de uma solução completa e robusta. Com apenas um código tem toda a informação na sua mão.
- **Segmento de Clientes** – Empresas comerciais (Ex: Lojas de roupa) e industriais (Ex: Corticeiras), que efetuem fabrico ou produção de produtos e que pretendam oferecer aos clientes informação sobre os mesmos (qualquer produto fabricado/comercializado poderá conter um código QR). Empresas de distribuição (Ex: CTT, UPS, DHL e entre outras), permitindo aos distribuidores obter mais informação sobre o produto a ser transportado.
- **Canais** – Publicidade em diversos meios de comunicação (jornais, televisão e entre outros), promoção através das redes sociais, contacto direto com clientes (clientes de softwares já existente e desenvolvidos pela empresa).
- **Relacionamento com Clientes** – Suporte via email ou telefónico em horário útil.
- **Fontes de Receitas** – Receitas obtidas através da subscrição do serviço (oferta de pacotes variados mensais, trimestrais e anuais). Customizações para além dos packs propostos têm um custo adicional.
- **Parcerias Chave** - Parcerias com empresas de cópias/impressão (fornecer preços mais baixo, caso os clientes pretendam fazer impressões dos códigos em alta qualidade e não tenham condições para o fazer na sua empresa).
- **Atividades Chave** – Desenvolvimento de software (Criação de Landing Pages personalizáveis, Criação de Landing Pages a partir do modelo criado, Gestão de Landing Pages, Criação e personalização de códigos QR), Investigação científica e tecnológica, Atendimento ao cliente, Marketing.

- **Recursos chave** – Equipa e técnicos de desenvolvimento de software (programadores, designers e etc), Equipa de investigação (para estudar o estado atual do mercado inserido e detetar possíveis mudanças e adições ao software), Hardware para o desenvolvimento (computadores, periféricos), Equipa de marketing, Software de apoio a desenvolvimento (Ex: Ambiente de desenvolvimento integrado)
- **Estrutura de Custos** – Este software vai ter custos relativamente aos recursos humanos pois é necessário pagar aos desenvolvedores, equipas de marketing e restante pessoal, custos de manutenção da plataforma, custos relativos as parcerias adquiridas e publicidade, licenças de software utilizadas para o desenvolvimento.

De seguida, é apresentada uma imagem (Figura 13) onde se encontram todos os componentes referidos anteriormente, de forma mais resumida. Obtemos então assim o nosso modelo de negócio Canvas.

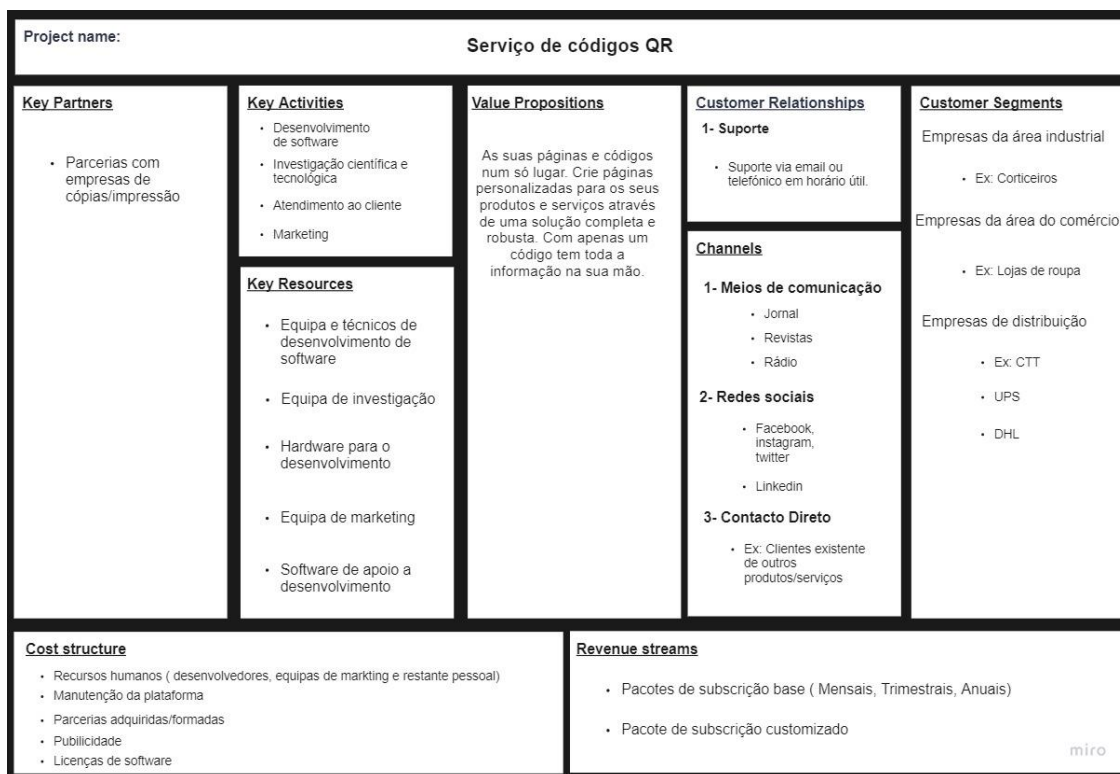


Figura 13 - Modelo de negócio Canvas

3.5 Analytic Hierarchy Process (AHP)

Normalmente quando necessitamos de realizar a tomada de decisão sobre algum assunto pouco pensamos sobre o assunto e realizamos a nossa escolha de cabeça quente. Para realizar e tomar decisão sobre algo é necessário proceder a realização de algum raciocínio prévio de forma a realizamos a melhor escolha.

Para existir decisão sobre algo é necessário estabelecer um conjunto de prioridades e organização das mesmas. Este tipo de organização normalmente pode ser adquirido através do uso de uma representação hierárquica. Um dos métodos que proporciona este tipo de necessidade é o método AHP, introduzido por Thomas Saaty. O mesmo envolve cálculos matemáticos tendo como base álgebra linear, sendo que difere de muitos métodos tradicionais pois requer que os valores numéricos utilizados estejam de acordo com as medições científicas. Este método é importante e de grande valor pois para o tipo de decisão a tomar é o mais adequado, sendo que segundo o autor o método é apropriado quando os critérios de decisão são algo subjetivos, abstratos e não quantificáveis [88].

3.5.1 Fase 1 - Construção da árvore hierárquica de decisão

Nesta fase é necessário fazer a definição do problema a resolver para de seguida se estruturar um diagrama hierárquico. É necessário decompor o problema/decisão numa hierarquia composta por no mínimo um objetivo, critérios e alternativas [89].

De seguida, na Figura 14, irá ser demonstrado o digrama hierárquico, bem como a explicação de todos os componentes que a compõem.

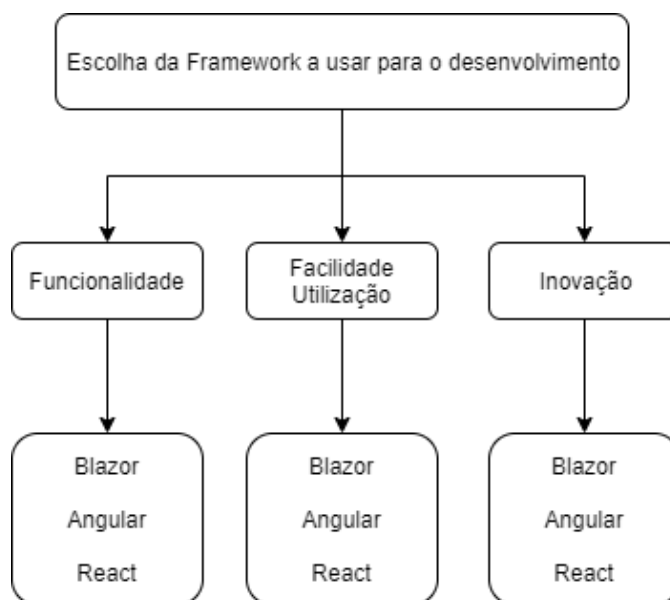


Figura 14 – Árvore hierárquica de decisão

Observando agora a Figura 14, conseguimos perceber que o objetivo deste diagrama desenvolvido é escolher uma Framework a usar para o desenvolvimento da plataforma. Foram escolhidos também 3 critérios para a comparação das mesmas sendo eles a funcionalidade, facilidade de utilização e a inovação, de forma a que seja possível quantificar e tomar a decisão correta. De seguida explicamos um pouco cada critério selecionado:

- **Funcionalidade** – Permitir que no desenvolvimento possa ser elaborado um software robusto e completo com todas funcionalidades possíveis sem limitações por parte da Framework ou necessidade de uso e integração de softwares ou Frameworks externas.
- **Facilidade de utilização** – É necessário entender a facilidade de utilização da Framework, neste caso aquela que permite ao programador desenvolver de forma mais facilitada. Alguns dos fatores que indicam a facilidade são por exemplo: o tempo de aprendizagem e a existência de boa documentação ou falta da mesma.
- **Inovação** – Nos dias de hoje existem diversas Frameworks para o desenvolvimento de software e muitas vezes são sempre reutilizadas as mesmas, um software completo que seja elaborado em algo novo/inovador acresce sempre mais valor do que um elaborado em soluções em que existe saturação no mercado.

3.5.2 Fase 2 - Comparação das alternativas e dos critérios escolhidos

Esta fase consiste na comparação das alternativas e dos critérios escolhidos. A cada critério selecionada deverá ser atribuído um nível de importância/prioridade, segundo que a mesma é efetuada em comparação com os restantes critérios resultando assim na chamada matriz de comparação [89].

Esta comparação é fundamental no uso deste método, Thomas Saaty definiu que a mesma deve comparar cada critério selecionada comparando as mesmas em grau de importância. O mesmo elaborou uma escala fundamental com atribuição de prioridade em forma numérica como pode ser verificado na tabela abaixo. Desta atribuição deve resultar então uma matriz de comparação onde são comparados os elementos a esquerda desta matriz com os respetivos elementos superiores, sendo que número de “julgamentos” de uma matriz de ordem n , calcula-se da seguinte maneira, $n(n-1) / 2$, sendo n o número de elementos a comparar. De referir que existem situações especiais em que pode ser menor. O autor afirma ainda que é importante fazer a pergunta certa de modo a que seja obtida a resposta certa [88].

Tabela 4 – A escala Fundamental de Thomas L Saaty [88][89]

Intensidade da importância	Definição	Explicação
1	Igual importância	Duas atividades contribuem de maneira igual para o objetivo
3	Moderada importância	A experiência e o julgamento favorecem levemente uma atividade em relação à outra
5	Forte importância	A experiência e o julgamento favorecem fortemente uma atividade em relação à outra
7	Muito Forte importância	Uma atividade é fortemente favorecida e a sua dominância é demonstrada na prática
9	Extrema importância	Uma atividade esta acima da outra com o maior grau de afirmação possível
2,4,6,8	Valores intermédios	Quando há necessidade de compromisso

Agora que definimos todos os parâmetros necessários e apresentamos os valores possíveis da escala Fundamental, iremos então proceder a realização da nossa matriz de comparação com os critérios Funcionalidade, Facilidade de Utilização e Inovação em combinação com os valores da escala de Saaty.

Tabela 5 – Matriz de comparação de critérios

Critérios	Funcionalidade	Facilidade de Utilização	Inovação
Funcionalidade	1	3	2
Facilidade de Utilização	1/3	1	1/2
Inovação	1/2	2	1

3.5.3 Fase 3 – Prioridade relativa de cada critério

Nesta fase vai ser atribuído uma prioridade relativa de cada critério. Numa primeira parte é necessário normalizar os valores da matriz de comparações, esta normalização tem objetivo

colocar todos os critérios na mesma unidade, sendo que para isto ser possível cada valor da matriz terá de ser dividido pelo total da sua respetiva coluna [89].

Tabela 6 – Matriz de comparação dos critérios do Segundo Nível

Critérios	Funcionalidade	Facilidade de Utilização	Inovação
Funcionalidade	1	3	2
Facilidade de Utilização	1/3	1	1/2
Inovação	1/2	2	1
Soma	11/6	6	7/2

Depois de efetuada a soma é que é possível efetuar a matriz normalizada como se pode observar na tabela abaixo.

Tabela 7 – Matriz normalizada dos critérios do Segundo Nível

Critérios	Funcionalidade	Facilidade de Utilização	Inovação
Funcionalidade	6/11	1/2	4/7
Facilidade de Utilização	2/11	1/6	1/7
Inovação	3/11	1/3	2/7

Por último é necessário obter o vetor de prioridades, este tem como objetivo principal identificar a importância de cada um dos critérios selecionados, isto é efetuado através do cálculo da média aritmética dos valores de cada linha da matriz normalizada, que obtemos na tabela anterior.

Tabela 8 – Peso dos valores/Prioridade relativa (Vetor de prioridades)

Critérios	Soma das linhas	Prioridade relativa
Funcionalidade	249/154	Soma das linhas/3 = 0.54
Facilidade de Utilização	227/462	Soma das linhas/3 = 0.16
Inovação	206/231	Soma das linhas/3 = 0.30

Conseguimos agora analisar qual o critério mais importante ao menos importante. Podemos descrever de importância temos os seguintes critérios: Funcionalidade, Inovação e por fim Facilidade de Utilização.

3.5.4 Fase 4 – Avaliar a consistência das prioridades relativas

Esta próxima fase consiste em analisar/avaliar a consistência das prioridades relativas, para isso é necessário calcular a Razão de Consistência (RC) para medir o quanto os julgamentos foram consistentes em relação a grandes amostras de juízos completamente aleatórios.

No entanto para efetuar o cálculo RC teremos de obter os valores correspondente a fórmula, ou seja, é necessário efetuar primeiro o cálculo do índice de consistência (IC) e o índice aleatório (IR), só assim conseguimos prosseguir. De seguida temos a demonstração dessas fórmulas matemáticas, começando pelo RC.

$$RC = \frac{IC}{IR} \quad (1)$$

De seguida temos a fórmula do índice de consistência:

$$IC = \frac{\gamma_{max} - n}{n - 1} \quad (2)$$

Na fórmula anterior do IC, sabemos que n é o número de critérios a ser utilizados. Neste caso o valor de n será 3. Temos agora de efetuar o cálculo do γ_{max} que é valor próprio, obtido através do uso do vetor próprio ou vetor de propriedades. Num primeiro passo fazemos o produto entre a matriz de comparação e o vetor próprio.

$$\begin{bmatrix} 1 & 3 & 2 \\ 1/3 & 1 & 1/2 \\ 1/2 & 2 & 1 \end{bmatrix} \times \begin{bmatrix} 0.54 \\ 0.16 \\ 0.30 \end{bmatrix} = \gamma_{max} \begin{bmatrix} 1.62 \\ 0.49 \\ 0.89 \end{bmatrix} \quad (3)$$

Finalmente para calcular o γ_{max} que é obtido através da média, da divisão dos valores resultantes na equação anterior (3) com o vetor próprio ou de prioridades. De seguida é apresentado essa mesmo cálculo.

$$\gamma_{max} = \text{média} \left\{ \frac{1.62}{0.54}, \frac{0.49}{0.16}, \frac{0.89}{0.30} \right\} = 3.01 \quad (4)$$

Agora que sabemos o valor de γ_{max} podemos finalmente calcular o índice de consistência de acordo com a fórmula previamente apresentada.

$$IC = \frac{3,01 - 3}{3 - 1} = 0.005 \quad (5)$$

Agora que obtemos estes valores vamos voltar a fórmula do rácio de consistência (RC), para fazer este cálculo falta-nos apenas saber o valor do índice aleatório (IR), este valor é referente a um número de comparações par a par efetuadas. Este índice aleatório é calculado para matrizes quadradas de ordem n pelo Laboratório Nacional de Oak Ridge nos Estado Unidos da América[89], existem no entanto investigações que apontam para outras tabelas com valores

diferentes, no entanto esta é aquela que é mais creditada. Abaixo apresentamos essa tabela de onde vamos selecionar um valor de acordo com a nossa ordem n , que é o número de critérios. A tabela vai até a ordem 15, mas por motivos de espaço e porque não necessitamos de mais valores apresentamos só os 10 primeiros valores para ter como referência.

Tabela 9 – Tabela de Índices aleatórios de ordem n .

N	1	2	3	4	5	6	7	8	9	10
IR	0.00	0.00	0.58	0.90	1.12	1.24	1.32	1.41	1.45	1.49

Visto que nossa ordem é igual a 3 ($n=3$) o valor correspondente do IR é 0.58. Finalmente efetuamos o cálculo da Razão de Consistência (RC).

$$RC = \frac{0,005}{0,58} \approx 0,01 \quad (6)$$

Observando o valor obtido de $RC \approx 0.01$, podemos observar então que $0.01 < 0.1$, neste caso podemos concluir que os valores das prioridades relativas atribuídas inicialmente estão consistentes.

3.5.5 Fase 5 – Construção da matriz de comparação paritária para cada critério

Nesta fase iremos proceder a construção da matriz de comparação paritária para critério tendo em conta cada umas das alternativas selecionadas anteriormente. Aqui irão ser efetuadas novas matriz de comparação e determinadas as prioridades relativas, só que desta vez iremos observar a importância relativa de cada uma das alternativas selecionadas anteriormente e que compõem a estrutura hierárquica desenvolvida na primeira fase deste processo [89].

Em primeiro lugar vamos efetuar as matrizes para o critério funcionalidade.

Tabela 10 – Matriz de comparação paritária do critério Funcionalidade

Funcionalidade	Blazor	Angular	React
Blazor	1	3	4
Angular	1/3	1	2
React	1/4	1/2	1
Soma	19/12	9/2	7

Tabela 11 – Matriz de comparação paritária do critério Funcionalidade normalizada

Funcionalidade	Blazor	Angular	React	Vetor prioridade
Blazor	12/19	2/3	4/7	0.62
Angular	4/19	2/9	2/7	0.24
React	3/19	1/9	1/7	0.14

Agora vamos efetuar as matrizes para o critério de Facilidade de Utilização.

Tabela 12 – Matriz de comparação paritária do critério Facilidade de Utilização

Funcionalidade	Blazor	Angular	React
Blazor	1	1/2	1/3
Angular	2	1	1/2
React	3	2	1
Soma	6	7/2	11/6

Tabela 13 – Matriz de comparação paritária do critério Facilidade de Utilização normalizada

Funcionalidade	Blazor	Angular	React	Vetor prioridade
Blazor	1/6	1/7	2/11	0.16
Angular	2/6	2/7	3/11	0.30
React	3/6	4/7	6/11	0.54

Por fim temos como último critério a Inovação a comparar com as Frameworks, onde vamos de novo elaborar as respectivas matrizes.

Tabela 14 - Matriz de comparação paritária do critério Inovação

Funcionalidade	Blazor	Angular	React
Blazor	1	4	5
Angular	1/4	1	2
React	1/5	1/2	1
Soma	29/20	11/2	8

Tabela 15 - Matriz de comparação paritária do critério Inovação

Funcionalidade	Blazor	Angular	React	Vetor prioridade
Blazor	20/29	8/11	5/8	0.68
Angular	5/29	2/11	2/8	0.20
React	4/29	1/11	1/8	0.12

Por fim, é elaborado novamente um diagrama em árvore hierárquica (Figura 15), desta vez com os valores obtidos nas diferentes fases para melhorar observar os valores obtidos e obter conclusões sobre os mesmos.

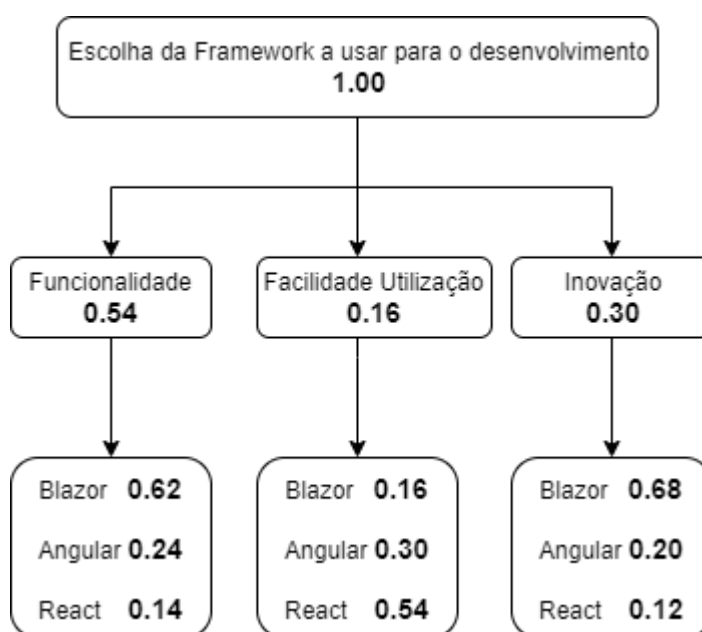


Figura 15 - Árvore hierárquica de decisão com os valores correspondentes

3.5.6 Fase 6 – Obter a prioridade composta para as alternativas

Nesta fase iremos utilizar os valores obtidos na fase 5 de comparação das Frameworks pelos critérios com os valores de comparação entre os próprios critérios. Aqui iremos multiplicar os valores obtidos na fase anterior com o vetor prioridade de comparação entre critérios como podemos verificar abaixo. Está é último cálculo a ser efetuado para ser tomada uma decisão de qual alternativa escolher.

$$\begin{bmatrix} 0.62 & 0.16 & 0.68 \\ 0.24 & 0.30 & 0.20 \\ 0.14 & 0.54 & 0.12 \end{bmatrix} \times \begin{bmatrix} 0.54 \\ 0.16 \\ 0.30 \end{bmatrix} = \begin{bmatrix} 0.56 \\ 0.24 \\ 0.20 \end{bmatrix} \quad (7)$$

3.5.7 Fase 7 – Escolha da alternativa

$$\begin{bmatrix} 0.56 \\ 0.24 \\ 0.20 \end{bmatrix} \quad (8)$$

De acordo com os valores obtidos na fase 6 podemos concluir que a Framework Blazor é aquela mais indicada para iniciar o desenvolvimento da solução, pois em função dos critérios definidos (funcionalidade, facilidade de utilização e inovação) e das suas respetivas importâncias, esta foi a que obteve o valor mais alto de 0.56 , sendo até um valor absoluto pois a soma das restantes alternativas ($0.24 + 0.22 = 0.44$) é menor que este valor.

3.6 Quality Function Deployment (QFD)

O Quality Function Deployment (QFD) é outro método de tomada de decisão que se descreve de forma geral como um conceito que tem como objetivo traduzir os objetivos/requisitos do cliente para os respetivos requisitos técnicos, produção e desenvolvimento de cada produto. Este conceito apareceu no Japão por volta dos anos 60 e 70 e espalhou-se rapidamente nos Estados Unidos da América nos anos 80, sendo que informação e variações deste método evoluíram e aparecem rapidamente [90].

Um dos passos mais importantes e utilizado obter informações do cliente para técnicas de engenharia tem um nome específico chamada House of Quality (HOQ), este processo requer informação por parte das equipas de marketing e técnicas, tudo isto proporcionada e motiva que haja mais contacto e comunicação entre todos o pessoal e cooperação das várias áreas funcionais [91].

Antes de efetuar esta tabela é necessário ter em conta a tabela realizada na secção 2.5.6 (Análise e comparação das plataformas), pois com ela temos toda a informação necessária para os requisitos técnicos a desenvolver. No entanto é necessário fazer um pré levantamento de requisitos dos clientes para que se possa preencher a coluna esquerda do QFD que responde à questão “O quê?”. De seguida, são apresentados aquilo que os clientes mais procuram no software a produzir, informação recolhida através de reuniões com os mesmos.

- Usabilidade
- Segurança da informação
- Customização/personalização
- Gestão
- Tolerância a falha

4 Análise e Desenho

Neste capítulo é apresentada uma análise e desenho da solução a conceber através da recolha de requisitos e identificação de possíveis alternativas de desenho.

O capítulo é composto por uma identificação dos requisitos funcionais e não funcionais, a apresentação e descrição do modelo do domínio, os casos de uso presentes (com descrição formal e detalhada), alternativas de possíveis arquiteturas. Por fim, é efetuado um desenho detalhado de alguns dos casos de uso com recurso e apoio a diversos diagramas.

4.1 Requisitos Funcionais

O primeiro passo nesta análise deverá ser evidenciar e definir os requisitos funcionais que irão ser necessários para a elaboração deste software/serviço. Todos os requisitos funcionais que irão ser evidenciados têm como fonte as etapas anteriores realizadas nesta dissertação. Para além disso foi também efetuada uma nova reunião na empresa com base na informação recolhida ao longo da dissertação.

Foram então definidos os seguintes requisitos funcionais deste software, devidamente numerados (R - Requisitos + Número):

- **R01-** Permitir gerar códigos QR (do tipo normal, frame, etc);
- **R02-** Possibilidade de inverter cores do código a gerar;
- **R03-** Possibilidade de inverter a posição do código;
- **R04-** Antes de gerar cada código deverá ser dada ao utilizador a opção de escolha do nível de correção de código QR;

- **R05-** O utilizador pode escolher gerar um código com ou sem *Quiet Zone* (parte da estrutura de um código);
- **R06-** A *landing page* a utilizar/criar deverá ser totalmente personalizável pelo utilizador. O mesmo escolhe os campos que quer colocar e a sua quantidade. (Ex: fotos, vídeos, links e outros)
- **R07-** Permitir ao utilizador incluir nas *landing pages* informação já existente (deverá ser possível para além dos campos proporcionados na criação da landing pages, incluir relatórios de qualidade, manuais técnicos, instruções de utilização, fichas de segurança, ou outros documentos sobre o produto observado);
- **R08-** Guardar o valor de cada campo das *landing pages* no próprio código QR (de forma a diminuir a carga de informação a guardar);
- **R09-** Possibilidade de definição de níveis de acesso distintos à informação guardada no código QR (informação pública e privada);
- **R10-** Criar campos/códigos dinâmicos (carregados de uma base de dados interna, da empresa que fornecesse o software, ou externa do cliente que utiliza o sistema);
- **R11-** O utilizador deverá conseguir guardar o código gerado em múltiplos formatos (.png, .jpeg, .bmp, etc);
- **R12-** O utilizador deverá ter um espaço de gestão, onde o mesmo poderá gerir as *landing pages* e os códigos QR criados para cada uma.
- **R13-** Permitir a visualização de dados analíticos (o utilizador irá conseguir ter alguns dados sobre a digitalização dos seus códigos, como por exemplo o número de vezes que o mesmo foi utilizado)
- **R14-** O utilizador deverá conseguir imprimir os códigos QR gerados.
- **R15-** Possibilidade de definir o idioma da informação (multi-idioma);
- **R16-** A informação/url gerado no código QR deverá ser encurtado (o url/informações geradas deverão encurtadas pois tanto o código QR e os urls têm limites de caracteres, portanto quando mais encurtado tiver a informação mais quantidade irá ser possível de gravar);
- **R17-** Possibilidade de rastreabilidade do produto final desde a expedição até ao consumo do mesmo;
- **R18-** Possibilidade de encomenda rápida de mais produtos iguais ao produto observado;

- **R19-** Possibilidade de registo de reclamações ou pedidos de informação sobre o produto observado;

4.2 Requisitos Não Funcionais

Igualmente importantes aos requisitos funcionais são os requisitos não funcionais. Estes tipos de requisito são importantes para se conseguir avaliar a qualidade do sistema, através de diversas propriedades.

Para identificar os requisitos não funcionais foi utilizado o modelo FURPS+, um modelo para classificação de atributos de qualidade de software (Tabela 16). Este modelo contém as categorias de funcionalidade, usabilidade, confiabilidade, desempenho e suportabilidade. O sinal “+” permite ainda identificar requisitos extra como: desenho, implementação, interface e físicos.

Perante esta dissertação são evidenciados na Tabela 16, os requisitos não funcionais consoante o modelo FURPS+ evidenciado.

Tabela 16 – Requisitos não funcionais de acordo com o modelo FURPS+

Classificação FURPS+	Requisitos não funcionais
RN01 - Funcionalidade (segurança)	Garantir que a informação do cliente e os campos privados introduzidos se encontram seguros (apenas acessível através de autenticação).
RN02 - Usabilidade	Garantir que o sistema satisfaz os seus utilizadores e é eficaz (cumpre os objetivos) e eficiente (cumpre os objetivos com mínimo esforço possível)
RN03 - Confiabilidade	Garantir que o sistema está pronto para falhas e consegue recuperar dos mesmos (tentar novamente obter um recurso que no momento se encontrava indisponível)
RN04 - Desempenho	Tempo de resposta/carregamento de uma <i>landing page</i> criada de no máximo 10 segundos
RN05 - Suportabilidade	Suporte para os browsers mais utilizados (Google Chrome, Microsoft Edge, Safari, Firefox, Internet Explorer)

4.3 Modelo de domínio

De forma, a que se possa ter um melhor entendimento dos requisitos levantados foi elaborado um modelo de domínio. Este tipo de diagrama tem como principal funcionalidade identificar o domínio do problema. O mesmo utiliza notação UML na forma de diagramas de classe de forma a ilustrar modelos de domínio. Este tipo de diagrama apresenta entidades e

as suas relações e não componentes de software como classes [92]. O diagrama pode ser observado na Figura 17.

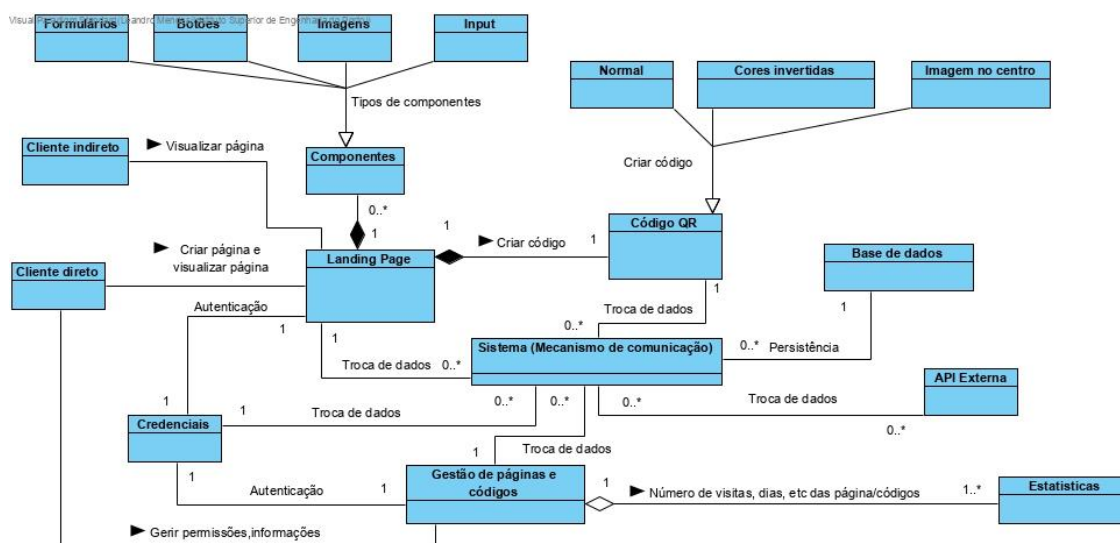


Figura 17 - Modelo de domínio da solução

No diagrama de modelo é possível perceber que o processo de autenticação acontece através da entidade de credenciais. O utilizador para conseguir utilizar a maior parte das funcionalidades disponíveis no sistema, terá que se autenticar fornecendo as suas credenciais. Para proceder ao processo de autenticação é necessário enviar os dados inseridos pelo utilizador para a base de dados para proceder a sua validação. O sistema deverá implementar um mecanismo que consiga tratar de todas as comunicações das suas entidades para o exterior, como as bases de dados internas ou externas a empresa, que são por si também duas entidades do diagrama.

No entanto o processo inicia quando um cliente direto desta solução pretende criar ou visualizar *Landing Pages*. Podemos observar no modelo (Figura 17) a entidade “*Landing Pages*”. Nesta entidade o cliente direto pode proceder à criação de *Landing Pages* personalizáveis. O utilizador deve conseguir adicionar e retirar componentes conforme as suas necessidades, sendo que existem diversos tipos de componentes. No diagrama apenas são identificados alguns como, por exemplo, formulários, botões e inputs. Após a seleção dos componentes a utilizar na página os dados seleccionados e inseridos têm de ser gravados na base de dados. É importante referir ainda que para se conseguir aceder a esta entidade é necessário se proceder previamente a autenticação do utilizador, através do processo referido anteriormente. Por fim, o utilizador deve ser redireccionado para a próxima página onde irá poder criar códigos QR.

Após criar uma página como referido é necessário criar um código QR, que é representado no modelo por uma entidade. Nesta entidade, como o nome indica, é onde são criados os códigos QR relativos a uma *Landing Page*, pois para se criar um código é necessário criar primeira uma *Landing Page*. Os códigos QR criados nesta secção podem ser de diversos tipos,

com as cores invertidas ou até mesmo com imagem no centro. É necessário atualizar na base de dados que foi criado um código referente à página. Aqui é também necessário que o utilizador se encontre autenticado para realizar qualquer ação.

Existe ainda uma entidade de gestão de páginas e códigos que obtém informações da base de dados. Um utilizador autenticado pode gerir as páginas e códigos criados ou que administra. Pode editar a página, criar novos códigos e criar novos códigos com a mesma estrutura da página, mas valores diferentes. Esta entidade dá origem a outra onde é possível observar estatísticas analíticas como: número de visitas as páginas, datas de acesso e número de digitalizações por código.

Ambos os clientes diretos (clientes da empresa) e indiretos (i.e., clientes dos clientes) poderão proceder à visualização das páginas criadas, através de comunicação com a entidade “*Landing Pages*”. Para que se possa aceder a uma página é necessário a leitura de um código QR, sendo que podem existir diversos códigos diferentes a apontar para a mesma página, mas cada código aponta apenas para uma página. Para se conseguir ver campos privados nesta página é necessário se encontrar autenticado e ter autorização para tal. Os dados carregados para a página são obtidos pelo código digitalizado e por uma comunicação com a base de dados.

4.4 Casos de uso

Para descrever melhor as funcionalidades da solução foi elaborado um diagrama de casos de uso (Figura 18). Um diagrama de casos de uso é uma das melhores ferramentas para visualizar os requisitos funcionais de um sistema. Este diagrama oferece uma análise de alto nível em que se dá mais atenção às interações com os utilizadores do que à funcionalidade [93].

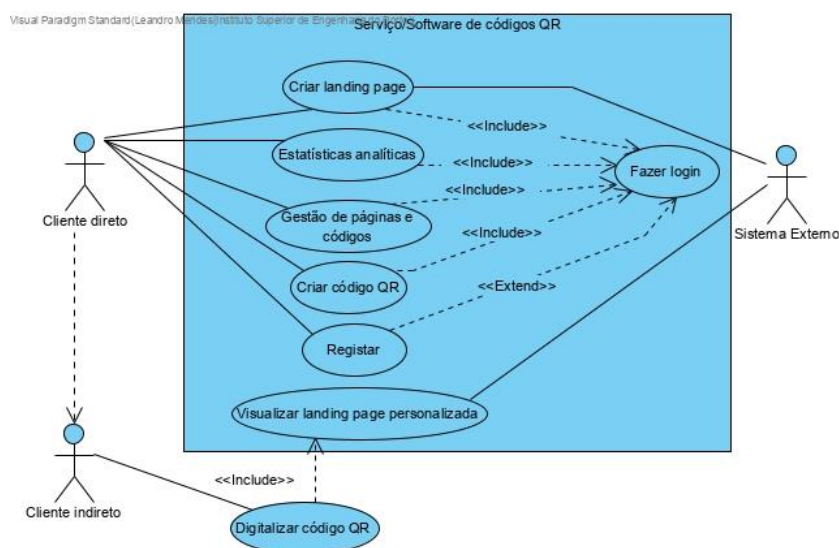


Figura 18 - Diagrama de casos de uso da solução

4.4.1 Descrição formal dos casos de uso

Nesta secção irá ser efetuado uma descrição dos casos de uso evidenciados na Figura 18. Esta descrição é efetuada em forma de tabela, onde a informação e fluxo de ações são apresentadas em detalhe.

Na Tabela 17, é possível observar a informação e fluxo de ações para o caso de uso “Fazer login/Autenticação”. São também indicados os requisitos que o caso de uso incorpora.

Tabela 17 – UC01: Fazer login/Autenticação

Objetivo	Permite ao utilizador autenticar-se para conseguir utilizar os serviços proporcionados.
Atores	Clientes diretos e indiretos (qualquer individuo que pretenda informar sobre os seus produtos ou serviços).
Pré-condições	Efetuar registo (possuir conta no sistema)
Pós-condições	Após o login, o utilizador é redirecionado para a página inicial do sistema.
Prioridade	Média
Frequência de uso	Alta
Campos	Email, palavra passe (opcional: número de telefone para autenticação a dois fatores)
Requisitos	<ul style="list-style-type: none">• R09, RN01, RN05
Fluxo principal	<ol style="list-style-type: none">1. O utilizador entra no site/página2. O utilizador seleciona a opção de login3. É apresentada a página de login4. O utilizador preenche os campos necessários (email e <i>password</i>)5. O utilizador clica no botão para submeter os dados6. O sistema verifica os dados submetidos e confirma existência do utilizador.

Na Tabela 18, é possível observar a informação e fluxo de ações para o caso de uso “Criar uma *Landing Page*”. Tal como indicado anteriormente são evidenciados os requisitos que o caso de uso incorpora.

Tabela 18 – UC02: Criar uma *landing page*

Objetivo	O utilizador deverá conseguir criar <i>landing pages</i> personalizadas, ou seja, colocar os campos /informação onde e pela ordem que desejar. Deverá ser possível adicionar informação proveniente de fontes externas à empresa que proporciona o serviço.
Atores	Clientes diretos (qualquer individuo que pretenda informar sobre os seus produtos ou serviços).
Pré-condições	É necessário que o utilizador se tenha autenticado.
Pós-condições	O utilizador é reencaminhado para a página de criação de códigos QR, onde poderá criar um código que redirecionara para a página criada.
Prioridade	Alta
Frequência de uso	Alta
Campos	Existe uma área de campos de configuração geral como a seleção do título da página, imagem de fundo, gestão de permissões e entre outros. Os restantes campos irão variar pois é possível introduzir campos dinamicamente. De referir que existem propriedades comuns em todas as seções como a possibilidade de colocar ou não etiqueta e colocar o campo público ou privado.
Requisitos	<ul style="list-style-type: none"> • R06, R07, R08, R09, R10, R15, RN01, RN02, RN05
Fluxo principal	<ol style="list-style-type: none"> 1. O utilizador entra no site/página 2. O utilizador efetua o login/autenticação 3. Seleciona a opção criar <i>landing page</i> 4. O utilizador pode ou não preencher alguns campos de configuração geral (fundo, logotipo, título página) 5. O utilizador adiciona campos conforme as suas necessidades e indica se são, são públicos ou privados. 6. Se optar, pode adicionar utilizadores para ver os campos privados ou administrar a página 7. O utilizador submete ou pré visualiza a página criada (pelo menos um campo deverá ser adicionado).

Na Tabela 19, é possível observar a informação e fluxo de ações para o caso de uso “Criar código QR”. Tal como indicado anteriormente são evidenciados os requisitos que o caso de uso incorpora.

Tabela 19 – UC03: Criar código QR

Objetivo	O utilizador deverá conseguir criar códigos QR totalmente personalizados, que irão servir como acesso rápido e fácil às <i>landing pages</i>
Atores	Clientes diretos (qualquer individuo que pretenda informar sobre os seus produtos ou serviços).
Pré-condições	É necessário que o utilizador se tenha autenticado e tenha criado uma <i>landing page</i> para que se possa associar o código com a página.
Pós-condições	Após criar o código o utilizador pode fazer download do mesmo em vários formatos, criar outro código QR diferente e voltar à página inicial ou à página criação de uma nova <i>landing page</i> .
Prioridade	Alta
Frequência de uso	Alta
Campos	O utilizador tem de obrigatoriamente selecionar um nível de correção do código (L, M, Q, H). Opcionalmente pode selecionar cores diferentes e colocar uma imagem no centro do código.
Requisitos	<ul style="list-style-type: none"> • R01, R02, R03, R04, R05, R08, R11, R14, R16, RN02, RN05
Fluxo principal	<ol style="list-style-type: none"> 1. O utilizador entra no site/página 2. O utilizador efetua o login/autenticação 3. O utilizador cria ou entra numa <i>landing page</i>. 4. O utilizador é redirecionado para a página de criação de códigos QR 5. O utilizador seleciona o nível de correção a utilizar (pode preencher os campos opcionais se o desejar) 6. O utilizador carrega no botão para gerar o código QR.

Na Tabela 20, é possível observar a informação e fluxo de ações para o caso de uso “Gestão de páginas e códigos”. Tal como indicado anteriormente são evidenciados os requisitos que o caso de uso incorpora.

Tabela 20 – UC04: Gestão de páginas e códigos

Objetivo	O utilizador deverá conseguir gerir as páginas e códigos criados. O utilizador poderá editar e eliminar páginas e códigos, bem como gerir permissões.
Atores	Clientes diretos (qualquer individuo que pretenda informar sobre os seus produtos ou serviços).
Pré-condições	É necessário que o utilizador se tenha autenticado e tenha criado pelo menos uma <i>landing page</i> .
Pós-condições	O utilizador após efetuar alguma operação de gestão numa página pode efetuar o mesmo para outras existente, pode voltar à página principal ou à página inicial.
Prioridade	Média
Frequência de uso	Média
Campos	O utilizador não necessita de preencher qualquer campo, pode apenas visualizar as suas páginas e códigos criados.
Requisitos	<ul style="list-style-type: none"> • R12, RN01, RN02, RN05
Fluxo principal	<ol style="list-style-type: none"> 1. O utilizador entra no site/página 2. O utilizador efetua o login/autenticação 3. O utilizador entra na página de gestão de páginas e códigos 4. O utilizador poderá visualizar, editar e eliminar páginas e códigos criados (se existirem).

Na Tabela 21, é possível observar a informação e fluxo de ações para o caso de uso “Estatísticas analíticas”. Tal como indicado anteriormente são evidenciados os requisitos que o caso de uso incorpora.

Tabela 21 – UC05: Estatísticas analíticas

Objetivo	O utilizador deverá conseguir visualizar informações estatísticas como visitas e cliques sobre as suas páginas e códigos
Atores	Clientes diretos (qualquer individuo que pretenda informar sobre os seus produtos ou serviços).
Pré-condições	É necessário que o utilizador se tenha autenticado e tenha criado pelo menos uma <i>landing page</i> .
Pós-condições	O utilizador após visualizar a informação poderá voltar à página principal, criação de <i>landing pages</i> e página de gestão dos códigos.
Prioridade	Baixa
Frequência de uso	Baixa
Campos	O utilizador não necessita de preencher qualquer campo.
Requisitos	<ul style="list-style-type: none"> • R13, RN01, RN02, RN05
Fluxo principal	<ol style="list-style-type: none"> 1. O utilizador entra no site/página 2. O utilizador efetua o login/autenticação 3. O utilizador entra na página de estatísticas analíticas 4. O utilizador poderá visualizar diversas informações sobre as páginas e códigos

Na Tabela 22, é possível observar a informação e fluxo de ações para o caso de uso “Visualizar *Landing Page*”. Tal como indicado anteriormente são evidenciados os requisitos que o caso de uso incorpora.

Tabela 22 – UC06: Visualizar *landing page*

Objetivo	Qualquer pessoa com acesso físico ao código QR deverá conseguir digitalizar o mesmo para visualizar uma <i>landing page</i> .
Atores	Clientes diretos (qualquer individuo que pretenda informar sobre os seus produtos ou serviços) e indiretos (clientes dos clientes diretos)
Pré-condições	É necessário a digitalização de um código QR (opcionalmente poderá ser necessário autenticação apenas para campos ocultos/privados)
Pós-condições	O utilizador após visualizar a informação poderá voltar à página principal, criação de <i>landing pages</i> e página de gestão dos códigos.
Prioridade	Baixa
Frequência de uso	Baixa
Campos	O utilizador não necessita de preencher qualquer campo para visualizar campos públicos. Se existir campos privados e se tiver permissão para os visualizar o utilizador poderá optar por se autenticar com o seu email e password.
Requisitos	<ul style="list-style-type: none"> • R10, R15, R17, R18, R19, RN01, RN02, RN03, RN04, RN05
Fluxo principal	<ol style="list-style-type: none"> 1. O utilizador digitaliza um código QR 2. O utilizador é redirecionado para a <i>landing page</i> correspondente. 3. Opcionalmente poderá se autenticar caso tenha permissão para visualizar campos privados presentes na página

4.4.2 Descrição detalhada do caso de uso principal: Criar uma *landing page*

O caso de uso de “criar uma *landing page*” é o mais importante de toda a solução, é importante entrar em detalhe de como o mesmo funciona e qual o seu aspeto visual. Este caso de uso ocorre após um utilizador se autenticar na plataforma. Após essa autenticação o utilizador pode visitar a página de criação de *landing pages*. Através do uso de *Mockups* irá ser explicado o objetivo visual e funcional deste caso de uso.

Esta página deverá conter uma *header* de saudação e informação ao utilizador, através do uso de uma imagem de fundo sobreposta por uma mensagem de boas vindas e informação da página atual na qual o utilizador se encontra.

Numa primeira fase deverá ser apresentado ao utilizador uma secção de configurações gerais. Nesta secção deverão ser apresentadas diversas possibilidades de configuração como um título para a página, um *background* para o *header* inicial (através de um URL indicado pelo utilizador) e um logotipo da empresa/serviço que o utilizador pretende promover (que será

devidamente colocado no canto superior esquerda da barra de navegação da página). De referir que, com o amadurecer da solução, mais configurações poderão ser adicionadas nesta secção.

Numa outra secção é possível adicionar *emails* de utilizadores autorizados a visualizar campos privados ou administrar a página.

Por fim, temos uma secção composta por uma lista de componentes existentes para as páginas e um botão de adicionar. Após seleccionar o componente na lista é possível carregar no botão de adicionar e, neste caso, as opções de personalização do componente serão apresentadas. O utilizador irá preencher esta secção consoante as suas necessidades e preferências. Após ter sido seleccionado no mínimo um componente o utilizador poderá fazer a submissão da configuração da página através de um botão.

De seguida, na Figura 19, é apresentado em forma de *Mockup* uma representação da idealização abordada no seu primeiro estado, ou seja, quando a página é carregada pela primeira vez.

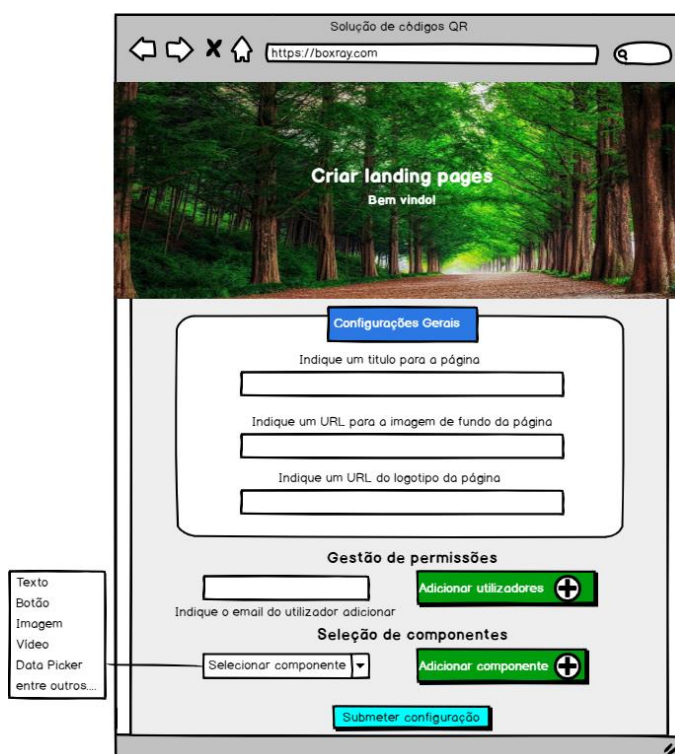


Figura 19 – *Mockup* do estado inicial da página de criação de *landing pages*

Após ser adicionado um componente, uma secção individual, em forma de balão ou cartão, será apresentada ao utilizador. O utilizador poderá fazer uma série de escolhas comuns a todos os componentes, tal como a adição de uma etiqueta (se o utilizador optar por adicionar uma irá ter de preencher um campo com o nome pretendido e escolher se deseja tornar o campo privado). Deverá existir também um botão para remover o componente adicionado.

Na Figura 20, é possível visualizar um *Mockup* resumido onde apenas é apresentado o componente adicionado à página.

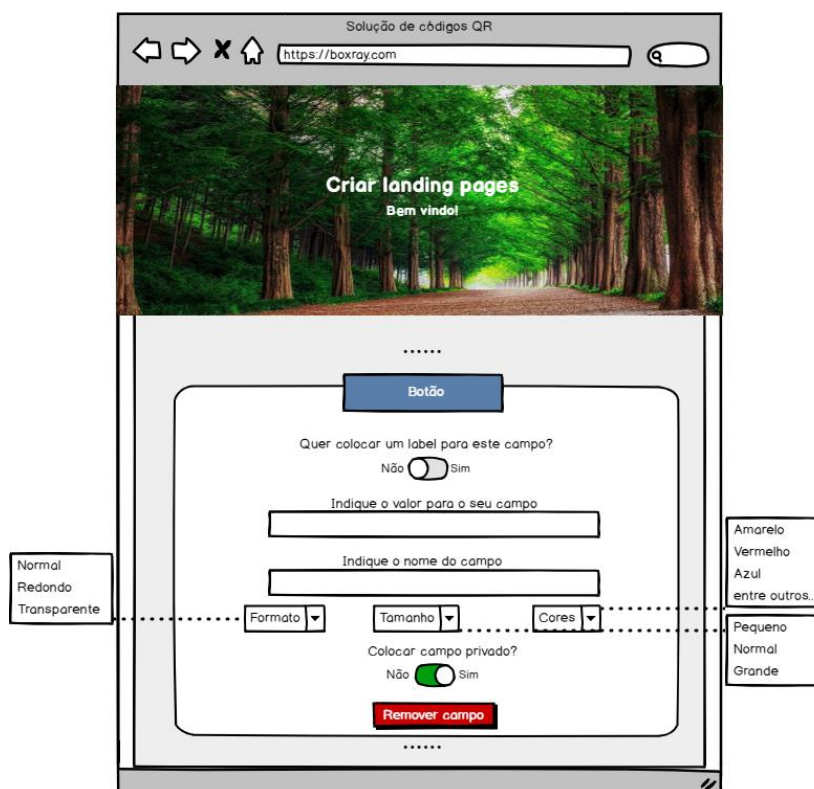


Figura 20 – *Mockup* resumido exemplo de um componente adicionado

Além dos campos anteriores, cada componente adicionado irá ter campos únicos como é possível observar pela Figura 20. Os campos podem ser, entre outros: ações a realizar, valor do componente selecionado, tamanhos e cores. É importante referir que as opções apresentadas são em linguagem de fácil compreensão a um utilizador comum, sem conhecimento de sistemas informáticos. Desta forma, o sistema torna-se abrangente e de fácil utilização.

4.5 Alternativas de Arquitetura

Após descritos os requisitos e casos de uso fundamentais da solução é importante pensar nas possíveis arquiteturas que o sistema pode adotar perante aquilo que foi identificado. São então descritas algumas alternativas possíveis através de diversos diagramas de componentes. Existem componentes que são comuns em todas, enquanto outros apenas irão aparecer se optarmos por determinada alternativa. De referir, que a ferramenta selecionada foi o Blazor Server. O Blazor mantém a lógica do cliente no servidor e através de um sinal troca mensagens com a DOM [68]. Ou seja, atualizações a interface, manipulação de eventos e chamadas JavaScript são tratadas através desse sinal (*SignalR*) [94].

4.5.1 Alternativa 1

São apresentados de seguida, os componentes principais desta arquitetura. Esta alternativa contém alguns componentes únicos e outros comuns às restantes (Figura 21).

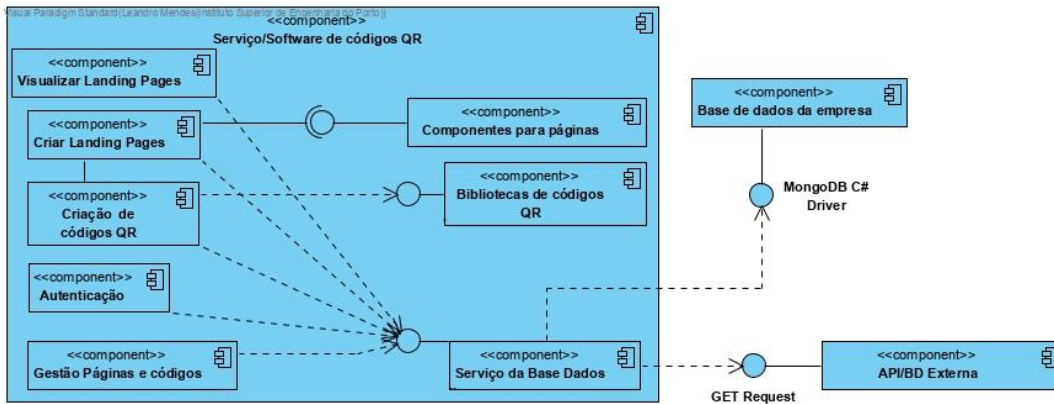


Figura 21 - Alternativa de arquitetura 1

- **Criar Landing Pages** – Neste componente temos toda a lógica presente para criar uma *landing page*. Uma *landing page* pode incluir diversos campos/componentes que a constroem, estes podem ser obtidos através do componente “Componentes para páginas”. Este tem conhecimento dos componentes que existem através da base de dados.
- **Visualizar Landing Pages** – Neste componente temos toda a lógica presente para visualizar uma *landing page* personalizada, para aceder ao mesmo é necessário fazer digitalização de um elemento externa ao sistema, um código QR criado.
- **Componentes para páginas** – Estes componentes representam botões, *links*, *inputs* e são complementares às *Landing pages*. Estes podem ser adicionados, removidos ou totalmente substituídos por outros posteriormente.
- **Criação de códigos QR** – Este componente irá incorporar a lógica para criar códigos QR, para o fazer irá necessitar de recorrer a bibliotecas ou serviços dependendo da alternativa selecionada
- **Autenticação** – Este componente como o nome indica irá proceder à autenticação do utilizador e, para o fazer, procede a uma consulta na base de dados. Esta consulta pode acontecer de diversas maneiras consoante alternativa adotada.
- **Gestão Páginas e códigos** – Este componente irá permitir fazer a gestão de todas as *landing pages* e códigos QR de cada utilizador. Para isso necessita de obter informação acedendo à base de dados.

Nesta alternativa temos uma base de dados externa que incorpora a sua interface. De forma a comunicar com a mesma utiliza-se o componente Serviço de Base de Dados.

- **Serviço de Base de Dados** – Este serviço contém toda a lógica necessária para comunicar com a base de dados e é interno ao sistema. Muitos dos restantes componentes invocam métodos presentes neste serviço que, por sua vez, envia pedidos para obter ou guardar informação na base de dados.
- **API/BD Externa** – Este componente é externo ao sistema e contém informação externas às bases de dados da empresa. Para se obter informação é efetuado um pedido http GET definido no componente “**Serviço de Base de Dados**”.

Nesta alternativa, os componentes Criar Landing Pages, Visualizar Landing Pages, Criação de códigos QR, Autenticação, Gestão Páginas e Códigos necessitam do componente Serviço de Base dados para efetuar as suas comunicações com a BD.

- **Bibliotecas de códigos QR** – Este componente irá representar todas bibliotecas necessárias para fazer um código QR, portanto o componente “Criação de códigos QR” nesta alternativa vai necessitar de utilizar o mesmo.

4.5.2 Alternativa 2

Nesta alternativa o serviço utilizado para fazer a comunicação com a base de dados encontra-se externo (com lógica independente) ao sistema principal. Esta comunicação é então efetuada através de uma API. Alguns componentes já foram apresentados anteriormente e não sofrem alterações.

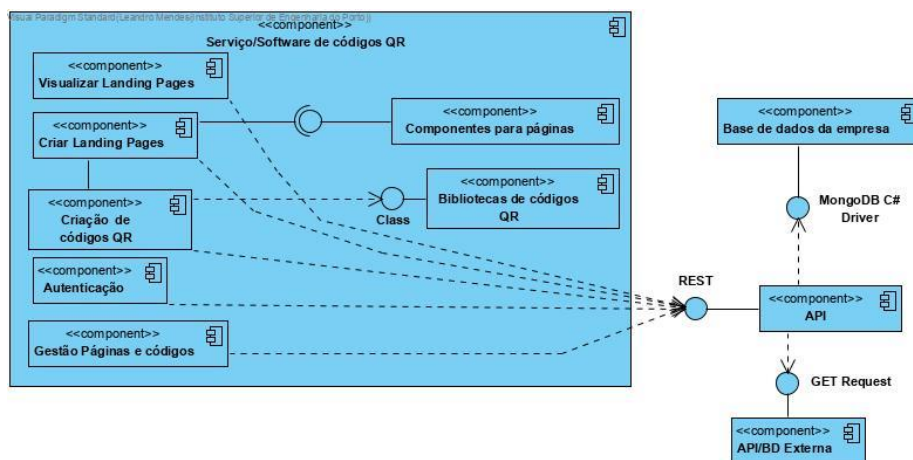


Figura 22 - Alternativa de arquitetura 2

- **API** – Esta API é externa ao sistema (lógica independente perante o sistema principal) e vai tratar de efetuar a comunicação à base de dados. Desta forma, esta API pode efetuar a comunicação à base de dados através de outra linguagem de programação.

Quanto aos componentes que necessitam de comunicação com a base de dados poderão comunicar com esta API, por exemplo, através de REST.

4.5.3 Alternativa 3

Nesta alternativa o serviço de comunicação de base dados é interno ao sistema, mas as bibliotecas necessárias para o componente “Criação de códigos” são externas e obtidos através de uma API (com lógica independente).

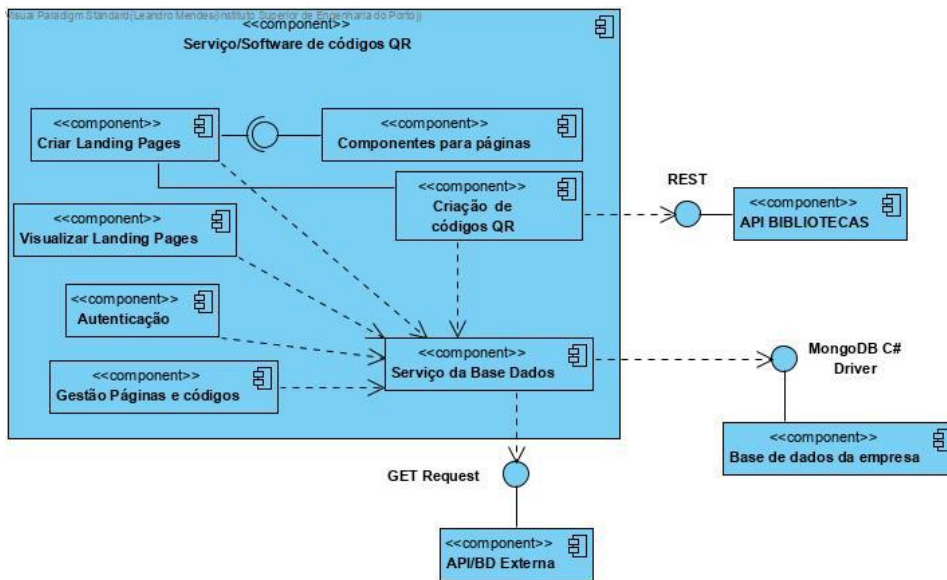


Figura 23 - Alternativa de arquitetura 3

- **API Bibliotecas** – Este componente irá ser uma API que irá conter todas as bibliotecas necessárias para a criação de códigos QR, estas podem ser instaladas diretamente na API ou efetuados pedidos externos as mesmas. A API irá tratar de gerir todas estas bibliotecas e fornecer a sua lógica para o componente “Criação de códigos QR”.

4.5.4 Alternativa 4

Nesta alternativa o serviço de comunicação a base de dados e as bibliotecas deixam de existir e passam a ser externos ao sistema (com lógica independente). Foi criado então o componente API Serviços.

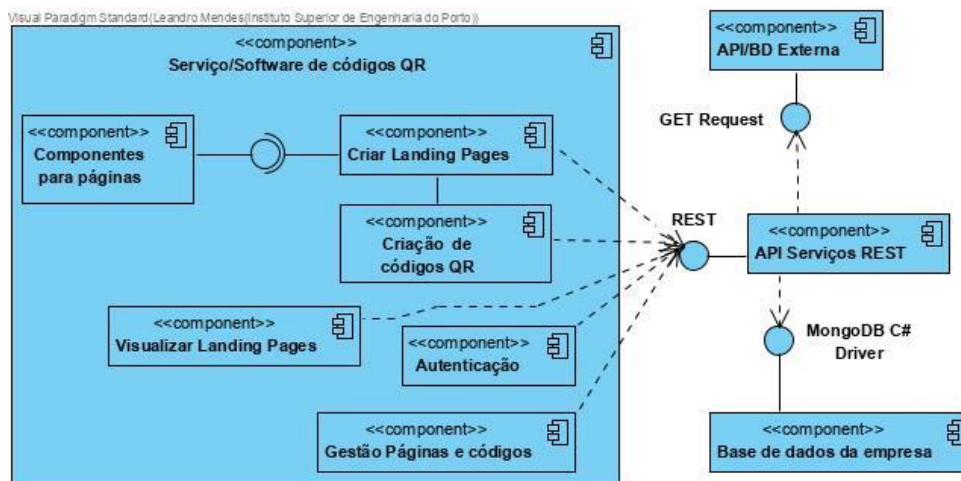


Figura 24 - Alternativa de arquitetura 4

- **API Serviços REST** – Este componente incorpora toda a lógica necessária para efetuar comunicação com a base de dados e comunicação com as bibliotecas de códigos QR. Estas bibliotecas podem ser externas, ou seja, pedidas a algum serviço ou então instaladas na própria API.

4.5.5 Escolha da alternativa

A necessidade de existir um serviço/API externo para tratar das comunicações com a base de dados é uma boa prática a adotar no desenvolvimento de qualquer software e portanto deve ser aplicado nesta solução. Quanto às bibliotecas de códigos QR, algumas delas são bastantes completas e apenas serão necessárias algumas linhas de código para as aplicar, portanto não existe necessidade de complicar algo simples criando mais uma API REST, sendo que se deverá optar por implementar diretamente na solução. De acordo com as necessidades da solução a conceber anteriormente evidenciadas, a alternativa que melhor representa os objetivos a atingir será a alternativa número 2, que se encontra representada pela Figura 22.

4.6 Desenho detalhado

Nesta secção será efetuado um desenho detalhado para os casos de uso mais importantes desta solução. Para que se possa perceber o que foi feito foram elaborados diagramas acompanhados por explicações detalhadas.

4.6.1 Criar Landing Page

4.6.1.1 Abordagem da solução para criar Landing Pages

Perante a investigação efetuada sobre a temática das landing pages, foi possível observar que existem diferentes abordagens para a implementação destas como a utilização de *Markdown* e *Drag & Drop*. Na preconização desta solução foi definido após o estudo anterior que as opções de configuração dos diversos componentes a personalizar deveriam ser expostos numa linguagem natural e do dia-a-dia. No entanto, seria necessário fazer a conversão dessa linguagem para código HTML. A abordagem encontrada foi a utilização de um documento de configurações na base de dados, onde a cada palavra selecionada pelo utilizador irá corresponder um HTML/CSS definido na base de dados. Por exemplo, ao configurar um botão é dado ao utilizador a escolha de mudar o tamanho do mesmo, são apresentados ao utilizador opções como: grande, médio e pequeno. Na base de dados essas palavras irão corresponder a código/classes HTML como: “btn-sm” ou “btn-bg”.

Esta foi abordagem escolhida pois como indicado anteriormente, um dos grandes objetivos é fazer com que esta solução possa ser utilizada por pessoas com poucos conhecimentos técnicos. Desta forma, é possível manter a complexidade da solução e o potencial daquilo que pode fazer, oferecendo uma experiência mais abrangente, acessível e diferente das existentes no mercado.

4.6.1.2 Diagrama SSD

Inicialmente, foi realizado um diagrama de sequência do sistema (Figura 25). Este tipo de diagrama é de alto nível e coloca o sistema como uma espécie de caixa negra. Desta forma é possível facilitar e entender melhor as interações entre o ator e o mesmo. Aqui é apenas apresentado de forma simples as interações entre um utilizador e o sistema para o caso de uso “Criar Landing Page”.

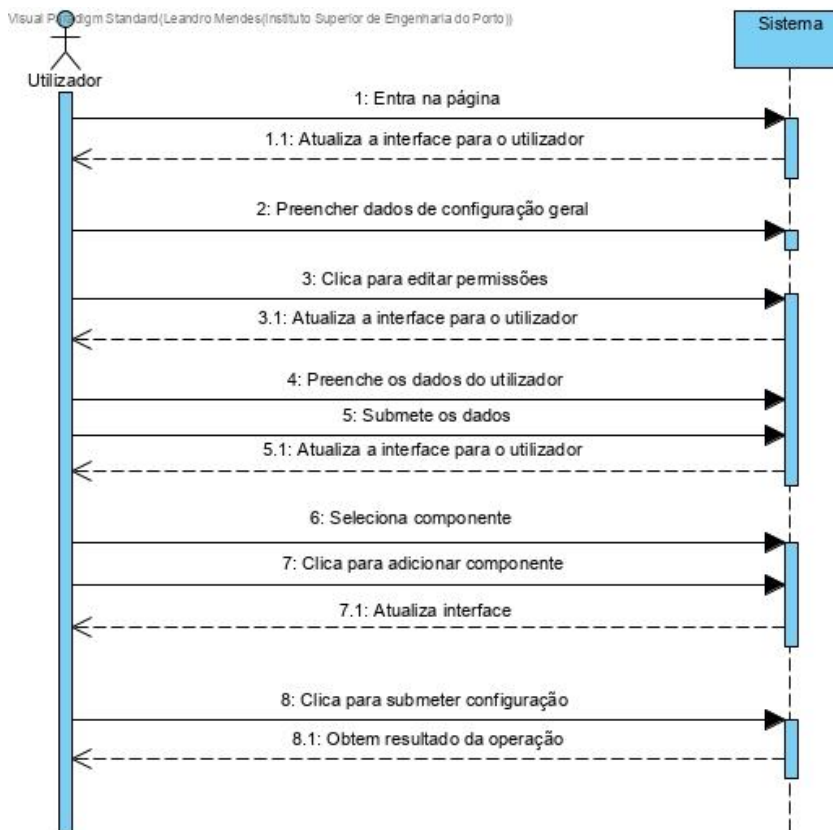


Figura 25 – Diagrama de sequência do sistema (Criar uma Landing Page)

Após ter sido efetuado uma visão mais superficial e de alto nível, é possível realizar um diagrama mais complexo em que se entra mais em detalhe, o chamado diagrama de sequência. Este diagrama é representado com o nome dos métodos e classes utilizados na implementação, presente no capítulo 5.

4.6.2 Visualizar Landing Page

4.6.2.1 Diagrama SSD

Tal como anteriormente, foi realizado um diagrama de sequência do sistema (Figura 26). Este tipo de diagrama é de alto nível e coloca o sistema como uma espécie de caixa negra. Desta forma é possível facilitar e entender melhor as interações entre o ator e o mesmo.

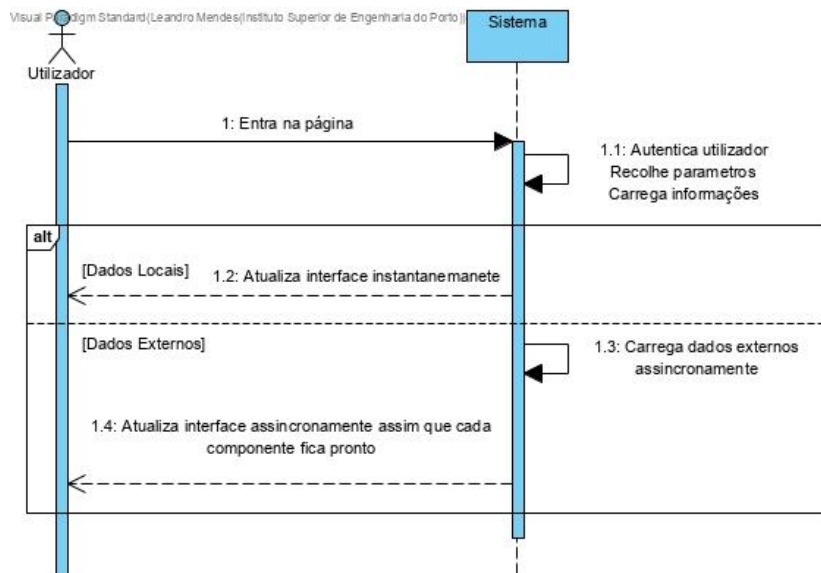


Figura 26 - Diagrama de sequência do sistema (Visualizar Landing Page)

Como é evidenciado, pela imagem Figura 26 a página efetua sempre um carregamento inicial para os dados locais, e a medida que vai obtendo o resultado dos pedidos externos atualiza a sua interface.

Uma visão mais detalhada através de um diagrama de sistema foi também efetuada. É possível identificar os passos necessários e possíveis para visualizar uma *landing page*. Este diagrama é representado com o nome dos métodos e classes utilizados na implementação, presente no capítulo 5.

4.6.3 Criar código QR

4.6.3.1 Diagrama SSD

Tal como anteriormente, foi também realizado um diagrama de sequência do sistema, como é possível observar na imagem abaixo (Figura 27). Este tipo de diagrama é de alto nível e coloca o sistema como uma espécie de caixa negra. Desta forma é possível facilitar e entender melhor as interações entre o ator e o mesmo.

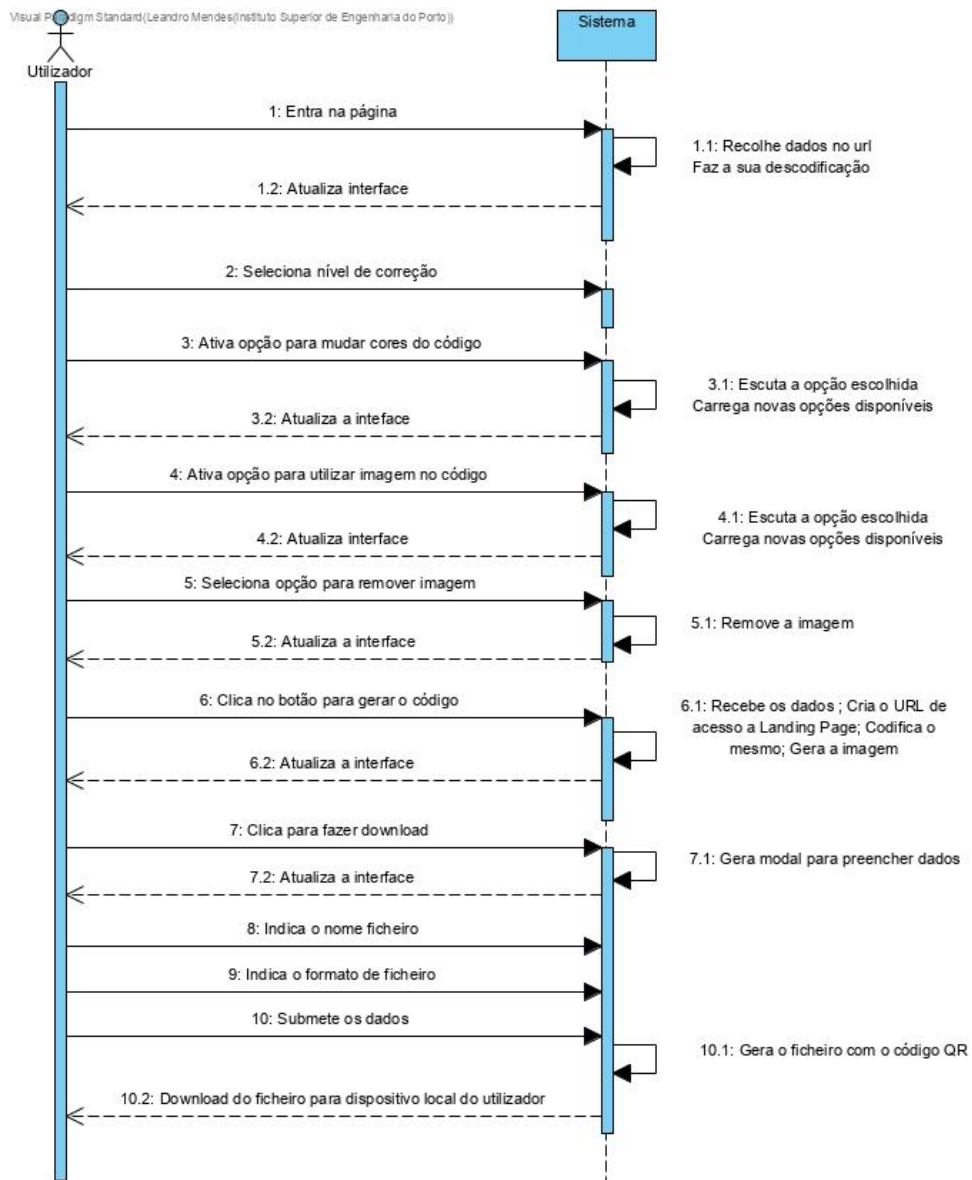


Figura 27 - Diagrama de sequência do sistema (Criar código QR)

Na Figura 27, é possível observar as diversas interações que o utilizador pode realizar quando está a criar os seus códigos QR. As diversas opções de personalização presentes (Ex: escolher cor, nível de correção) possibilitam uma maior variedade e satisfação geral do utilizador com os códigos criados. No capítulo 5, tal como nos casos de uso anteriores, é apresentada um diagrama de sequência com base na implementação com o nome dos métodos e classes utilizados.

5 Implementação

Após uma análise e desenho detalhado da solução a conceber é necessário apresentar alguns dos pontos importantes. Serão aqui apresentadas as metodologias utilizadas para o desenvolvimento, alguns dos pontos principais de implementação sendo que será apresentada alguma da lógica e código utilizado para atingir os objetivos pretendidos.

5.1 Metodologia adotada para gestão do desenvolvimento

Para o desenvolvimento de qualquer tipo de software é necessário cumprir com uma metodologia previamente definida. Deste modo é possível atingir um elevado nível de organização e perceber no ciclo de desenvolvimento aquilo que está feito e o que está por fazer, tal como outras variáveis importantes no desenvolvimento.

A empresa que requereu este software, a Ifthen, adota um conjunto de regras e boas práticas para a sua equipa de colaboradores. Apesar deste software ser um projeto realizado a nível individual as regras e boas práticas de desenvolvimento utilizadas pela empresa foram aplicadas também neste projeto.

De forma a gerir as versões do código fonte desenvolvido a empresa utiliza o Git. O Git é uma ferramenta gratuita e de código aberto que serve para controlar as diversas versões do código utilizada [95]. A empresa utiliza em conjunto com o Git a ferramenta bitbucket. O bitbucket é um serviço que permite fazer hospedagem ao código controlado por versões e faz a integração com o Git. Esta permite a criação de equipa ou de organizações para os seus diversos repositórios e para que se possa trabalhar[96]. A solução concebida faz uso da ferramenta para fazer o gerenciamento da solução e para que seja mais fácil utilizar os comandos Git foi utilizada uma GUI denominada sourcetree. O sourcetree simplifica as interações com o Git e melhorar a visualização e gestão de repositórios, a empresa oferece liberdade de escolha nesta interface [97].

É importante referir que utilizar serviços de hospedagem como o Bitbucket nesta solução é importante, pois o mesmo permite e tem entregado opções para construir, testar e implantar. Isto permite realizar implantação e entrega contínua, que apesar de não estar integrado atualmente na solução, permite que a mesma esteja preparada para o poder fazer no futuro.

5.2 Modelo de dados

Para se proceder à implementação da abordagem anterior é necessária uma base de dados. Foi decidido que a lógica de operações com a base de dados se iria encontrar numa API REST, que irá processar os pedidos efetuados. A lógica de implementação desta solução foca-se em dados que podem ter campos dinâmicos e diferentes, ou seja, a estrutura podia variar. Foi então necessário inicialmente escolher entre uma base de dados relacional ou não relacional. Uma base de dados não relacional não necessita de ter o seu esquema totalmente definido inicialmente e os dados não necessitam de ter uma estrutura, o que permite uma grande escalabilidade e flexibilidade, um ponto muito importante para uma solução como esta em que a estrutura e dados podem aumentar [98]. Sendo assim, foi utilizada uma base de dados não relacional: primeiro porque cada componente poderia vir a ter no futuro mais opções de personalização; segundo podem ser adicionados novos componentes. Abaixo, é demonstrado a estrutura do documento base de configuração dos componentes.

```
> _id: ObjectId("5df773267ee5823f584d0f1d")
  > Tipos: Array
  > Botao: Array
    > 0: Object
      Normal: "btn"
      Redondo: "btn btn-round"
      Transparente: "btn btn-link"
    > 1: Object
    > 2: Object
  > Link: Array
  > Imagem: Array
  > Email: Array
  > Pdf: Array
  > Vídeo: Array
  > Data: Array
  > Horas: Array
  > Data e Horas: Array
  > Input: Array
  > Espaço temporal: Array
  > Callback Local: Array
  > Secao: Array
  > Áudio: Array
  > Texto: Array
```

Figura 28 – Documento de configurações e componentes disponíveis

A base de dados utilizada nesta solução foi o MongoDB. Visto que a API será efetuada em c#, irá ser necessário uma biblioteca para conseguir efetuar as comunicações com a base de dados. A biblioteca necessária para conseguir efetuar comunicações é o MongoDB C# Driver. Inicialmente é necessário instalar e fazer referência à mesma na API. Na API acede-se ao ficheiro do arranque e insere-se uma linha que aponta para um ficheiro de configuração.

Nesse ficheiro são indicadas algumas propriedades necessárias para a ligação como o nome da coleção (semelhante as tabelas numa base de dados relacional) a aceder, o nome da base de dados e o mais importante a string de conexão.

5.3 Estrutura da solução

Para que se possa ter um melhor entendimento daquilo que foi efetuado são apresentados os diretórios existentes na solução principal e o que cada um contém. Desta forma é possível entender melhor aquilo que foi criado. Primeiro são evidenciados todos os diretórios principais existentes:

- **Areas** – Este diretório contém todos os ficheiros (.cshtml/cs) necessários para aplicar o ASP.NET Core Identity um sistema de autorização e autenticação (evidenciado na seção 5.4);
- **Data** – Contém todas as classes (.cs) necessárias e de auxílio às páginas criadas, contém na sua maioria definição de objetos.
- **p_Componentes** – Este diretório contém todos os subcomponentes utilizados e inicializados pela página de leitura das *landing pages*.
- **Pages** - Contém todas as páginas possíveis de visitar da solução (.cshtml).
- **Resources** - Contém todos os ficheiros de recursos de língua (.resx), para poder ser efetuada a tradução. Existe necessidade destes ficheiros pois a solução irá funcionar em português e inglês.
- **Services** – A definição de alguns serviços utilizados pelas páginas. (Ex: serviços REST)
- **Shared** – Uma pasta com componentes que são comuns a praticamente todos as páginas, de forma a evitar repetição de código HTML.
- **Wwwroot** – Uma pasta genérica que é utilizada pelas aplicações web para expor css e outro tipo de configurações, no entanto aqui serve também para colocar os ficheiros JavaScript necessários para algumas operações de interoperabilidade dinâmicas a DOM.
- **Root** – Na pasta root da solução encontra-se ficheiros para fazer *imports* gerais a solução, configurações de arranque da solução e entre outros ficheiros que são criados por defeito na criação de uma solução do tipo .net core.

5.4 Autenticação e autorização

A solução em si irá permitir colocar alguns campos das páginas criadas em privado, para apenas pessoas com devida autenticação e autorização possam visualizar esses mesmos, situação já abordada anteriormente. O Blazor, por omissão, utiliza os mecanismos de autenticação do ASP.NET Core para conseguir identificar o utilizador. Como existia a necessidade de um mecanismo do género foi utilizado este, denominado de *Identity*, como base para a implementação.

Foram necessárias algumas alterações ao projeto como a adição de novas classes desta implementação e modificações no ficheiro de arranque. Este processo pode ser feito automaticamente com auxílio do *visual studio*, sendo que será necessário depois modificar alguns trechos de códigos. As modificações são necessárias pois a implementação tem como base um projeto de .NET CORE e não Blazor.

O *Identity* utiliza por defeito uma base de dados SQL. No entanto, nesta solução para criação das *landing pages* foi decidido utilizar a base de dados não relacional MongoDB. De forma a que os dados dos utilizadores e os dados das *landing pages* se encontrassem na mesmo tipo de base de dados foi decidido alterar a implementação do *Identity* para utilizar também esta base de dados.

Para este caso em específico não seria necessário modificar muito a implementação, mas mudar apenas umas das várias camadas que o constituem, neste caso, a fonte de dados. Para conseguir atingir este objetivo foi necessário recorrer a uma biblioteca. Apesar de existirem algumas bibliotecas disponíveis como a *AspNetCore.Identity.MongoDB* e *AspNetCore.Identity.MongoDbCore* estas encontravam-se com versões antigas da implementação do *Identity*, apresentavam problemas e tinham uma maior complexidade de implementação [99][100]. Portanto, após serem utilizadas e testadas, foi escolhida a biblioteca *AspNetCore.Identity.Mongo* [101].

Para substituir e implementar esta biblioteca foi necessário modificar o ficheiro de arranque e modificar definições por defeito do Identity. No arranque é ainda possível colocar algumas opções para a criação das contas como o número de dígitos da palavra passe, regras de letras e dígitos e confirmações de conta via email. O código implementado encontra-se na Figura 29 abaixo.

```
services.AddIdentityMongoDbProvider<MongoUser, MongoRole>(identityOptions =>
{
    identityOptions.Password.RequiredLength = 6;
    identityOptions.Password.RequireLowercase = false;
    identityOptions.Password.RequireUppercase = false;
    identityOptions.Password.RequireNonAlphanumeric = false;
    identityOptions.Password.RequireDigit = false;
    identityOptions.User.RequireUniqueEmail = true;
    identityOptions.SignIn.RequireConfirmedAccount = true;
}, mongoIdentityOptions => {
    mongoIdentityOptions.ConnectionString = "mongodb://localhost/myDB";
})
.AddDefaultTokenProviders()
.AddErrorDescriber<LocalizedIdentityErrorDescriber>()
.AddDefaultUI();
```

Figura 29 - Extrato do código para implementar Identity com MongoDB (Startup.cs)

No entanto esta não é única ação necessária para conseguir completar esta implementação. Se observarmos as classes adicionadas pelo *Identity*, iremos observar que o mesmo é injetado nestes ficheiros através da seguinte linha de código:

```
@inject SignInManager<IdentityUser> SignInManager
@Inject UserManager<IdentityUser> UserManager
```

Figura 30 - Extrato código da injeção do utilizador nas páginas *Identity*

Como observamos anteriormente, ao implementar esta biblioteca no ficheiro de arranque é enviado como parâmetro *MongoRole*, *MongoUser*. Portanto é necessário atualizar todas as classes do Identity existente com a referência correta, como podemos ver no código abaixo. Após esta alteração todo o código é funcional com a base de dados Mongo.

```
@inject SignInManager<MongoUser> SignInManager
@Inject UserManager<MongoUser> UserManager
```

Figura 31 - Extrato de código da alteração necessária a injeção do utilizador

5.5 Criação das Landing Pages

A página de criação das páginas é uma das mais importantes desta solução, portanto é importante referir como foram implementados alguns dos seus detalhes. Inicialmente, perante a metodologia adotada para esta implementação toda a configuração disponível para cada componente encontra-se disponível na base de dados. Dessa forma, inicialmente, ao abrir a página, todos esses dados são obtidos da base dados para um único documento do tipo *Bson*, uma estrutura para tratar dados provenientes de bases de dados Mongo. Através deste pedido é possível também perceber que componentes se encontram disponíveis.

5.5.1 Configurações gerais (*Header e Footer*)

Nesta página são apresentados por omissão dois componentes que estão sempre presentes na criação das páginas. Um componente de seção inicial que permite personalizar alguns itens da barra de navegação, logotipos e imagens de apresentação da página. O segundo componente permite personalizar o *footer* da página e colocar alguns itens no mesmo. O HTML necessário para estes dois componentes base são definidos em subcomponentes para uma melhor organização do código, que por sua vez perante a sintaxe do blazor, são inicializados através do nome atribuído ao ficheiro neste caso *GeralComp* e *FooterComp*.

5.5.2 Componentes do corpo da página

O mais importante na criação da página é conteúdo do corpo. No iniciar da página, como já referido, são carregados os componentes disponíveis para um campo de seleção. O utilizador poderá escolher aqui aquele que pretende adicionar para personalizar. O valor selecionado

pelo utilizador é obtido através de interoperabilidade com a DOM através da JavaScript, como podemos observar no exemplo abaixo com um trecho de código de um dos métodos desta página.

```
valor = await JsRuntime.InvokeAsync<string>("elementoselect");
```

Figura 32 - Exemplo de interoperabilidade

Depois da opção ter sido selecionada o método “getFormato(valor)” é chamado. Este método vai utilizar o documento obtido inicialmente e fazer um tratamento sobre ele para carregar as definições corretas para o valor selecionado, como podemos observar abaixo por um trecho de código de um dos métodos que faz esse tratamento. São aqui carregadas configurações de tamanho, cores e outros tipos necessários. Para o fazer o documento é convertido num JObject que depois permite aceder a elementos pelo seu nome.

```
var jsonConvert = JsonConvert.SerializeObject(BsonTypeMapper.MapToDotNetValue(documento));
JObject jo = JObject.Parse(jsonConvert);

JObject formato = (JObject)jo[0];
JObject formato1 = (JObject)jo[1];
JObject formato2 = (JObject)jo[2];
var formatos = formato.ToString();
var formatos1 = formato1.ToString();
var formatos2 = formato2.ToString();

formato1 = JsonConvert.DeserializeObject<Dictionary<string, string>>(formatos).ToList();
tamanho = JsonConvert.DeserializeObject<Dictionary<string, string>>(formatos1).ToList();
cor = JsonConvert.DeserializeObject<Dictionary<string, string>>(formatos2).ToList();
```

Figura 33 - Tratamento do documento obtido da base de dados para obter dados para o componente selecionado

O componente é depois adicionado com um clique num botão de confirmação por parte do utilizador. Aqui é criada uma nova instância de um objeto com as características do elemento a adicionar à página e que pretendemos personalizar. Esta instância é inserida de imediato numa lista. Todos os componentes adicionados e as suas opções de escolha ficam gravadas na lista. De acordo com a sintaxe do Blazor é possível misturar código C# com o HTML. Isto faz com que seja possível percorrer uma lista através de um ciclo e adicionar dinamismo à solução. O objetivo aqui é que seja possível adicionar e remover os componentes a configurar de forma dinâmica. A Figura 34 evidencia como foi alcançado e cumprido este objetivo.

```

<div class="container" id="colar">
  @foreach (var componente in componentes)
  {
    string tmp1 = componente.nome;
    int tmp2 = componente.numo;
    var cf = componente.cformatol;
    var ct = componente.ctamanho;
    var cc = componente.ccor;
    switch (componente.nome)
    {
      case "Input":
        if (componente.Adicionado == false) {...
        <InputComp valor=@tmp1 OnClick="removerDiv" acremove=@tmp2 formatol=@cf tamanho=@ct
        <div class="space-10"></div>
        break;
      case "Email":
        if (componente.Adicionado == false)
        {
          acremove++;
          componente.Adicionado = true;
        }
        <EmailComp valor=@tmp1 OnClick="removerDiv" acremove=@tmp2 formatol=@cf tamanho=@ct
        <div class="space-10"></div>
        break;
        .....
    }
  }
</div>

```

Figura 34 - Imagem de código de adição dinâmica de componentes

Como observado Figura 34, foi efetuado um ciclo dentro de uma “div”. Este ciclo percorre toda lista criada anteriormente e por cada item observa o nome do componente em questão, para que se possa saber de que se trata. Através de um “switch” com o nome de cada elemento é possível perceber se o mesmo é por exemplo um input. Caso o item em questão corresponda a um item no “switch”, este entra no mesmo. Em cada caso do “switch” é renderizado o componente correspondente. Para que não houvesse uma grande desorganização de código HTML, cada componente adicionar contém a sua definição em ficheiros a parte como referido na seção da estrutura do projeto.

Os componentes existentes são: o input (texto com caixa); o áudio (que permite introduzir um url para áudios nos formatos mp3/wav/ogg); o botão (permite ao clique abrir um url, ou servir como submissão de campos de dados); o componente data (permite selecionar a data num calendário); o componente de espaço no tempo (permite selecionar um intervalo de tempo através de datas num calendário); o componente tempo (permite selecionar horas); o componente data e tempo (que permite escolher uma data e tempo de uma só vez); o componente Form (que adiciona um formulário de contacto com campos pré-definidos); o componente Imagens (que permite colocar imagens através de um URL no formato jpg/png/gif); o componente Video (que permite colocar vídeos por URL de certos sites como Youtube/Vimeo) ; o componente Video (que permite colocar vídeos por URL de certos sites como Youtube/Vimeo); o componente Callback (permite fazer pedidos externos GET para diversos tipos de URL. Ao contrário dos anteriores não verifica o formato, mas o conteúdo devolvido, se o conteúdo devolvido for reconhecido e puder ser tratado pelo sistema um componente com o conteúdo será apresentado).

Cada componente é inicializado pelo nome do seu ficheiro. Os componentes necessitam de receber parâmetros do componente principal para as suas diversas opções de personalização. Para cada um deles é possível alterar formatos, cores, tamanho, posições e outras opções de

personalização. Todos os componentes podem ser colocados em modo privado, desta forma só utilizadores com autenticação e autorização poderão ver o seu conteúdo.

De acordo com a sintaxe e funcionalidade desta framework, sempre que for adicionado ou retirado um componente da lista, a interface é automaticamente atualizada através de um método específico com o nome `StateHasChanged()`.

Existem muitos outros métodos de auxílio e que despoletam ações dentro desta página, no entanto apenas são aqui evidenciados os de maior relevância para que seja possível ter uma noção das ações mais importantes realizadas.

5.5.3 Gravar e pré-visualizar páginas

Na seção final desta página existem duas opções: uma para gravar todas as configurações efetuadas e passar a página seguinte, ou efetuar um pré-visualização do que está a ser criado. De referir que o valor principal de cada componente (como por exemplo o URL de para um vídeo), é gravado no próprio URL de acesso a página que é codificado de maneira segura para o uso em questão, uma funcionalidade pedida pela empresa requerente. Portanto, os restantes dados são enviados para a API que os guarda na base de dados. Na API são recebidos e inseridos os dados, como é possível observar pelo exemplo do trecho do código da Figura 35.

```
var gg = "1";
try {
    _Infos.InsertOne(g);
    var id = g.First().ToString();
    return id;
}
catch (Exception ex)
{
    return null;
}
```

Figura 35 - Exemplo de inserção de documento na base de dados

Para que se possa pré-visualizar como a página a contruir se encontra, foi necessário fazer uma consulta e gravação na base de dados, pois como já observamos anteriormente é assim que encontramos o elemento correspondente ao selecionado. Como este documento é apenas provisório e deve ser descartado para não ocupar espaço na base de dados, foi utilizada uma funcionalidade da base de dados Mongo. Essa funcionalidade chama-se *Time to live (TTL)* e permite que um documento que seja inserido na base de dados expire automaticamente após um determinado tempo definido. Para que se consiga atingir este propósito é necessário inserir no documento um campo do tipo data com o nome "expireAt". Esse campo é depois indexado como um campo que deve fazer o campo expirar, numa dessas opções é possível determinar quanto tempo depois da data inserida na base de dados, o documento deve ser eliminado. Neste caso após dois minutos o documento desaparece, como é possível observar na Figura 36.

```

public string createTTL(BsonDocument g)
{
    var keys = Builders<BsonDocument>.IndexKeys.Ascending("expireAt");
    var model = new CreateIndexModel<BsonDocument>(keys, new CreateIndexOptions() { ExpireAfter = new TimeSpan(0, 1, 0) });
    try
    {
        _Infos.Indexes.CreateOne(model);
    }
}

```

Figura 36 - Trecho de código da inserção de um documento utilizando a funcionalidade de *TTL*

5.5.4 Diagrama de sequência

Foi elaborado um diagrama de sequência para que se possa perceber a informação referido ao longo desta seção. Neste diagrama (Figura 37), estão presentes os seguintes componentes: CreateForm, EditModalComp, InfoService, MongoDB, e Interop.js. O CreateForm, é o componente principal deste caso de uso onde se encontra localizado a maior parte do HTML necessário e funções para a realização de ações. Este componente encontra-se no formato razor (.cshtml). Este tipo de formato permite juntar html com código c#, código este tipicamente executado no lado do servidor o que resulta em código dinâmico [102]. O componente EditModalComp serve para definir e abrir um modal adjacente ao componente principal CreateForm. Está também presente o componente InfoService, um componente REST que trata das comunicações com a base de dados. O componente de base de dados, neste caso, usa o MongoDB. Por fim, temos um ficheiro JavaScript que serve para definir qualquer método JavaScript. Existe a necessidade destes métodos pois a framework Blazor, apesar de tentar eliminar/diminuir a utilização de JavaScript, ainda se encontra limitada em alguns aspetos, nomeadamente no acesso ao DOM.

De seguida, são explicadas as ações identificadas no diagrama de sequência (Figura 37):

- **Ação 1** – Inicialmente ao entrar nesta página, as configurações/estilos disponíveis para customização são carregadas. Essa informação encontra-se na base de dados, portanto é necessário indicar ao componente REST InfoService para fazer esse pedido à base de dados e devolver a informação de volta ao componente principal deste caso de uso (CreateForm). Após os dados serem recebidos no componente estes são tratados. Devido à sintaxe utilizada pelas razor pages, que permitem associar dados ao html, após o tratamento dos dados a página irá automaticamente refrescar a sua interface através do método StateHasChanged().
- **Ação 2** – Depois de ser apresentada a página ao utilizador, o mesmo pode preencher o formulário de configurações gerais (url do logotipo, url de imagem de fundo, título da página e entre outros).
- **Ações 3,4 e 5** – O utilizador pode editar as permissões da página, ou seja, quem pode ver ou administrar a página. Para o fazer o mesmo carrega num botão na página chamado “Editar Permissões”. Após o clique é enviado um sinal ao componente “EditModalComp”, este inicia e abre dentro do componente principal “CreateForm”.

Quando o componente “EditModalComp” inicia permite ao utilizador preencher o mesmo com um novo email e tipo de permissão e submeter os dados se assim o desejar. Depois da informação ser submetida e trabalhada é devolvida ao componente principal “createForm”, que por sua vez irá atualizar a interface com os dados recebidos através do método StateHasChanged().

- **Ações 6 e 7** – O utilizador pode selecionar através de uma lista no componente principal “createForm” os componentes de página a adicionar. Os dados para preencher a lista são trabalhados através de ação inicial de carregar os dados. Para se conseguir saber qual o elemento selecionado é necessário recorrer a um método JavaScript para aceder ao valor na DOM HTML. O valor volta depois ao componente principal onde os dados são novamente trabalhados para apresentar as opções de customização caso o utilizador opte por adicionar a opção selecionada. Para confirmar a seleção é necessário carregar num botão de confirmação, após esse clique é renderizado o componente correspondente. Por fim, é atualizada a interface através do método StateHasChanged().
- **Ação 8** – Quando todos os campos pretendidos foram adicionados e preenchidos o utilizador pode submeter a configuração. Para submeter o utilizador carrega no botão de submissão no componente principal “createForm”. É assim iniciado o processo de validação dos dados inseridos. Devido ao conteúdo ser gerado dinamicamente a verificação é efetuada via JavaScript, que retorna se os dados estão ou não válidos. Se os dados se encontrarem válidos são recolhidos através de JavaScript. Depois dos dados serem recolhidos da *DOM* e enviados de volta ao componente principal, é efetuado a gravação na base de dados dos mesmos. Para se gravar os dados, enviamos os mesmos para o componente *REST* “InfoService” para fazer esse pedido à base de dados. Se a gravação for bem-sucedida é enviado de volta ao componente principal o identificador do documento. Na base de dados apenas são guardados os aspetos de personalização de cada componente. Os valores principais (como por exemplo o URL de uma imagem) são colocados/guardados no URL de acesso a página. O utilizador é notificado do resultado da operação de gravação na base de dados e se o URL de acesso foi bem formulado. Se tudo correr bem é redirecionado para a próxima página do site, a página de criação de códigos QR.

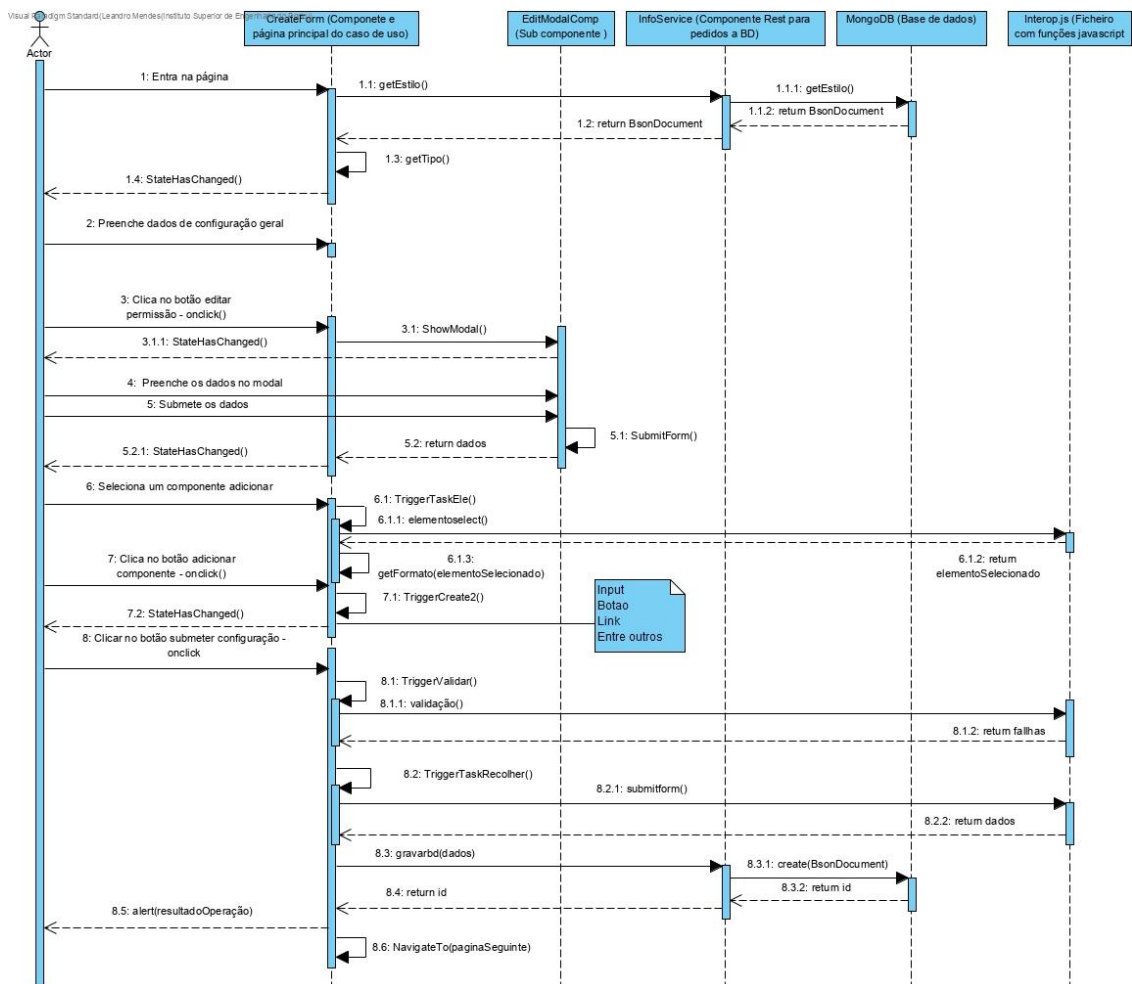


Figura 37 - Diagrama de sequência (Criar uma Landing Page)

5.5.5 Diagrama de classes

Por fim, foi elaborado um diagrama de classes (Figura 38). Para que se possa criar uma Landing Page é necessário definir e criar uma classe/página. A página “CreateForm” é do tipo razor que permite a utilização de HTML e código c# num só ficheiro. Nesta página encontra-se todo o código e HTML principal da página, onde por defeito nos permite personalizar a aparência do *header* e do *footer* da página. Aqui é possível indicar alguns elementos a colocar e a personalizar em ambos. A página permite ainda adicionar componentes, para o fazer utiliza como base uma classe com um conjunto de propriedades definidas, essa classe denomina-se “Componentes”. Nesta classe é possível observar o nome do componente e se encontra adicionado. São ainda carregadas um conjunto de propriedades de customização perante o componente escolhido. Existem diversos componentes possíveis a adicionar, no entanto apenas foi evidenciado na imagem o componente botão de formar a simplificar o diagrama.

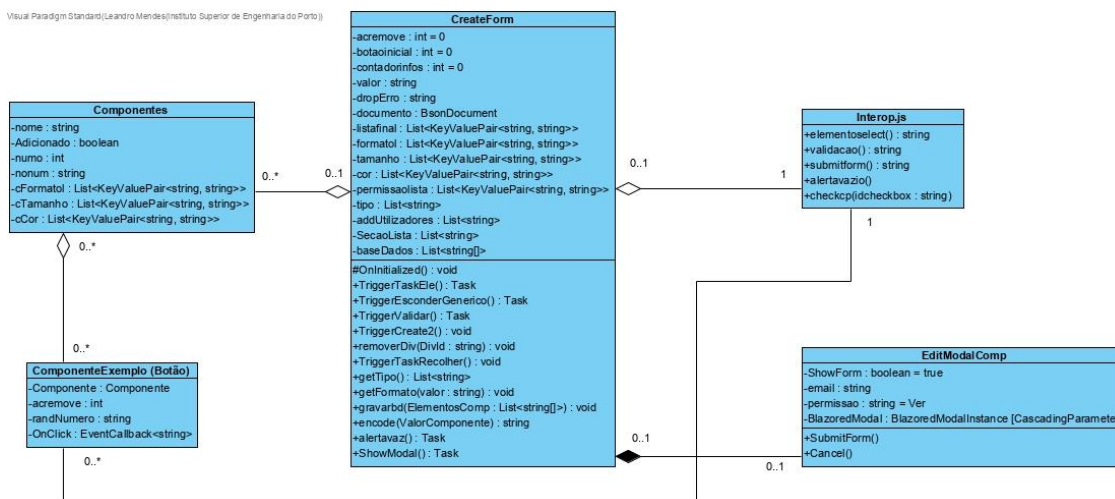


Figura 38 – Diagrama de classe do caso de uso “Criar Landing Page”

A classe principal e os componentes utilizam interoperabilidade de JavaScript para fazer modificações a DOM, esses métodos JavaScript encontram-se na classe Interop.js. Na classe principal é ainda possível definir permissões para os utilizadores, isto é feito com recurso a um modal que pode ser aberto a partir da página principal. Este modal denomina-se “EditModalComp” e contém os seus próprios métodos para que se possa gerir essas permissões e se possa devolver a informação de volta a classe principal.

5.6 Criar código QR

Depois de criada uma *landing page* o URL é enviado para a página de criação de códigos QR. O preencher desta página é obrigatório pois é o único método de acesso posterior a página criada. Nesta página apesar de existirem uma série de métodos e alterações a interface, o mais importante é perceber como são gerados os códigos QR.

Inicialmente, foi observado a necessidade de fazer uma implementação completamente nova ou utilizar uma das várias bibliotecas já existentes. Após uma pesquisa foram encontradas algumas alternativas bastante completas como: o *ZXing*, que permitia gerar códigos de barras e códigos QR com diversos portes para várias linguagens e o *IronBarcode*, uma biblioteca que permitia gerar códigos com imagens, cores diferentes e outras funcionalidades, mas com licença paga. Existem ainda outras alternativas não evidenciadas que possibilitam a criação destes códigos.

5.6.1 Biblioteca de códigos QR seleccionada (*QRCoder*)

Para esta solução foi decidido utilizar a biblioteca *QRCoder*, pois esta satisfazia as necessidades da solução. A biblioteca é implementada em C#/.NET e não tem dependência de outras bibliotecas. Esta foi escolhida para esta solução porque: primeiro é implementada em

.NET tal como a Framework utilizada Blazor, ou seja, é possível aplicar a mesma diretamente na solução sem a necessidade de construir uma API; segundo permite cumprir a maior parte dos objetivos definidos como criar códigos normais, inversão de cores, escolha do nível de correção e criação do códigos com imagens embutidas (*FrameQR*); por fim porque da pesquisa efetuada é a que contém a melhor documentação e permite criar os códigos com apenas algumas linha de código.

Nesta página na sua funcionalidade mais básica irá ser possível criar um código QR, com o URL para apontar para a *landing page* criada anteriormente e com um nível de correção selecionado pelo utilizador. Um trecho do código resumido para atingir este objetivo pode ser observado na Figura 39. Para gerar um código é necessário criar uma instância do objeto que a biblioteca implementa, a essa instância é depois enviado o URL codificado representado pela variável "real" e também o nível de correção escolhido, neste caso o nível "L". Por fim, gera-se o código com a informação enviada e é gerado um Bitmap que representa o código QR.

```
QRCodeGenerator qrGenerator = new QRCodeGenerator();  
qrCodeData = qrGenerator.CreateQrCode(real, QRCodeGenerator.ECCLevel.L);  
QRCode qrCode = new QRCode(qrCodeData);  
qrCodeImage = qrCode.GetGraphic(5);
```

Figura 39 - Código para gerar um código QR básico

A forma como se gera o *qrCodeData*, ou seja, a informação do código contém sempre os dois parâmetros referenciados anteriormente. No entanto, o utilizador pode também escolher alterar ambas as cores do código, inserir uma imagem no centro escolhendo o seu tamanho perante a original e se contém bordas e pode ainda desligar ou ligar a zona de sossego (*Quiet Zone*). Nessa situação, o *qrCodeImage* vai alterar perante a escolha do utilizador, recebendo como parâmetros as opções anteriores. Abaixo, na Figura 40 é representada uma situação em que o utilizador opta por alterar todas definições anteriores:

```
qrCodeImage = qrCode.GetGraphic(5, "#ffffff", "#ffffff",  
    ImagemCentroEmBitmap, TamanhoImagem, TamanhoBordaImagem, true);
```

Figura 40 - Código exemplo de todas as opções de geração de código selecionadas.

As cores são enviadas em formato RGB, traduzidas de uma paleta de cores selecionada pelo utilizador. A imagem que se pode colocar no centro é selecionada localmente pelo utilizador e é também transformada em Bitmap. Os tamanhos da imagem central, são em percentagem perante o tamanho da original. Por fim, é apenas decidido se é utilizado ou não a *Quiet Zone* com um booleano.

O bitmap do código final obtido permite fazer a transformação do código em vários formatos diferentes, onde é oferecido ao utilizador a escolha do formato pretendido para efetuar o download do código.

5.6.2 Diagrama de seqüência

Na imagem (Figura 41), estão presentes os seguintes componentes: o CriarQR, Interop.js e DownloadModalComp. O CriarQR, é o componente principal deste caso de uso onde se encontra localizado a maior parte do HTML necessário e funções para a realização de ações, este componente encontra-se no formato razor (.cshtml). O componente DownloadModalComp, que serve para definir e abrir um modal adjacente ao componente principal CriarQR. Por fim, temos um ficheiro JavaScript que serve para definir qualquer método JavaScript, tal como já foi referido anteriormente.

De seguida, são explicadas as ações identificadas no diagrama de seqüência (Figura 41):

- **Ação 1** – O utilizador inicialmente entra na página, ao entrar na página o componente principal (CriarQR) recolhe os parâmetros enviados por URL e trata esses dados. Esses dados são descriptados e decodificados. Depois de trabalhados os dados, a interface inicial para criar um código QR é apresentada através do método StateHasChanged(). Como já referido este método refresca a interface automaticamente.
- **Ação 2** – Numa segunda fase o utilizador seleciona o nível de correção a utilizar para gerar o código QR.
- **Ações 3 e 4** – Numa fase posterior, o utilizador pode efetuar várias escolhas opcionais, sendo que uma delas é mudar/inverter ambas as cores do código a gerar. Inicialmente, o utilizador seleciona através do clique num interruptor no componente principal que quer ativar a opção de troca de cor. Ao efetuar o clique as opções de escolha necessitam de ser geradas dinamicamente com auxílio de métodos JavaScript. Portanto, o componente principal envia um sinal através de um método seu para um método JavaScript (no componente/página Interop.js), que por sua vez realiza a ação de adicionar os novos campos dinamicamente. Por fim, o método JavaScript atualiza a interface do utilizador.
- **Ações 5,6 e 7** – Esta é uma das outras escolhas opcionais, sendo que aqui o utilizador decide se quer colocar uma imagem do código. Inicialmente, o utilizador seleciona através do clique num interruptor no componente principal que quer ativar a opção de colocar uma imagem. Ao efetuar o clique as opções de escolha necessitam de ser geradas dinamicamente com auxílio de métodos JavaScript. Portanto, o componente principal envia um sinal através de um método seu para um método JavaScript (no componente/página Interop.js), que por sua vez realiza a ação de adicionar os novos campos dinamicamente. Por fim, o método JavaScript atualiza a interface do utilizador. Após a interface ser atualizada o utilizador deve indicar o tamanho da imagem em percentagem relativo ao centro do código QR. Para adicionar uma imagem o utilizador clica num botão que vai permitir ao mesmo escolher um ficheiro, depois escolher um ficheiro o componente principal (CriarQR) irá processar e guardar

o mesmo numa MemoryStream, assim como os detalhes do mesmo. Após o processamento o ficheiro/imagem selecionado é demonstrado ao utilizador através da atualização da interface como o método StateHasChanged ().

- **Ação 8** – Outra das escolhas opcionais é a remoção da imagem selecionada. O utilizador clica num botão que por sua vez remove a imagem. A interface é assim posteriormente atualizada com o método StateHasChanged().
- **Ação 9** – O utilizador clica num botão no componente principal (CriarQR) para gerar o código QR. Quando o utilizador clica no botão um método é despoletado no componente principal, que por sua vez, executa uma série de outros métodos. Aqui irá ser gerado o URL de acesso a Landing Page, URL esse que irá ser encriptado/codificado. Esse URL é posteriormente guardado no código gerado com as opções selecionadas. Após o código ter sido gerado a interface é atualizada, com a uma pré-visualização do aspeto do código
- **Ação 10 e 11** – Por fim, o utilizador pode optar por fazer download do código gerado. Para o fazer deve carregar num botão de download no componente principal (CriarQR). Após o clique é enviado um sinal a um componente modal (DownloadModalComp), este inicia e abre dentro do componente principal (CriarQR). Quando o componente modal (DownloadModalComp) inicia permite ao utilizador preencher o mesmo com um nome do ficheiro, o formato do ficheiro (Ex: .png, .jpeg) e submeter os dados se assim o desejar. Depois da informação ser submetida e trabalhada, o download do ficheiro é iniciado para a pasta por defeito de downloads do browser.

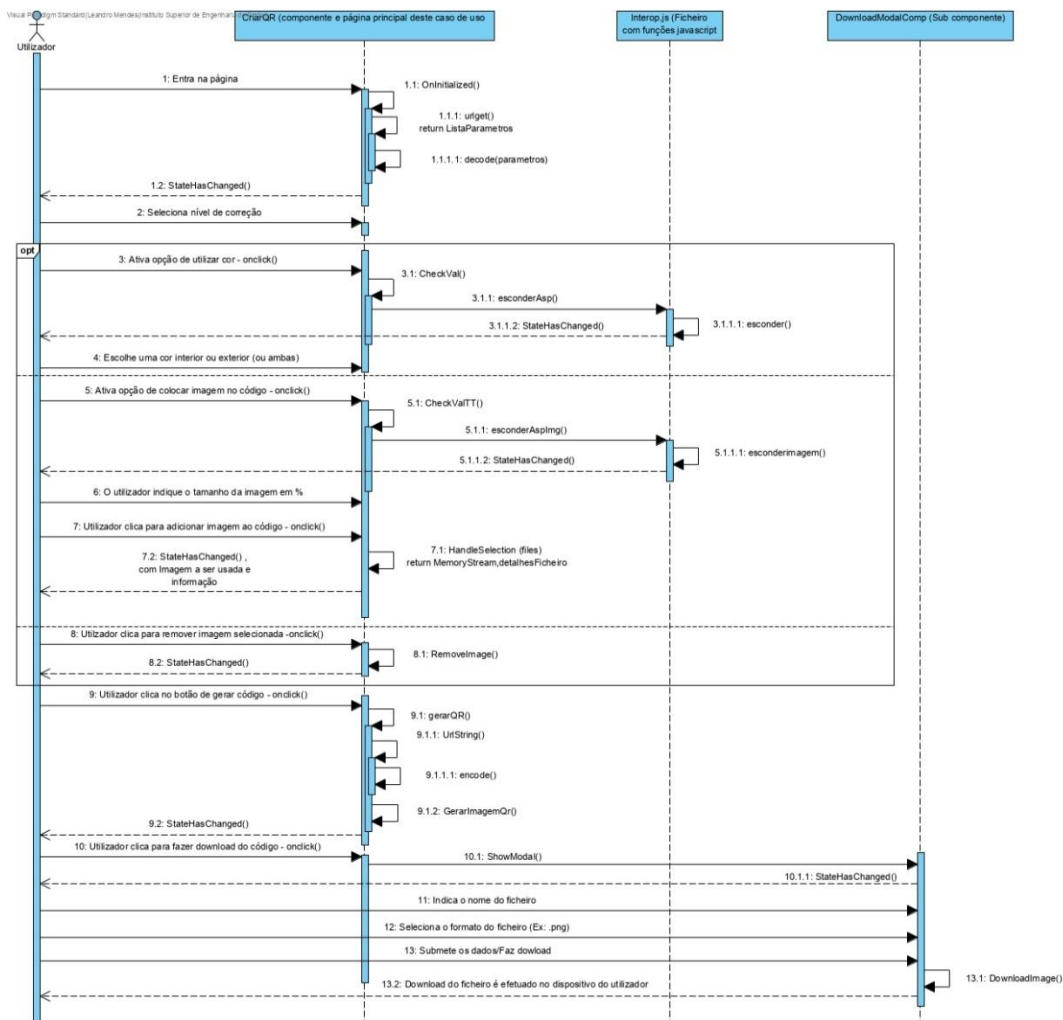


Figura 41 - Diagrama de sequência (Criar código QR)

5.6.3 Diagrama de classes

Por fim, foi elaborado um diagrama de classes (Figura 42). A página/classe “CriarQR” é do tipo razor, tal como referido anteriormente, e é necessária para criar os códigos QR. Esta classe implementa um barra de navegação e *footer* sem qualquer parametrização. A classe principal utiliza ainda interoperabilidade de JavaScript para algumas modificações na DOM. A classe principal, utiliza a biblioteca *QRCode* para conseguir gerar os códigos QR. Esta biblioteca implementa uma classe denominada *QRCodeGenerator* que contém todo o código necessário para implementar a funcionalidade de geração de códigos QR.

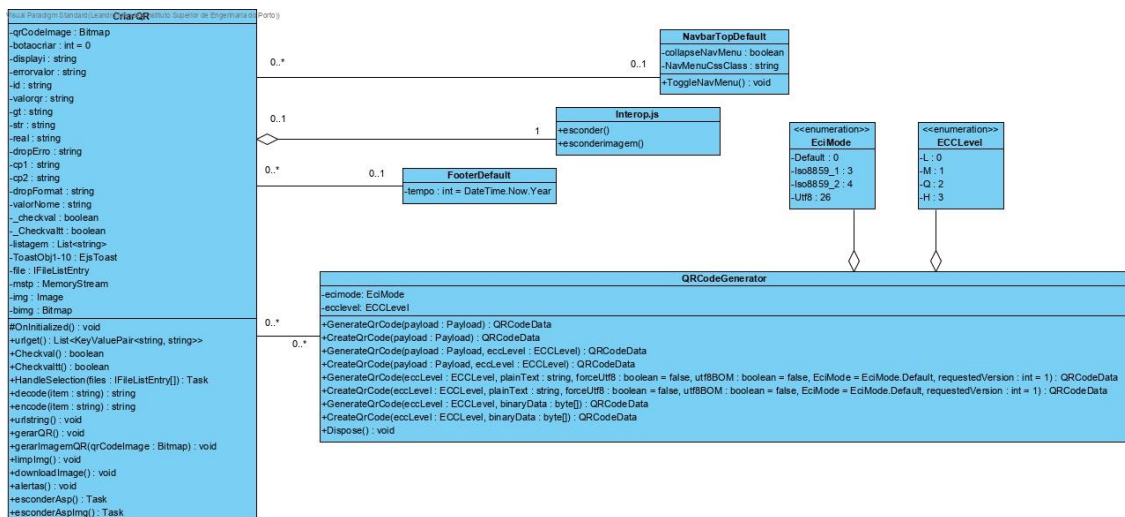


Figura 42 - Diagrama de classe do caso de uso (Criar código QR)

5.7 Visualizar Landing Pages

Na página de visualização das *landing pages* é possível observar um comportamento semelhante ao de criação das mesmas. Numa fase inicial, os dados configurados e personalizadas pelo utilizador são carregados da base de dados através de um pedido a API REST, que devolve um documento do tipo *Bson* com a maior parte dos dados pretendidos. Os valores principais são obtidos através do URL guardado no próprio código, como abordado anteriormente.

A semelhança do que acontecia na página de criação, por cada item do documento, um componente correspondente é renderizado. Isto é possível através da sintaxe da framework com um ciclo pela lista embutido entre o HTML. Cada componente aqui contém também a sua própria definição num ficheiro a parte e recebe da página principal o item correspondente daquele momento do ciclo.

5.7.1 Pedidos externos/remotos

No entanto, existe uma situação especial que são os pedidos externos ou remotos. Com estes pedidos não sabemos de antemão o que nos irá ser devolvido. Para isso, quando percebemos que o item é do tipo “Callback” não é logo renderizado um componente, mas realizada uma ação diferente. Neste caso, por defeito é colocado um campo de carregamento para dar a entender ao utilizador que o pedido está a ser processado, pois não sabemos o espaço temporal para ação ser concluída. Este item é também adicionado a uma lista diferente, onde só são colocados itens deste tipo.

Num dos métodos de renderização do Blazor, foi necessário fazer modificações para que se possa chamar um método que irá efetuar os pedidos externos. A lista dedicada a estes

pedidos é percorrida e são obtidos os resultados como: a resposta obtida e o tipo da mesma. Esta renderização é efetuada assincronamente para que os restantes elementos possam continuar a ser colocados na interface, enquanto outros aguardam resultado. No trecho do código da Figura 43 é possível observar aquilo que é afirmado anteriormente.

```
protected override async Task OnAfterRenderAsync(bool firstRender)
{
    if (firstRender)
    {
        Debug.WriteLine("tou a verdadeiro");
        foreach (var item in tpmcb)
        {
            bool checkft = await WebApiResponse(item);
            vc.Add(new VerCallback
            {
                estadoResposta = checkft,
                tipoResposta = responsetp,
                conteudoResposta = responsevalue,
            });
            Debug.WriteLine(item);
        }
        // firstRender = false;
        StateHasChanged();
    }
    else
    {
        Debug.WriteLine("tou a ver falso sempre");
    }
}
}
```

Figura 43 - Código do método de ciclo de vida de renderização modificado

Os pedidos são efetuados através de uma simples chamada por HTTP, que ocorre num método chamado WebApiResponse. No entanto, muitas vezes no momento do pedido o serviço pode estar indisponível, para evitar considerar a resposta como não obtida ou nula, a ação do pedido é efetuada pelo menos 3 vezes para que possa evitar indisponibilidades ou erros momentâneos. Um exemplo dessa implementação pode ser observado no código abaixo.

```
public async Task<bool> WebApiResponse(string check)
{
    string resultAsString;
    HttpResponseMessage response;
    var tentativas = 3;
    do
    {
        tentativas--;
        try
        {
            response = await Http.GetAsync(check);
        }
        catch
        {
            checkft = false;
            responsetp = "erro";
            responsevalue = null;
            return checkft;
        }
        if (response.IsSuccessStatusCode)
        {
            break;
        }
    }
    while (tentativas > 0);
    if (response.IsSuccessStatusCode)
    {
        checkft = response.IsSuccessStatusCode;
        responsetp = response.Content.Headers.ContentType.MediaType;
        responsevalue = response.Content.ReadAsStreamAsync().Result;
        return checkft;
    }
}
```

Figura 44 - Trecho de código de pedidos externos

Após ter sido obtido o resultado dos pedidos, são verificados se foram obtidas respostas positivas e se correspondem a algum dos tipos de resposta aceites pela solução. Perante o tipo de resposta obtido são renderizados os componentes correspondentes e retirado o componente de carregamento. Desta forma, é sempre possível carregar os componentes com toda a informação já existente na base de dados e aguardar pelos pedidos externos sem nunca bloquear a interface pela resposta de todos os componentes.

5.7.2 Diagrama de sequência

Na imagem abaixo (Figura 45), estão presentes os seguintes componentes: o ReadForm, BlazorIdentity, InfoService, MongoDB. O ReadForm, é o componente principal deste caso de uso onde se encontra localizado a maior parte do HTML necessário e funções para a realização de ações, este componente encontra-se no formato razor (.cshtml). O componente BlazorIdentity, que serve para garantir e gerir a autenticação dos utilizadores. Temos também presente o componente InfoService e MongoDB que já foram referidos e explicados anteriormente.

De seguida, são explicadas as ações identificadas no diagrama de sequência (Figura 45):

- **Ação 1** – Inicialmente ao entrar nesta página é verificado se o utilizador se encontra autenticado. Para isso é utilizado um componente interno que trata de autenticação e autorização: o ASP.NET *Identity*. O utilizador apenas necessita de se autenticar caso queira ver componentes privados dos quais tenha permissão. Este componente comunica com o serviço REST InfoService que por sua vez comunica com a base de dados (MongoDB). Após verificar se o utilizador existe, o mesmo é devolvido para o componente principal (ReadForm) passando por todos os outros até lá chegar. Depois é retirado o valor principal de cada componente (Ex: URL de um vídeo) e o id da página contidos no URL de acesso a mesma. Com o id obtido são carregadas funcionalidades extra da base de dados com recurso ao serviço REST, serviço esse que volta a comunicar com a base de dados e a devolver ao componente principal ReadForm. Os dados recebidos são depois trabalhados para que possam ser demonstrados ao utilizador final. No fim, existem duas opções: caso os dados sejam locais, devido a sintaxe utilizada pelas razor pages que permitem associar dados ao html, após o tratamento dos dados, a página irá automaticamente refrescar a sua interface através do método `StateHasChanged()`; caso sejam remotos o refrescar da interface irá ser assíncrona e irá esperar pelo resultado do pedido externo, após obter resultado atualiza a interface através do método `StateHasChanged()`.

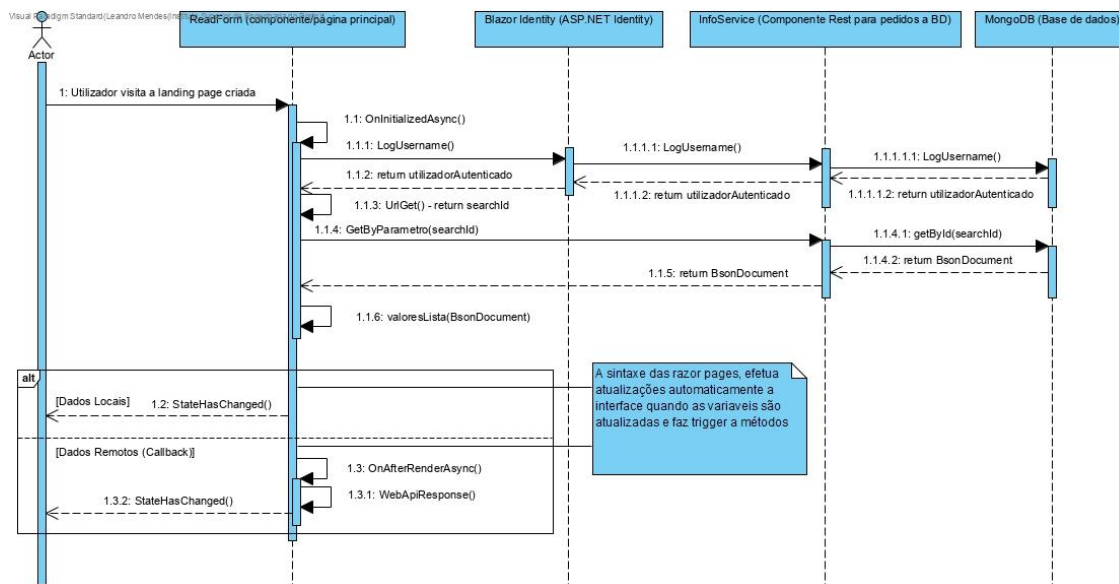


Figura 45 - Diagrama de sequência (Visualizar Landing Page)

5.7.3 Diagrama de classes

Por fim, foi elaborado um diagrama de classes (Figura 46). Para que se possa visualizar uma Landing Page é necessário definir e criar uma classe/página. A página/classe “ReadForm” é do tipo razor, tal como referido anteriormente. Esta é página principal deste caso de uso e onde se encontra a maior parte do código necessário. Para que se possa aceder a esta página é necessário a digitalização de um código QR previamente criado. A página começa por receber dados contidos no próprio código e dados da base de dados acerca da página. Os dados recebidos indicam os componentes a adicionar a página, sendo que todos eles são transformados em classes do tipo Form, desta maneira é possível uma melhor organização para dispor os dados. As configurações da barra de navegação e do footer são as primeiras a ser carregadas e como já indicado são do tipo Form. Os restantes componentes são carregados gradualmente, sendo que foi necessário criar classes adicionais temporárias para certos componentes. Caso o componente seja do tipo Callback foi criada uma classe temporária (VerCallback) para se conseguir verificar o estado do pedido e se retornou algo. Esta classe foi implementada para caso do primeiro pedido falhar, desta forma o sistema pode tentar pelo menos mais 3 vezes o mesmo pedido para garantir que não foi uma falha momentânea. O componente de email implementa também uma classe extra (ExampleModel), pois é um componente complexo que envolve submissão de um formulário. A classe é implementada para que possam ser validados os dados deste formulário, adicionado dinamicamente.

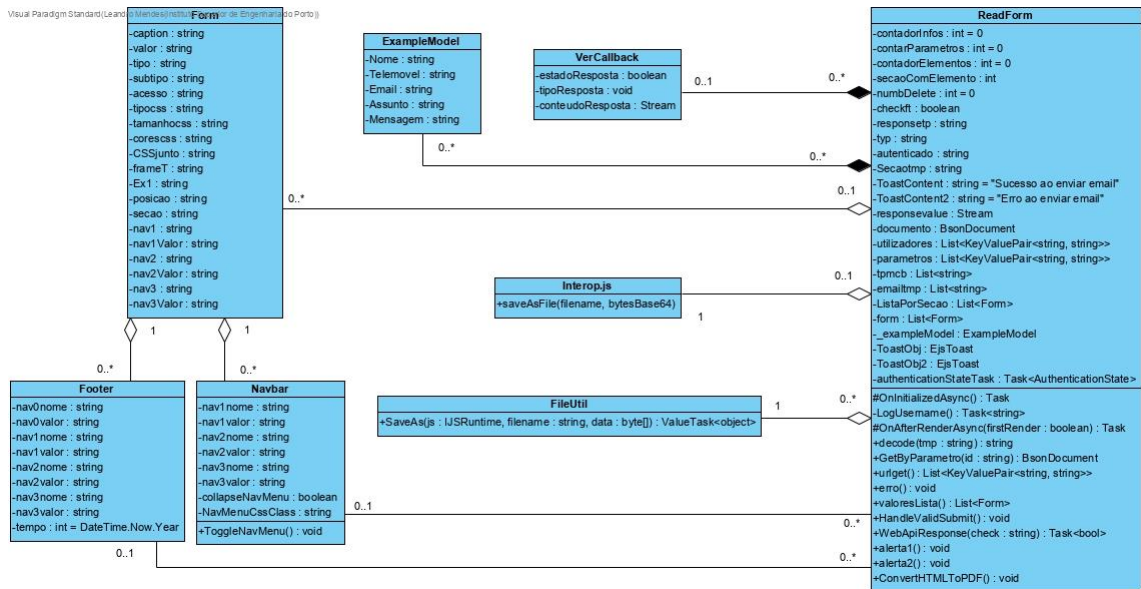


Figura 46 - Diagrama de classe do caso de uso “Visualizar Landing Page”

A classe principal utiliza ainda interoperabilidade de JavaScript e uma classe FileUtil, para quando o componente adicionado necessitar de efetuar um download, seja possível observar o progresso do mesmo.

6 Experimentação e Avaliação

Neste capítulo serão evidenciadas as experiências e avaliações efetuadas na solução. No desenvolvimento de uma solução é importante avaliar se os objetivos foram cumpridos e em que medida através de diferentes técnicas e metodologias. Numa fase inicial deste capítulo serão apresentadas as hipóteses, as grandezas e escolha dos métodos de avaliação adequados. Por fim, são apresentados os resultados obtidos sobre a solução preconizada.

6.1 Hipótese

Para se conseguir avaliar a solução desenvolvida, que consiste no desenvolvimento de uma plataforma personalizada através do uso de códigos QR, é necessário definir hipóteses de avaliação perante a solução preconizada.

6.1.1 1ª Hipótese

É necessário verificar se a empresa que requisitou o desenvolvimento do software, se encontra satisfeita com a solução preconizada e se está de acordo com o que foi idealizado. Esta hipótese é concretizada da seguinte forma.

- **Hipótese Nula (H0)** – A solução desenvolvida não está de acordo com requisitos e objetivos que foram discutidos e idealizados pela empresa requerente.
- **Hipótese Alternativa (H1)** - A solução desenvolvida está de acordo com requisitos e objetivos que foram discutidos e idealizados pela empresa requerente

6.1.2 2ª Hipótese

Além dos objetivos da empresa requerente é necessário verificar se possíveis utilizadores da solução também se encontram satisfeitos com a solução preconizada e se está de acordo com o que as suas necessidades. É então especificada a hipótese:

- **Hipótese Nula (H0)** – Os utilizadores desta solução consideram o sistema incompleto e que não cumpre as suas necessidades.
- **Hipótese Alternativa (H1)** - Os utilizadores desta solução consideram o sistema completo e que cumpre as suas necessidades.

6.2 Grandezas

O objetivo principal desta dissertação é efetuar o desenvolvimento de uma solução web para a Ifthen consoante os objetivos apresentados inicialmente. De forma a que se possa avaliar se foram cumpridos os objetivos e em quais hipóteses o software se enquadra é importante definir as grandezas. As grandezas que melhor se enquadram para se proceder esta avaliação são indicadas de seguida.

6.2.1 Satisfação

O software a desenvolver deve satisfazer a empresa Ifthen pois foi a mesma requisitou o desenvolvimento desta solução. Deve-se verificar também o nível de satisfação de possíveis utilizadores do software.

6.2.2 Eficácia

O objetivo de qualquer software é fazer aquilo que foi delineado para o mesmo fazer. Deve-se verificar se a solução cumpre os objetivos e requisitos propostos.

6.2.3 Eficiência

A solução pode fazer tudo o que é pedido, mas não da melhor forma. É importante perceber os recursos e esforço utilizado por parte dos utilizadores da solução para atingirem os seus objetivos.

6.2.4 Desempenho

A solução sempre que possível deverá se encontrar otimizada em termos de desempenho. Um dos fatores mais importantes de desempenho é o tempo de carregamento de uma página. A

visualização das *Landing Pages* criadas não deve ultrapassar, na sua média, o valor delimitado na especificação dos requisitos não funcionais (seção 4.2) da solução a conceber.

6.3 Metodologia de Avaliação

Para que possa avaliar as grandezas definidas anteriormente e comprovar as hipóteses em estudo é necessário escolher a metodologia que melhor irá responder às questões efetuadas.

6.3.1 Questionários para os requerentes e utilizadores da solução

Perante os casos e hipóteses que pretendemos comprovar a melhor metodologia para utilizar é a utilização de questionários/inquéritos. Estes questionários irão ajudar a averiguar se as hipóteses e grandezas escolhidas são avaliados corretamente. Para ambos os casos, os requerentes e utilizadores da solução optou-se por este método de avaliação.

Existem diversos tipos de questionários possíveis de realizar, no entanto o escolhido foi o SUS (System Usability Scale). Este método foi desenvolvido por Brooke em 1996. Esta escala é conhecida por ser “rápida e suja” no sentido em permite avaliar de forma rápida e acessível o nível de usabilidade de algum sistema ou produto. Apesar de existirem outras escalas o SUS possui diversas características de destaque dos restantes existentes. O mesmo é independente da tecnologia, sendo que pode ser usado desde hardwares até Websites e plataformas [103].

A escala SUS irá permitir verificar as grandezas anteriormente definidas e verificar as hipóteses atingidas. Um estudo realizado por Tullis e Stetson em 2004, demonstra que o SUS é o método que fornece as melhores avaliações de usabilidade para um Website, quando comparado a outros questionários como o QUIS ou o CSUQ [104].

Um questionário SUS normal é constituído por um conjunto de 10 questões. Estas questões estão divididas em 2 grupos, sendo que existem 5 questões positivas e outras 5 questões formuladas de forma negativa. A numeração é utilizada para fazer a distinção, pois os itens com os números ímpares são as questões positivas e número pares as negativas [105].

De acordo com Nielsen, a usabilidade é um adjetivo que permite refletir sobre o quão fácil é utilizar um sistema ou a sua interface. O mesmo acreditava que a usabilidade se focava em 5 pontos [106]. Acredita-se também que essas 5 heurísticas são relacionáveis com as perguntas de um questionário SUS [107]. As 5 heurísticas e as questões a que podemos relacionar são as seguintes [106][107]:

- Capacidade de aprendizagem: Facilidade de completar tarefas básicas, no primeiro uso da solução (Questões: 3,4,7,10).

- Eficiência: Após a aprendizagem, o quão rápido o utilizador consegue concluir tarefas (Questões: 5, 6 e 8).
- Memória: Depois de um período sem usar a solução o utilizador deve conseguir lembrar-se de como a utilizar de forma fácil (Questão: 2).
- Erros: os erros cometidos pelo utilizador, a gravidade do mesmo e se é possível recuperar (Questões: 6).
- Satisfação: Se a solução é agradável, satisfaz o utilizador (Questões: 1, 4, 9).

As 10 questões utilizadas no sistema original foram traduzidas procurando-se manter o sentido das questões. As questões são agora apresentadas [108]:

1. Eu acho que gostaria de utilizar este sistema frequentemente
2. Eu achei que o sistema era desnecessariamente complexo
3. Eu achei que o sistema era fácil de utilizar
4. Eu achei que iria necessitar de ajuda de um técnico de forma a conseguir utilizar este sistema
5. Achei que as várias funcionalidades deste sistema estavam bem integradas
6. Eu achei que existiam demasiadas inconsistências neste sistema
7. Eu imagino que a maioria das pessoas aprenderia a utilizar rapidamente este sistema
8. Eu achei o sistema muito complicado de utilizar
9. Eu senti-me muito confiante a utilizar este sistema
10. Tive necessidade de aprender muitas coisas antes de conseguir entender/utilizar este sistema

Este sistema avalia cada resposta dada, utilizando a escala de Likert. Esta escala varia de 1 a 5, sendo que 1 corresponde a “discordo totalmente” e 5 a “concordo totalmente”. Para que se possa entender melhor esta escala na sua tradução efetiva são apresentadas as respostas possíveis abaixo e a sua escala correspondente [105].

Tabela 23 – Escala de Likert com respostas correspondentes

SUS	Discordo totalmente	Discordo	Neutro	Concordo	Concordo totalmente
Questões do SUS	1	2	3	4	5

É importante também perceber o sistema de pontuação efetuado e aplicado no SUS. Neste sistema a pontuação de cada resposta individual não tem qualquer valor. É importante antes de realizar o cálculo final perceber como é atribuída a pontuação a cada resposta individual. A pontuação de cada resposta vai desde 0 a 4 e varia consoante o seu teor, positivo ou negativo. As questões 1,3,5,7 e 9 obtêm a sua pontuação através da sua posição na escala menos 1. As questões 2,4,6,8 e 10 obtêm a sua pontuação subtraindo a 5 a sua posição na escala [108]. Para melhor se entender o cálculo da pontuação foi elaborada uma tabela representativa, onde n é escala atribuída pela pessoa que esta a responder ao questionário

Tabela 24 - Pontuação de cada resposta perante a escala

Questões do SUS	1	2	3	4	5	6	7	8	9	10
Escala da resposta	n	n	n	n	n	n	n	n	n	n
Pontuação	n-1	5-n	n-1	5-n	n-1	5-n	n-1	5-n	n-1	5-n

Depois de atribuída a pontuação a cada resposta é possível efetuar o cálculo da pontuação total. O cálculo do SUS é obtido primeiro através da soma da pontuação de cada resposta, depois de obtida a soma multiplica-se esse valor por 2.5. O valor obtido é a pontuação total do SU [108].

De referir que o valor obtido é o de uma pessoa que tenha resolvido o questionário. Para obter o valor final para um grupo é necessário somar os valores finais obtidos por cada um e dividir pelo número de pessoas, ou seja, realizar a média. O valor obtido encontra-se entre 0 e 100. Por fim, é possível classificar o nível do produto ou sistema através do valor obtido [109]. Através da Figura 47 podemos classificar o valor obtido:

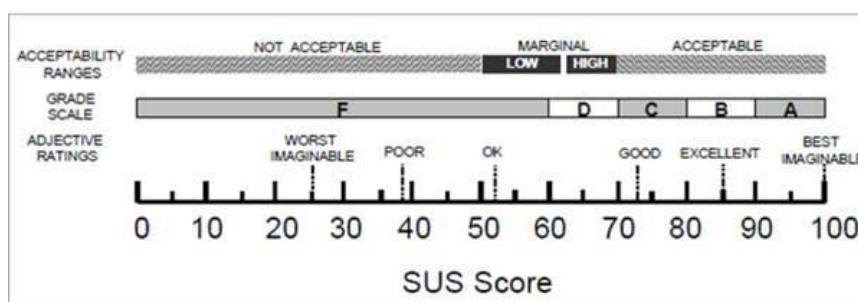


Figura 47 – Classificação do valor obtido no SUS [109]

É importante referir que apesar do valor ir de 0 a 100 não se pode interpretar os valores de um SUS como percentagens. Um questionário SUS permite obter bons resultados numa amostra pequena de entre (8 a 12 utilizadores). Utilizar o SUS permite chegar a uma conclusão correta de forma mais rápida e com elevado nível de consistência quando comparada a outro tipo de questionário existente [109].

Além das 10 questões que constituem um questionário SUS, foi colocada uma questão extra de opinião de aspetos a melhorar na solução preconizada. Esta questão não irá entrar para os cálculos das pontuações, mas irá ajudar a entender melhorias a ser realizadas no futuro a solução.

6.3.2 Formulário de avaliação aos requisitos funcionais

De forma, a que fosse possível avaliar se os requisitos funcionais elaborados na seção 4.1 foram cumpridos, foi elaborado um formulário. O formulário contém todos os requisitos desde o R01 até ao R19 e para cada um deles é possível responder sim ou não, para indicar se estes foram cumpridos. Este formulário é apenas destinado a empresa requerente pois são estes os maiores interessados que os requisitos delineados sejam cumpridos. O formulário foi entregue ao supervisor e a um dos responsáveis máximos da empresa que previamente fizeram uso da solução.

6.3.3 Testes de desempenho

A solução desenvolvida passa por fornecer um serviço aos clientes diretos da empresa requerente, que por sua vez, apresentam o conteúdo criado aos seus clientes (clientes indiretos da empresa). A visualização das *Landing Pages* criadas tem como público alvo os clientes indiretos, portanto é importante perceber o desempenho das mesmas. Se o desempenho for pobre ou deficiente de alguma maneira irá contribuir para queixas ou desistências dos clientes diretos de utilizar este serviço para apresentar informação aos seus clientes (clientes indiretos da empresa).

O fator de desempenho mais importante para esta solução e que irá ser avaliado é a velocidade de carregamento das *Landing Pages* criadas. O tempo de carregamento da página de um website irá ter impacto nos seus visitantes. A Google afirma que a velocidade de carregamento de uma página tem impacto na classificação que atribui em termos de pesquisa e que interfere com a experiência do utilizador. Para um visitante, quanto maior for o tempo de carregamento, mais chance existe de o utilizador abandonar a página sem ver o seu conteúdo [110]. Estima-se que o tempo máximo de espera de um utilizador ronde cerca de 10 segundos e foi este o limite definido para as páginas criadas pela solução [111].

As *Landing Pages* criadas pela solução preconizada não irão ser todas iguais pois cada utilizador que a constrói irá ter controlo sobre que conteúdo irá inserir. De referir, que as páginas criadas a partir da solução irão ter sempre um limite máximo possível, pois o

conteúdo principal de cada componente (Ex: URL de uma imagem) irá no URL de acesso a página. Um URL tem como limite máximo 2,083 caracteres portanto isso irá ditar um limite prático do conteúdo de cada página. Para efetuar estes testes foram então tidas em conta diversas amostras (páginas) diferentes.

A primeira página criada para testar terá então cerca de 80 componentes inseridos, que será uma aproximação do limite máximo possível perante a limitação do tamanho de um URL. Todos os tipos de componentes foram utilizados, com atenção de colocar cada tipo na mesma quantidade. A segunda amostra irá conter cerca de 40 componentes também igualmente distribuídos. A terceira terá cerca de 20 componentes com a mesma quantidade de tipos. A quarta amostra terá cerca de 20 componentes, mas todos eles serão do tipo “Callback”, pois este tipo efetua um pedido externo (GET) e aguarda pela resposta para trabalhar os dados. A implementação deste tipo de componente foi pensada para que o carregamento da página fosse efetuado sempre por defeito e a medida que fosse obtido o resultado esta seria atualizada. Esta amostra irá permitir verificar se esta implementação foi bem efetuada e que o tempo de carregamento inicial não terá sido afetado. A quinta amostra terá 10 componentes também eles igualmente distribuídos.

A ferramenta utilizada foi o *Lighthouse*. Esta é uma ferramenta de automação que procura auditar e melhorar a qualidade de uma página Web. Permite avaliar o desempenho, acessibilidade, melhores práticas e muito mais. No entanto, para o contexto de avaliação apenas será utilizada a opção de desempenho pois é o que se pretende avaliar.

6.4 Resultados

Nesta seção são apresentados os métodos utilizados para obter os resultados e os resultados de cada uma das metodologias adotadas.

6.4.1 Resultados Questionários

No início, foi efetuada uma breve explicação verbal aos inqueridos sobre o tema da solução bem como o problema e objetivos a cumprir, de modo a que os inqueridos pudessem enquadrar no tema.

De seguida, antes da realização dos questionários é necessário que os utilizadores façam uso da solução. Para que este uso seja organizado e de forma semelhante perante todos os intervenientes foram definidas tarefas chave a realizar. Desta maneira, e garantido que a maioria das funcionalidades é efetuada e testada. As tarefas a realizar foram as seguintes:

- Efetuar o registo e autenticação;
- Criar uma Landing Page e um código QR;

- Digitalizar um código QR e visualizar uma Landing Page criada;
- Aceder a seção de gestão e modificar uma permissão.

Após terem sido efetuadas as etapas delineadas anteriormente, procedeu-se a realização do questionário. O questionário teve 11 respostas, um número que se enquadra neste tipo de questionário, como visto anteriormente e que permite efetuar uma boa avaliação. O recuso utilizado para a realização foi o *Google Forms*. Abaixo, foram elaborados gráficos para cada uma das questões elaboradas, para que seja mais fácil observar as diversas respostas e retirar as devidas conclusões.

1- Eu acho que gostaria de utilizar este sistema frequentemente
12 respostas

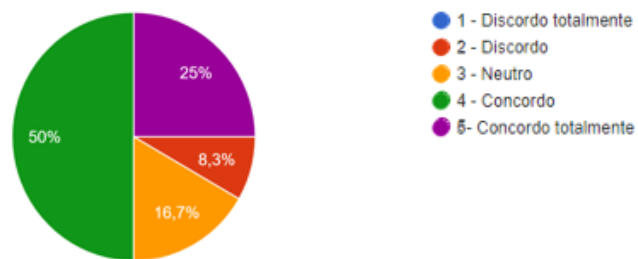


Figura 48 – Q1: Eu acho que gostaria de utilizar este sistema frequentemente

2- Eu achei que o sistema era desnecessariamente complexo
12 respostas

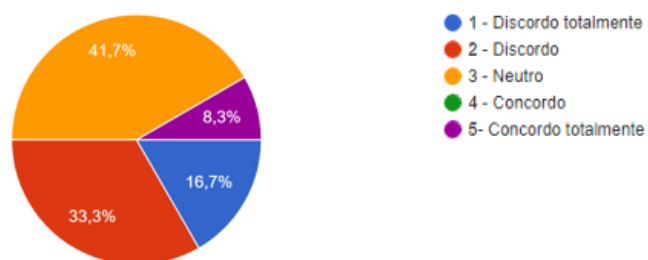


Figura 49 – Q2: Eu achei que o sistema era desnecessariamente complexo

3- Eu achei que o sistema era fácil de utilizar
12 respostas

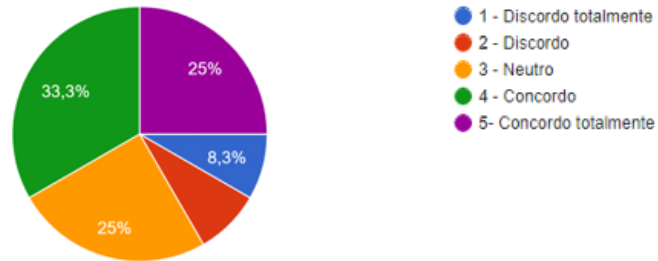


Figura 50 – Q3: Eu achei que o sistema era fácil de utilizar

4- Eu achei que iria necessitar de ajuda de um técnico de forma a conseguir utilizar este sistema
12 respostas

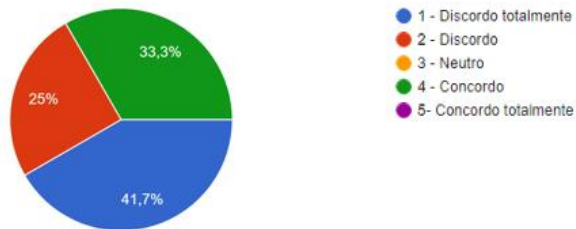


Figura 51 – Q4: Eu achei que iria necessitar de ajuda de um técnico de forma a conseguir utilizar este sistema

5- Achei que as várias funcionalidades deste sistema estavam bem integradas
12 respostas

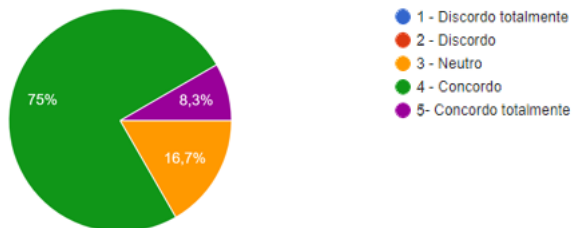


Figura 52 – Q5: Achei que as várias funcionalidades deste sistema estavam bem integradas

6- Eu achei que existiam demasiadas inconsistências neste sistema
12 respostas

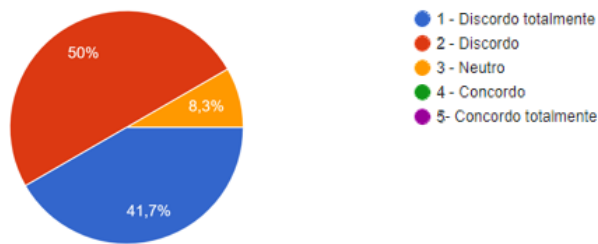


Figura 53 – Q6: Achei que as várias funcionalidades deste sistema estavam bem integradas

7- Eu imagino que a maioria das pessoas aprenderia a utilizar rapidamente este sistema
12 respostas

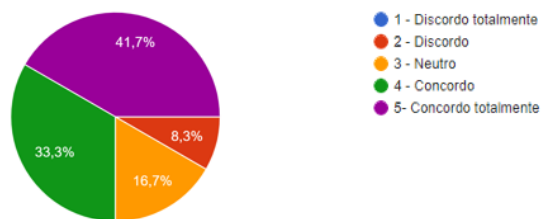


Figura 54 – Q7: Eu imagino que a maioria das pessoas aprenderia a utilizar rapidamente este sistema

8- Eu achei o sistema muito complicado de utilizar
12 respostas

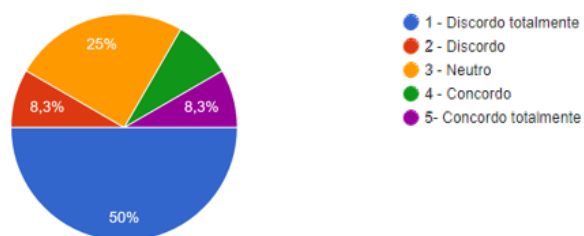


Figura 55 – Q8: Eu achei o sistema muito complicado de utilizar

9- Eu senti-me muito confiante a utilizar este sistema
12 respostas

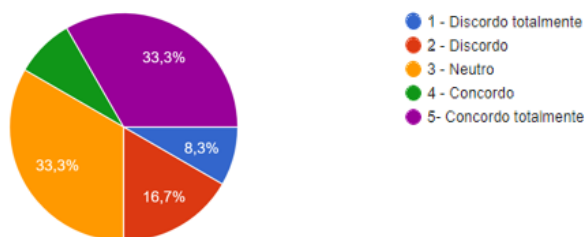


Figura 56 – Q9: Eu senti-me muito confiante a utilizar este sistema

10- Tive necessidade de aprender muitas coisas antes de conseguir entender/utilizar este sistema
12 respostas

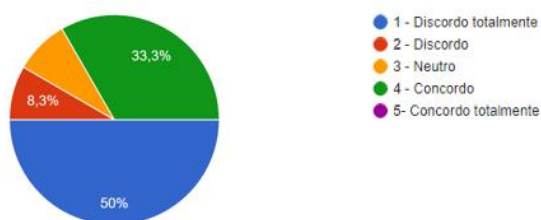


Figura 57 - Q10: Tive necessidade de aprender muitas coisas antes de conseguir entender/utilizar este sistema

Como já foi referido, foi colocada uma questão extra de tipo aberto, para que os utilizadores podem-se dar a sua opinião. O nome da questão colocada foi “Opinião de melhoria” e encontrava-se no fim do questionário. As respostas dadas pelos inqueridos podem ser observados abaixo:

1. “Autenticação por impressão digital.”
2. “Ter sempre os dois métodos de inserir imagens: link e escolhendo o ficheiro pelo computador nos componentes adicionados.”
3. “Achei interessante, se quisesse promover um negócio, seria uma plataforma que poderia utilizar.”
4. “Gostava que algumas perguntas fossem menos vagas ou que existissem notas com mais informação. Gostava também de ter a liberdade de fazer upload das minhas imagens diretamente no website. Acho que o conceito do QR code é bom, mas o facto de não dar para editar o conteúdo sem ter que criar um código novo parece-me ser inconveniente. Penso o mesmo do facto de não existir um url que eu possa usar sem ter que usar sempre o meu telemóvel.”

5. “Para cada suplemento que pretendemos acrescentar à página deveria de haver um tipo de pré-visualizar/ uma imagem a explicar o que estávamos a fazer.”
6. “Não guardar informação no código QR, demasiadas opções para o utilizador leigo, termos demasiado específicos para o mesmo utilizador.”
7. “Colocar campo de informação para explicar melhor cada campo adicionado.”
8. “Acesso direto a página sem ser necessário o uso do código qr em questão.”
9. “É sem dúvida uma ideia muito interessante, mas a falta de fornecimento de um url de acesso ao site criado é um ponto negativo muito importante.”
10. “Renomear os componentes para fácil compreensão e localização dos mesmos. Integração de uma página explicativa”
11. “Utilizar linguagem mais compreensível”
12. “Meter a opção de criar um novo campo mais óbvia, adicionar um separador de gestão de conta e retirar o login (pois pode confundir e as pessoas pensarem que foram logged off)”

6.4.2 Avaliação da pontuação total do questionário SUS

Inicialmente, é importante conseguir medir o nível de usabilidade do produto. Esta medição é efetuada, como referido, com as respostas obtidas e através de um conjunto de cálculos as pontuações obtidas.

O primeiro passo foi conseguir recolher as respostas de cada utilizador da plataforma utilizada de uma forma perceptível e que permitisse efetuar os cálculos. Para isso, foi utilizado uma folha de Excel para exportar e trabalhar os mesmos. Esta é uma definição disponível por defeito no Google Forms.

Depois de ter os dados obtidos por defeito ao exportar a informação do Google Forms, foi adicionada uma nova coluna chamada “Pontuação Crua” (Figura 58). Esta coluna representa a soma das pontuações para cada questão, lembrando que aos itens de número ímpar (1,3,5,7,9) é retirado 1. Para as questões de número par é retirado ao número 5 a pontuação dada.

Depois de terem sido efetuadas as somas para cada uma das questões é ainda necessário multiplicar esse valor por 2.5, esses valores podem ser observados na coluna “Pontuação SUS” (Figura 58). Com todos os valores obtidos é depois efetuada uma média para obter a pontuação total como é possível observar na Figura 58.

Inquiridos	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Q10	Pontuação Crua	Pontuação SUS
Inquirido 1	4	3	4	2	4	2	5	1	4	1	32	80
Inquirido 2	4	3	3	2	4	1	3	3	3	4	24	60
Inquirido 3	3	2	1	4	4	3	3	4	2	4	16	40
Inquirido 4	5	3	3	4	5	2	4	3	3	3	25	62,5
Inquirido 5	4	2	3	1	4	1	4	1	5	1	34	85
Inquirido 6	5	3	5	1	4	1	5	1	5	1	37	92,5
Inquirido 7	4	2	4	2	4	2	4	3	3	1	29	72,5
Inquirido 8	4	1	5	1	3	1	5	1	5	1	37	92,5
Inquirido 9	2	5	2	4	4	2	2	5	2	4	12	30
Inquirido 10	3	1	4	1	4	2	4	1	1	1	30	75
Inquirido 11	4	3	4	4	3	2	5	2	3	4	24	60
Inquirido 12	5	2	5	1	4	1	5	1	5	2	37	92,5
											Pontuação Obtida:	70,20833333

Figura 58 – Resultados obtidos com cálculo da pontuação SUS

A pontuação obtida pela solução através da realização do questionário é 70.20. Como já vimos anteriormente de acordo com a Figura 47, que representa a classificação do valor obtido idealizada por Brooke [109], o valor obtido para a solução preconizada enquadra-se na categoria C. Isto significa que é possível classificar a usabilidade geral desta solução como boa. Os resultados obtidos comprovam que para a 2ª Hipótese, a hipótese alternativa foi cumprida

6.4.3 Avaliação por heurísticas (Questionário SUS)

De acordo com a relação evidenciada anteriormente das heurísticas com as questões, irá ser agora avaliado se a solução cumpre com as heurísticas evidenciadas. As heurísticas são as seguintes.

6.4.3.1 Capacidade de aprendizagem

A capacidade de aprendizagem é relacionada com as questões: 3,4,7,10. Portanto vamos observar cada uma dessas questões de forma individual.

A questão 3 “Eu achei que o sistema era fácil de utilizar”, podemos observar pela Figura 50 que 25% “concordou totalmente”, 33.3% “concordou”, 25% optaram por ficar “neutros”. Ambos o “discordo” e o “discordo totalmente” obtiveram 8.3% cada um. Podemos observar que não houve uma grande concordância em relação a esta questão. Em média, esta questão obteve uma pontuação de cerca de 3.58 segundo a escala de Likert, ou seja, a maior parte achou sistema fácil de utilizar

A questão 4 “Eu achei que iria necessitar de ajuda de um técnico de forma a conseguir utilizar este sistema”, podemos observar pela Figura 51 que 41.7% “discordou totalmente”, 33.3% “discordou” e 25% “concordou”. Em média, esta questão obteve uma pontuação de cerca de 2.25 segundo a escala de Likert, ou seja, a maior parte irá conseguir utilizar o software sem necessitar de ajuda

A questão 7 “Eu imagino que a maioria das pessoas aprenderia a utilizar rapidamente este sistema”, podemos observar pela Figura 54 que 41.7% “concordou totalmente”, 33.3% “concordou”, 16.7% optou por ficar “neutro” e 8.3% “discordou”. Em média, esta questão

obteve uma pontuação de cerca de 4.08 segundo a escala de Likert, ou seja, a maior parte irá conseguir aprender a utilizar as tarefas deste sistema.

A questão 10 “Tive necessidade de aprender muitas coisas antes de conseguir entender/utilizar este sistema”, podemos observar pela Figura 57 que 50% “discordou totalmente”, 8.3% “discordou”, 8.3% “neutro” e 33.3% “concordou”. Em média, esta questão obteve uma pontuação de cerca de 2.25 segundo a escala de Likert, ou seja, a maior parte irá conseguir utilizar o software sem qualquer tipo de conhecimento prévio

Como a relação é perante todas as questões evidenciadas anteriormente e todas obtiveram bons resultados, podemos concluir que o software é de fácil aprendizagem.

6.4.3.2 Eficiência

A eficiência é relacionada com as questões: 5, 6 e 8. Portanto vamos observar cada uma de forma individual.

A questão 5 “Achei que as várias funcionalidades deste sistema estavam bem integradas”, podemos observar pela Figura 52 que 8.3% “concordou totalmente”, 75% “concordou”, 16.7% optou por ficar “neutro”. Em média, esta questão obteve uma pontuação de cerca de 3.91 segundo a escala de Likert, ou seja, a maior parte concorda que todas as funcionalidades estão integradas corretamente.

A questão 6 “Eu achei que existiam demasiadas inconsistências neste sistema”, podemos observar pela Figura 53 que 41.7% “discordou totalmente”, 50% “discordou” e 8.3% optou por ficar “neutro”. Em média, esta questão obteve uma pontuação de cerca de 1.66 segundo a escala de Likert, podemos concluir que estamos perante um sistema consistente na sua utilização

A questão 8 “Eu achei o sistema muito complicado de utilizar”, podemos observar pela Figura 55 que 50% “discordou totalmente”, 8.3% “discordou”, 25% “neutro”, 8.3% “concordou” e 8.3% “concordou totalmente. Em média, esta questão obteve uma pontuação de cerca de 2.16 segundo a escala de Likert, ou seja, a maior parte concorda que o sistema é de fácil utilização

Como a relação é perante todas as questões evidenciadas anteriormente e todas obtiveram bons resultados, podemos concluir que o software é de eficiente.

6.4.3.3 Memória

A memória esta relacionada com a questão: 2. Portanto vamos observar essa questão de forma individual.

A questão 2 “Eu achei que o sistema era desnecessariamente complexo”, podemos observar pela Figura 49 que 16.7% “discordou totalmente”, 33.3% “discordou”, 41.7% optou por ficar “neutro” e 8.3% “concordou totalmente”. Em média, esta questão obteve uma pontuação de

cerca de 2.5 segundo a escala de Likert, ou seja, a maioria concorda que o sistema não é complexo e contém apenas aquilo que é necessário.

Perante a heurística a comparar, podemos afirmar que as tarefas realizadas vão ser pela grande maioria fáceis de memorizar.

6.4.3.4 Erros

Os erros estão relacionados com a questão: 6. Os valores atribuídos a esta questão já foram observados anteriormente sendo que a mesma obteve uma média de 1.66.

Neste caso apenas temos esta questão como relação com a heurística. Podemos concluir que os erros nesta solução são quase inexistentes.

6.4.3.5 Satisfação

A satisfação é relacionada com as questões: 1, 4 e 9. Portanto vamos observar cada uma de forma individual. Para a questão 4, já observamos os valores da média que é de 2.25.

A questão 1 “Eu acho que gostaria de utilizar este sistema frequentemente”, podemos observar pela Figura 48 que 25% “concordou totalmente”, 50% “concordou”, 16.7% optou por ficar “neutro” e 8.3% “discordou”. Em média, esta questão obteve uma pontuação de cerca de 3.91 segundo a escala de Likert, ou seja, a maioria das pessoas iria optar por utilizar muitas vezes este sistema para cumprir com as suas necessidades

A questão 9 “Eu senti-me muito confiante a utilizar este sistema”, podemos observar pela Figura 56 que 33.3% “concordou totalmente”, 8.3% “concordou”, 33.3% optou por ficar “neutro”, 16.7% “discordou” e 8.3% “discordou totalmente”. Em média, esta questão obteve uma pontuação de cerca de 3.41 segundo a escala de Likert, ou seja, a maioria das pessoas sentiu-se confiante a utilizar e realizar as diversas tarefas da solução

Perante a heurística a comparar, podemos afirmar que os utilizadores deste sistema/solução se sentiram satisfeitos. A oferta que o sistema proporciona é correspondente as expectativas apresentadas inicialmente.

6.4.4 Avaliação das repostas à questão aberta

Como referido anteriormente, foi realizada uma questão aberta, onde os utilizadores puderam dar a sua opinião sobre a solução. Desta maneira é possível saber em mais detalhe possíveis modificações ou melhorias a aplicar. Existiram sugestões como: permitir a adição de conteúdo através da seleção de ficheiros locais, pois a solução atualmente nos seus componentes permite apenas a introdução de URLS a apontar para o conteúdo correspondente. Este método foi adotado devido as necessidades e objetivos da empresa, no entanto incluir também a possibilidade de usar os ficheiros locais iria permitir mais diversidade na escolha e possivelmente atrair mais possíveis utilizadores da solução. Portanto, poderá ser uma boa inclusão no futuro.

Um ponto fortemente evidenciado, foi o facto de apenas ser possível aceder a página por código QR, esta foi a decisão tomada desde o início e que pretendia ter uma solução única e eficaz de fazer o acesso a página. Oferecer o acesso a página através de url direto, pode de certa forma destruir o conceito da solução visto que muita da informação principal é guardada no mesmo. No entanto, existir uma alternativa poderá ser equacionada no futuro.

Mais que um inquirido evidenciou também a ideia de criar novos códigos com a mesma informação. Uma funcionalidade que esta ser implementada ao momento da escrita, pois trata-se de algo fulcral que deveria estar incluído nesta primeira versão. Assim, perante a necessidade do utilizador e se o mesmo tiver permissões administrativas sobre a página, quando entrar na mesma através de um código QR, é possível recolher toda a informação e criar um novo código com aspeto diferente a apontar para a mesma página.

Deverá ser ainda criada documentação auxiliar a solução e incluir nos próprios componentes pequenos ícones em que seja possível obter mais informação daquilo que tem de ser feito. Desta maneira, os utilizadores com mais dificuldades poderão sem dúvida ter uma boa experiência. Por fim, deverá ser encontrada também uma metodologia que permita criar novos códigos apenas com a informação do URL diferente e conseguir editar páginas atuais sem deixar que os códigos criados anteriormente deixem de funcionar.

Muitas das sugestões evidenciadas pelos utilizadores são de grande relevo e de implementação futura quase seja dada continuidade a solução. Sendo que algumas delas já se encontram em desenvolvimento ao momento da escrita.

6.4.5 Resultado e avaliação do formulário de avaliação requisitos

Após os responsáveis da empresa responderem ao formulário é possível avaliar os seus resultados. Este formulário foi também efetuado com recurso ao *Google Forms* e serviu como uma espécie de *checklist* aos requisitos. Ambos os intervenientes responderam que todos os requisitos funcionais delineados inicialmente para esta solução foram cumpridos a 100%. Os resultados obtidos comprovam que para a 1ª Hipótese, a hipótese alternativa foi cumprida.

6.4.6 Resultado dos testes de desempenho

No segmento de desempenho a ferramenta escolhida permite obter medições através de 6 métricas distintas, a qual foram atribuídos letras simbólicas [112]. A primeira métrica é primeiro preenchimento de conteúdo “**M1**” (assinala o primeiro momento de preenchimento com um texto ou imagem); o índice de velocidade “**M2**” (indica a velocidade com que foi preenchido visualmente todo o conteúdo da página, que por sua vez é a métrica mais importante); maior preenchimento de conteúdo “**M3**” (o momento que assinala quando a página foi preenchida com o maior texto ou imagem); tempo até interação “**M4**” (tempo que demora para a página estar totalmente interativa); tempo de bloqueio total “**M5**” (indica o

tempo que a página fica bloqueada a responder a inputs do utilizador, cliques e entre outros); mudança de esquema cumulativo “M6” (o tempo total de mudanças de esquema/layout).

De seguida, são apresentados os resultados obtidos as amostras evidenciadas (seção 6.3.3) onde foram efetuados os testes de desempenho. Cada amostra (página criada) foi testada 10 vezes de forma a que os resultados fossem os mais precisos possíveis, o que faz 50 medições no total. Os valores a ser medidos não são sempre uniformes pois podem ser observados valores muito discrepantes. Estes valores podem aparecer devido a existência de interferência (Ex: picos de latência) e são conhecidos como *outliers*. Os resultados para cada amostra foram obtidos através da mediana das 10 medições e são apresentados na Tabela 25.

Tabela 25 – Resultados obtidos através da mediana das amostras utilizadas

Métricas	M1	M2	M3	M4	M5	M6
Amostra 1 - (80)	4.8	7.1	5.2	5.3	410	0
Amostra 2 - (40)	4.7	6.2	5.1	5.2	400	0
Amostra 3 - (20)	4.7	5.3	5.1	5.2	385	0
Amostra 4 - (20 apenas do tipo “Callback”)	4.7	5.1	5.5	5.6	540	0.001
Amostra 5 - (10)	4.7	4.4	5.1	5.3	390	0

6.4.7 Avaliação dos resultados dos testes de desempenho

Através dos resultados obtidos é possível observar e retirar diversas conclusões. Inicialmente é possível observar que certas métricas pouco se alteram perante o conteúdo presente na página. As métricas M1, M3, M4, M5, M6 apresentaram sempre valores semelhantes para todas as amostras. Como já foi referido, estas amostras apenas diferem na quantidade de componentes a serem apresentados (80, 40, 20, 20 e 10 componentes) portanto é normal que algumas métricas obtidas valores semelhantes em todas elas.

O facto de o número de tipos de componentes ter sido igualmente distribuído causa com que o primeiro e maior preenchimento (M1 e M3) seja sempre igual ou semelhante.

O carregamento de scripts JavaScript, css e outras bibliotecas é um dos diversos fatores que causa impacto no tempo com que a página fica bloqueada e interativa (M4, M6) e estes também são iguais para todas as páginas criadas. Os valores obtidos comprovam esta semelhança.

O tempo de mudança de esquema (*layout*) é nestas também semelhante e igual ao valor 0, pois a maior parte do conteúdo presente pode ser carregado de imediato através do URL. A única amostra que apresenta um valor diferente para esta métrica é a amostra 4. Esta apenas

contém componentes do tipo “Callback” onde é efetuado um carregamento inicial e a medida que se obtém o resultado dos pedidos externos a interface é atualizada. Isto irá provocar mudanças ao esquema o que fez com que o valor obtido fosse um pouco maior. A presença de duas amostras com o mesmo número de componentes em que uma delas continha componentes do tipo “Callback” serviu para analisar se este tipo de componentes afetava de alguma forma o desempenho. Pelos resultados já observados é possível concluir que apenas o tempo de mudança de esquema será afetado e de forma mínima.

Por fim, temos a métrica mais importante para avaliar esta solução e onde é possível observar uma maior discrepância, o índice de velocidade (M2). À primeira vista os diferentes tempos parecem ser afetados pelo número de componentes. De forma a que fosse possível perceber se existe relação entre esta métrica e número de componentes foi calculado o coeficiente de correlação. A fórmula mais utilizada e indicada para os dados desta solução é a de Pearson, que foi aplicada na Tabela 26 [113].

Tabela 26 – Cálculo coeficiente de correlação

Componentes	Índice Velocidade (M2)
80	7.1
40	6.2
20	5.3
20	5.1
10	4.4
Coeficiente de correlação (r)	0,96233833

O coeficiente obtido foi de 0,96, o que comprova uma forte relação entre as duas variáveis [113]. Isto significa que quanto maior for o número de componentes presentes na página maior será o tempo de carregamento da página. Um utilizador deverá ter esta informação em conta, pois pode optar por dar preferência a uma página mais rápida do que à quantidade de conteúdo.

Nesta solução era importante perceber se as páginas criadas não ultrapassavam o limite imposto de desempenho na definição dos requisitos não funcionais (seção 4.2). Este limite era de 10 segundos, que representa o tempo máximo de espera de um utilizador. Como é possível observar este limite não é atingido mesmo utilizando o número máximo de componentes, o que comprova o cumprimento deste requisito. Foi possível ainda comprovar que o número de componentes presentes na página irá afetar diretamente o tempo de carregamento. Os resultados obtidos reforçam ainda mais que para a 1ª Hipótese, a hipótese alternativa foi cumprida.

7 Conclusão

Neste capítulo são apresentadas as conclusões relativamente à dissertação e solução preconizada. São aqui analisados os objetivos propostos a atingir, objetivos esses delineados no início desta dissertação. Serão também expostas algumas das limitações e dificuldades encontradas na escrita, pesquisa e desenvolvimento da solução. De acordo com os pontos anteriores, são apresentadas algumas melhorias e trabalho futuro. Por fim, é efetuada uma apreciação final e pessoal sobre toda a dissertação e o processo envolvente.

7.1 Objetivos alcançados

O espaço ou vazio existente no mercado para o tipo de solução aqui estudado, era o problema que esta dissertação pretendia ajudar a solucionar. O objetivo desta solução passava por conciliar e combinar a criação de *Landing Pages* com opções de personalização, com a rapidez e facilidade de acesso a informação disponível na página através de códigos QR. Os códigos QR criados deveriam permitir também a sua personalização, a par das páginas criadas. Ou seja, criar uma solução completa com estes dois elementos requereu que fossem delineados pequenos objetivos de forma a atingir a totalidade desejada.

Um dos objetivos era a realização de uma solução web que permitisse a criação das *Landing Pages*, possibilitando que clientes deste software possam oferecer aos seus clientes informação sobre os seus produtos, serviços ou qualquer outra informação que queiram disponibilizar. Nesta medida, a solução concebida cumpre com o que foi delineado inicialmente, pois permite ao utilizador fazer configurações gerais contidas na maioria das páginas Web como, por exemplo, barra de navegação e rodapé. Permite também adição de componentes ao corpo da página. Aqui o utilizador pode optar por adicionar qualquer tipo de componentes disponibilizados (Ex: input, texto, imagem, vídeos ou datas). Inclusive permite a adição de um componente que aceita diversos tipos de URL, de forma a que possa ser

efetuado um pedido externo pelo conteúdo a disponibilizar. O conteúdo presente no URL do pedido poderá ser alterado no futuro e, se for um dos vários formatos aceites pela solução, será corretamente exposto. Desta forma, o conteúdo inserido pode ser alterado no futuro, oferecendo dinamismo à solução. É importante referir que cada componente tem opções de personalização e de permissão diferentes (público/privado). Podemos concluir que este objetivo foi cumprido na sua íntegra.

Outro dos objetivos seria o acesso às páginas criadas de forma automatizada e facilitada, com recurso aos códigos QR. Após a criação das páginas o utilizador é sempre redirecionado para uma página de criação de códigos. São oferecidas diversas opções ao utilizador, algumas a par das já existentes e encontradas na pesquisa científica, outras não existente nas plataformas e pesquisa evidenciada. O utilizador poderá escolher entre os diversos níveis de correção existentes, inverter ambas as cores do código ou até mesmo inserir uma imagem no centro do código. No fim é disponibilizado o ficheiro com o código QR em diversos formatos para que o utilizador possa dar o uso que desejar. Os códigos criados irão conter muita da informação referente à criação anterior da página. Por fim, podemos concluir que este objetivo é alcançado pois o utilizador final poderá aceder à *Landing Page* criada através de uma simples digitalização do código QR gerado.

A necessidade de existir um espaço dedicado à gestão das páginas criadas e códigos criados era também um dos objetivos a ser alcançado. Nesta página, o utilizador pode fazer alterações gerais às permissões da página, como adição de novos utilizadores com capacidades de ver os campos privados existentes ou oferecer direitos de administração da página. O utilizador consegue ainda criar novas páginas a partir de uma existente, no sentido em que pode modificar o valor principal dos campos criados inicialmente e gerar um novo código QR. Ou seja, o utilizador utiliza o mesmo modelo criado para uma página anterior. Esta seção encontra-se um pouco limitada e existem algumas melhorias futuras a ser produzidas que são analisadas na próxima secção. No entanto, na sua essência o objetivo foi cumprido.

Por fim, um dos grandes objetivos era que qualquer pessoa sem conhecimentos técnicos e informáticos usasse com sucesso e facilidade esta solução. Este objetivo era importante pois permite diferenciar a solução preconizada de algumas soluções semelhantes já existentes e porque permite incluir um público diferente do habitual deste tipo de solução. Para que isso fosse possível, optou-se por utilizar na solução uma metodologia diferente das estudadas, como descrita na secção 4.6.1.1. Foi optado por proporcionar a personalização de cada componente em linguagem perceptível por qualquer pessoa. Por exemplo, se uma pessoa optar por mudar o tamanho de um botão, nas opções do mesmo ira ter entre as escolhas palavras como pequeno, médio ou grande. Estas palavras são depois, através da base de dados, transformadas no HTML correspondente (Ex: pequeno = btn-small). Através dos inquéritos realizados à solução consegue-se perceber que o sistema em si obteve no geral uma boa pontuação. Mas podemos observar também por heurísticas, principalmente a capacidade de aprendizagem, que apesar de na sua média apresentar um bom valor, apresenta respostas muito dispersas e pouco uniformes em concreto na questão 4 (Figura 51). Esta questão aborda a necessidade de um técnico no uso da solução, verificando-se a

existência de resultados contraditórios. Perante o objetivo exposto, o resultado a esta questão deveria ser mais uniforme para comprovar a facilidade uniforme do uso da solução, o que leva a concluir que o mesmo apenas se encontra parcialmente cumprido existindo a necessidade de melhoria.

7.2 Limitações encontradas e trabalho futuro

Após a conclusão da solução, como anteriormente evidenciado, foi possível observar se os objetivos da solução foram ou não cumpridos na sua plenitude. No entanto, é possível também descrever algumas das limitações encontradas durante o desenvolvimento e possível trabalho futuro de melhoria.

Uma das limitações encontrada ao conceber a solução foi a própria Framework utilizada, o Blazor. Apesar de ter sido a escolha inicial, através de pesquisa e do método de apoio à decisão, a escolha tomada acabou por não ser a melhor. Após já se estar no estado um pouco avançado de desenvolvimento, começou-se a perceber que a mesma não se encontrava madura suficiente para conseguir atingir certas funcionalidades. Devido ao dinamismo necessário para esta implementação, por vezes foi necessário utilizar métodos muito extensos para atingir um objetivo, que noutras *Frameworks* era proporcionado de raiz ou através de um processo mais fácil. O facto de existirem ainda poucas bibliotecas externas de apoio ao desenvolvimento nesta framework também dificultou o processo.

Para o futuro, e com o auxílio dos inquéritos realizados, foram já evidenciadas algumas mudanças necessárias, caso o desenvolvimento da solução seja continuado. Em primeiro, um dos pontos mais importantes será tornar a solução ainda mais fácil de utilizar, pois como já foi descrito, não existiu uniformidade nos resultados de algumas questões e as sugestões dadas pelos inqueridos na questão aberta demonstram pontos que devem ser tidos em conta e aplicados. Ou seja, cada componente adicionado ao corpo da página deverá apresentar uma seção ou ícone de informação, para o caso de as pessoas necessitarem de obter mais informações sobre aquilo que estão a fazer e de como o fazer. Em paralelo poderá ser criada uma página que reúna toda a informação dos componentes presentes e de como os utilizar.

São possíveis ainda diversas melhorias a ser implementadas relativamente as *Landing Pages*. Deverá ser possível criar um novo código para a mesma página, após aceder à mesma através de uma digitalização e se tiver permissões de utilizador. Como os valores principais dos componentes ficam guardados no URL inserido no código, a criação de um novo código QR para a mesma página poderá ocorrer apenas por aqui. Desta forma, é possível criar um código igual ao atual ou novos códigos com novas características (Ex: cores diferentes) para a mesma página.

Por fim, deverá ser possível modificar o modelo de uma página atual, sem deixar que códigos QR que tenham sido anteriormente criados deixem de funcionar completamente. O utilizador deverá conseguir adicionar ou remover componentes a uma página criada. Se um URL de

acesso a página tiver sido criado antes das alterações efetuadas e conter referência a algum elemento que já não exista, o componente simplesmente não irá ser apresentado. O mesmo acontece caso exista configurações na base de dados de um componente, mas o URL não contenha informações sobre o mesmo.

Estas limitações e sugestões culminam do trabalho científico e técnico realizado que foi aqui posteriormente avaliado e apresentado. As diversas sugestões descritas podem ser assim implementadas no futuro se a empresa requerente assim o desejar e achar necessário.

7.3 Apreciação final e pessoal

Com o finalizar desta dissertação consigo concluir que a maioria dos objetivos propostos pela empresa requerente foram atingidos e que a solução concebida irá ter o uso idealizado. Em termos científicos, esta dissertação permitiu-me perceber melhor alguns conceitos que até agora desconhecia, bem como algumas abordagens para investigar e recolher informação.

A maior vantagem pessoal adquirida, terá sido o conhecimento e experiência técnica. O facto de a dissertação ter sido efetuada em contexto de estágio fez com que fosse possível adquirir mais experiência no mercado de trabalho. Aqui consegui adquirir novos conhecimentos de programação bem como novas metodologias e técnicas de trabalho.

Importante referir que, no decorrer desta dissertação, foram aplicados a maior parte dos módulos lecionados no âmbito da unidade curricular que a dissertação se encontra (TMDEI). Nesta dissertação foram aplicados módulos como a escrita técnico-científica, análise de valor e Experimentação e Avaliação. A documentação e contexto adquiridos nos módulos lecionados, foi fulcral para melhorar o conteúdo e qualidade.

Por fim, posso concluir que esta dissertação concretizou o seu propósito com sucesso e proporcionou uma experiência positiva sem igual que me irá ajudar no futuro quer a nível pessoal como profissional, apesar de todas as adversidades encontras pelo caminho.

Referências

- [1] Heiner Lasi. Hans Georg Kemper. Peter Feltke. Thomas Feld. Michael Hoffmann, "Industry 4.0 in Business & Information System Engineering," *Business & Inf. Syst. Eng.*, vol. 6, no. 4, pp. 239–242, 2014.
- [2] M. Hermann, T. Pentek, B. Otto, and T. * Pentek, "Design Principles for Industrie 4.0 Scenarios: A Literature Review Competence Center Corporate Data Quality (CC CDQ) View project Industrial Data Space View project Design Principles for Industrie 4.0 Scenarios: A Literature Review," no. September 2016, p. 16, 2015.
- [3] I. I. Consortium, "Industry 4.0 Definition Industrial internet Factsheet," no. September 2015, p. 2015, 2019.
- [4] H. Kagermann, W. Wahlster, and J. Helbig, "Securing the future of German manufacturing industry: Recommendations for implementing the strategic initiative INDUSTRIE 4.0," *Final Rep. Ind. 4.0 Work. Gr.*, no. April, pp. 1–84, 2013.
- [5] H. Lasi, P. Fettke, H. G. Kemper, T. Feld, and M. Hoffmann, "Industry 4.0," *Bus. Inf. Syst. Eng.*, vol. 6, no. 4, pp. 239–242, Aug. 2014.
- [6] Y. Lu, "Industry 4.0: A survey on technologies, applications and open research issues," *Journal of Industrial Information Integration*, vol. 6. Elsevier B.V., pp. 1–10, 01-Jun-2017.
- [7] D. Gorecky, M. Schmitt, M. Loskyll, and D. Zühlke, "Human-machine-interaction in the industry 4.0 era," in *Proceedings - 2014 12th IEEE International Conference on Industrial Informatics, INDIN 2014*, 2014, pp. 289–294.
- [8] A. Lele, "Industry 4.0," in *Smart Innovation, Systems and Technologies*, vol. 132, Springer Science and Business Media Deutschland GmbH, 2019, pp. 205–215.
- [9] "What Is a Landing Page? All Your Questions, Answered." [Online]. Available: <https://blog.hubspot.com/blog/tabid/6307/bid/7177/what-is-a-landing-page-and-why-should-you-care.aspx>. [Accessed: 24-Mar-2020].
- [10] "O que é uma landing page? – DesignPT." [Online]. Available: <https://designportugal.net/o-que-e-uma-landing-page/>. [Accessed: 24-Mar-2020].
- [11] "Quanto custa fazer um site." [Online]. Available: <https://www.e-konomista.pt/quanto-custa-fazer-um-site/>. [Accessed: 13-Apr-2020].
- [12] "Definitive Guide - How Much Should A Website Cost You? (Apr 20)." [Online]. Available: <https://www.websitebuilderexpert.com/building-websites/how-much-should-a-website-cost/>. [Accessed: 13-Apr-2020].
- [13] "What Is Markdown and How Do You Use It?" [Online]. Available: <https://www.howtogeek.com/448323/what-is-markdown-and-how-do-you-use-it/>. [Accessed: 04-May-2020].
- [14] "Getting Started | Markdown Guide." [Online]. Available: <https://www.markdownguide.org/getting-started/>. [Accessed: 04-May-2020].
- [15] "What is Markdown, and why should you use it? (4 reasons)." [Online]. Available: <https://www.ultraedit.com/company/blog/community/what-is-markdown-why-use-it.html>. [Accessed: 04-May-2020].
- [16] "Arrastar e soltar - DragDrop | MDN." [Online]. Available: https://developer.mozilla.org/pt-BR/docs/DragDrop/Drag_and_Drop. [Accessed: 07-May-2020].
- [17] "HTML Drag and Drop API." [Online]. Available: https://www.w3schools.com/html/html5_draganddrop.asp. [Accessed: 07-May-2020].
- [18] "Carrd - Simple, free, fully responsive one-page sites for pretty much anything." [Online]. Available: <https://carrd.co/>. [Accessed: 27-Mar-2020].
- [19] "Choose a Starting Point - Carrd." [Online]. Available: <https://carrd.co/build>. [Accessed: 27-Mar-2020].
- [20] "Construtor de Sites Wix | Crie Seu Próprio Site Incrível." [Online]. Available: <https://pt.wix.com/new/intro/split?ref=acquisition>. [Accessed: 10-Apr-2020].

- [21] “Templates Grátis | Modelos de Sites em HTML5 | Wix.com.” [Online]. Available: <https://pt.wix.com/website/templates>. [Accessed: 10-Apr-2020].
- [22] “Wix App Market | Apps Incríveis para o Seu Site | Wix.com.” [Online]. Available: <https://pt.wix.com/app-market/main>. [Accessed: 10-Apr-2020].
- [23] “Painel de controle | Wix.com.” [Online]. Available: <https://www.wix.com/dashboard/2e2576fc-bc00-4344-b92c-b2a9b1a0f319/home>. [Accessed: 10-Apr-2020].
- [24] “Criar uma landing page.” [Online]. Available: <https://mailchimp.com/pt/help/create-a-landing-page/>. [Accessed: 25-Mar-2020].
- [25] “Tipos de modelos de landing pages.” [Online]. Available: <https://mailchimp.com/pt/help/types-landing-page-templates/>. [Accessed: 25-Mar-2020].
- [26] “Usar blocos de conteúdo de formulário de inscrição.” [Online]. Available: <https://mailchimp.com/pt/help/use-signup-form-blocks/>. [Accessed: 25-Mar-2020].
- [27] “Use Product Content Blocks.” [Online]. Available: <https://mailchimp.com/en/help/use-product-content-blocks/>. [Accessed: 25-Mar-2020].
- [28] “History of QR Code | QRcode.com | DENSO WAVE.” [Online]. Available: <https://www.qrcode.com/en/history/>. [Accessed: 12-Nov-2019].
- [29] P. Kieseberg *et al.*, “QR code security,” *MoMM2010 - 8th Int. Conf. Adv. Mob. Comput. Multimed.*, pp. 430–435, 2010.
- [30] K. C. Liao and W. H. Lee, “A novel user authentication scheme based on QR-code,” *J. Networks*, vol. 5, no. 8, pp. 937–941, 2010.
- [31] “What is a QR Code? | QRcode.com | DENSO WAVE.” [Online]. Available: <https://www.qrcode.com/en/about/>. [Accessed: 18-Nov-2019].
- [32] “Error correction feature | QRcode.com | DENSO WAVE.” [Online]. Available: https://www.qrcode.com/en/about/error_correction.html. [Accessed: 10-Dec-2019].
- [33] P. Suthesbanjard, “QR-Code Generator,” pp. 89–92, 2010.
- [34] A. Mishra and M. Mathuria, “A Review on QR Code,” *Int. J. Comput. Appl.*, vol. 164, no. 9, pp. 17–19, 2017.
- [35] S. Vongpradhip and S. Rungrangsilp, “QR code using invisible watermarking in frequency domain,” *Int. Conf. ICT Knowl. Eng.*, pp. 47–52, 2011.
- [36] T. Kan, C. Teng, and M. Y. Chen, “Handbook of Augmented Reality,” *Handb. Augment. Real.*, 2011.
- [37] T. J. Soon, “QR Code,” *Synthesis (Stuttg.)*, pp. 59–78, 2008.
- [38] A. A. Wara, “Enhancing User Experience using Mobile QR-Code Application,” vol. 03, no. 06, pp. 1310–1315, 2014.
- [39] “QR Code Barcode Size Settings - OnBarcode.com.” [Online]. Available: http://www.onbarcode.com/qr_code/qr_code_size_setting.html. [Accessed: 03-Jan-2020].
- [40] S. Han *et al.*, “Lithographically encoded polymer microtaggant using high-capacity and error-correctable QR Code for anti-counterfeiting of drugs,” *Adv. Mater.*, vol. 24, no. 44, pp. 5924–5929, 2012.
- [41] D. Samretwit and T. Wakahara, “Measurement of reading characteristics of multiplexed image in QR code,” *Proc. - 3rd IEEE Int. Conf. Intell. Netw. Collab. Syst. INCoS 2011*, pp. 552–557, 2011.
- [42] G. Starnberger, L. Frohofer, and K. M. Goeschka, “QR-TAN: Secure mobile transaction authentication,” *Proc. - Int. Conf. Availability, Reliab. Secur. ARES 2009*, pp. 578–583, 2009.
- [43] “Information capacity and versions of QR Code | QRcode.com | DENSO WAVE.” [Online]. Available: <https://www.qrcode.com/en/about/version.html>. [Accessed: 12-Dec-2019].
- [44] S. Tiwari, “An introduction to QR code technology,” *Proc. - 2016 15th Int. Conf. Inf. Technol. ICIT 2016*, vol. 1, pp. 39–44, 2017.
- [45] F. Parra, “COMUNICAÇÃO CONTEMPORÂNEA, CULTURA DIGITAL E PRÁTICAS SOCIOCULTURAIS: RELAÇÕES ENTRE USUÁRIO-INTERATOR E TECNOLOGIA QR CODE PRÓ-REITORIA ACADÊMICA PROGRAMA DE PÓS-GRADUAÇÃO EM COMUNICAÇÃO E CULTURA Felipe Parra Alves de Oliveira SOCIOCULTURAIS: RELAÇÃO,” no. November, 2016.
- [46] “iQR Code | QRcode.com | DENSO WAVE.” [Online]. Available: <https://www.qrcode.com/en/codes/iqr.html>. [Accessed: 20-Dec-2019].

- [47] "SQRC® | QR Code Solutions | System Solution | DENSO WAVE." [Online]. Available: <https://www.denso-wave.com/en/system/qr/product/sqrc.html>. [Accessed: 20-Dec-2019].
- [48] H. K. Chu, C. S. Chang, R. R. Lee, and N. J. Mitra, "Halftone QR codes," *ACM Trans. Graph.*, vol. 32, no. 6, pp. 1–8, 2013.
- [49] "Halftone QR Codes." [Online]. Available: http://vecg.cs.ucl.ac.uk/Projects/SmartGeometry/halftone_QR/halftoneQR_sigga13.html. [Accessed: 06-Jan-2020].
- [50] "QR Code Model 1 and Model 2 | QRcode.com | DENSO WAVE." [Online]. Available: <https://www.qrcode.com/en/codes/model12.html>. [Accessed: 17-Dec-2019].
- [51] J. H. Chang, "An introduction to using QR codes in scholarly journals," *Sci. Ed.*, vol. 1, no. 2, pp. 113–117, 2014.
- [52] K. Saranya, R. S. Reminaa, and S. Subhitsha, "Modern applications of QR-Code for security," *Proc. 2nd IEEE Int. Conf. Eng. Technol. ICETECH 2016*, no. March, pp. 173–177, 2016.
- [53] M. F. Tretinjak, "The implementation of QR codes in the educational process," *2015 38th Int. Conv. Inf. Commun. Technol. Electron. Microelectron. MIPRO 2015 - Proc.*, no. May, pp. 833–835, 2015.
- [54] A. Hłobaz, K. Podlaski, and P. Milczarski, "Applications of QR Codes in Secure Mobile Data Exchange," *Commun. Comput. Inf. Sci.*, vol. 431, pp. 277–286, 2014.
- [55] "Micro QR Code | QRcode.com | DENSO WAVE." [Online]. Available: <https://www.qrcode.com/en/codes/microqr.html>. [Accessed: 17-Dec-2019].
- [56] "Frame QR | QRcode.com | DENSO WAVE." [Online]. Available: <https://www.qrcode.com/en/codes/frameqr.html>. [Accessed: 20-Dec-2019].
- [57] "Frame QR® | QR Code Solutions | System Solution | DENSO WAVE." [Online]. Available: <https://www.denso-wave.com/en/system/qr/product/frame.html>. [Accessed: 01-Feb-2020].
- [58] "Many ways of using QR Codes | QRcode.com | DENSO WAVE." [Online]. Available: <https://www.qrcode.com/en/about/howtouse.html>. [Accessed: 01-Feb-2020].
- [59] F. Ales, "QR Codes on Packaging : A Technology Acceptance Model approach comparing informative and entertaining content .," 2019.
- [60] "Amazon's SmileCodes are QR codes users scan to get discounts and other offers - The Verge." [Online]. Available: <https://www.theverge.com/2018/1/31/16954336/amazons-smilecodes-qr-codes-discounts>. [Accessed: 03-Feb-2020].
- [61] "Amazon SmileCodes: The New 2D Barcodes To Avail Amazon Benefits." [Online]. Available: <https://scanova.io/blog/amazon-smilecodes/>. [Accessed: 03-Feb-2020].
- [62] M. C. Marreiros and C. Martins, "ESCOLA SUPERIOR DE HOTELARIA E TURISMO DO ESTORIL MESTRADO EM TURISMO ESPECIALIZAÇÃO EM GESTÃO ESTRATÉGICA DE EVENTOS INFLUÊNCIA DOS PATROCÍNIOS DE EVENTOS NA INTENÇÃO DE COMPRA: O Caso da Super Bock nos Festivais de Música," 2013.
- [63] A. S. B. T. de Sousa, "Uso Do Qr Code No Marketing Digital: a Perspetiva Do Utilizador Português Instituto Superior De Contabilidade E Administração Do Porto Instituto Politécnico Do Porto," 2014.
- [64] "QR Code Generator." [Online]. Available: <https://www.the-qrcode-generator.com/>. [Accessed: 03-Jan-2020].
- [65] "QR Code Generator | Create Your Free QR Codes." [Online]. Available: <https://www.qr-code-generator.com/>. [Accessed: 03-Jan-2020].
- [66] "QRCode Monkey - The free QR Code Generator to create custom QR Codes with Logo." [Online]. Available: <https://www.qrcode-monkey.com/>. [Accessed: 03-Jan-2020].
- [67] "Visual QR Code Generator - Visualead." [Online]. Available: <https://www.visualead.com/>. [Accessed: 03-Jan-2020].
- [68] "Blazor | Build client web apps with C# | .NET." [Online]. Available: <https://dotnet.microsoft.com/apps/aspnet/web-apps/blazor>. [Accessed: 27-Dec-2019].
- [69] B. Joshi, "Blazor," in *Beginning Database Programming Using ASP.NET Core 3*, Berkeley, CA: Apress, 2019, pp. 337–380.
- [70] "Introduction to ASP.NET Core Blazor | Microsoft Docs." [Online]. Available: <https://docs.microsoft.com/en-us/aspnet/core/blazor/?view=aspnetcore-3.1>. [Accessed: 27-Dec-2019].

- [71] "Get started with ASP.NET Core Blazor | Microsoft Docs." [Online]. Available: <https://docs.microsoft.com/en-us/aspnet/core/blazor/get-started?view=aspnetcore-3.1&tabs=visual-studio>. [Accessed: 27-Dec-2019].
- [72] "React – A JavaScript library for building user interfaces." [Online]. Available: <https://reactjs.org/>. [Accessed: 27-Dec-2019].
- [73] "ReactJS - Overview - Tutorialspoint." [Online]. Available: https://www.tutorialspoint.com/reactjs/reactjs_overview.htm. [Accessed: 27-Dec-2019].
- [74] "Angular - Architecture overview." [Online]. Available: <https://angular.io/guide/architecture>. [Accessed: 03-Jan-2020].
- [75] "Angular 8 Architecture - Javatpoint." [Online]. Available: <https://www.javatpoint.com/angular-8-architecture>. [Accessed: 03-Jan-2020].
- [76] "Fun with Angular: A Quick Overview." [Online]. Available: <https://www.telerik.com/blogs/fun-with-angular-a-quick-overview>. [Accessed: 03-Jan-2020].
- [77] L. D. Miles, "Techniques Of Value Analysis And Engineering," *6th Annu. Intl. Emp. Qual. Control Conf.*, Jan. 1962.
- [78] P. A. Koen *et al.*, "Fuzzy front end: effective methods, tools, and techniques," *PDMA Toolb. 1 new Prod. Dev.*, 2002.
- [79] S. Nicola, "Análise de Valor." .
- [80] P. Koen *et al.*, "Providing clarity and a common language to the 'fuzzy front end,'" *Res. Manag.*, vol. 44, no. 2, pp. 46–55, 2001.
- [81] P. A. Koen, H. M. J. Bertels, and E. Kleinschmidt, "Managing the front end of innovation-part I: Results from a three-year study," *Res. Technol. Manag.*, vol. 57, no. 2, pp. 34–43, 2014.
- [82] P. Teza, V. Miguez, R. Fernandes, and G. Dandolini, "Direcionadores do processo de inovação: o papel da estratégia, liderança e cultura," *Navus Rev. Gestão e Tecnol.*, vol. 3, Nov. 2013.
- [83] P. A. Koen, H. M. J. Bertels, and E. J. Kleinschmidt, "Managing the front end of innovation-part II: Results from a three-year study," *Res. Technol. Manag.*, vol. 57, no. 3, pp. 25–35, 2014.
- [84] A. Osterwalder and Y. Pigneur, "Modeling value propositions in e-business," *ACM Int. Conf. Proceeding Ser.*, vol. 50, pp. 429–436, 2003.
- [85] A. Osterwalder, Y. Pigneur, G. Bernarda, A. Smith, and T. Papadakos, "Value Proposition Design."
- [86] J. Magretta, "Why Business Models Matter," 2002.
- [87] A. Joyce and R. L. Paquin, "The triple layered business model canvas: A tool to design more sustainable business models," *J. Clean. Prod.*, vol. 135, pp. 1474–1486, Nov. 2016.
- [88] T. L. Saaty, "What is the Analytic Hierarchy Process?," in *Mathematical Models for Decision Support*, Springer Berlin Heidelberg, 1988, pp. 109–121.
- [89] S. Nicola, "Análise_valor_Aula_2_22OUT_2019_1hora_AHP." 2019.
- [90] L. K. Chan and M. L. Wu, "Quality function deployment: A literature review," *Eur. J. Oper. Res.*, vol. 143, no. 3, pp. 463–497, Dec. 2002.
- [91] L. P. Sullivan, "Quality Function Deployment.," *Quality Progress*, vol. 19, no. 6. pp. 39–50, 1986.
- [92] "Modelo de Domínio: Visualizando Conceitos."
- [93] "Use Case Diagrams - Use Case Diagrams Online, Examples, and Tools." [Online]. Available: <https://www.smartdraw.com/use-case-diagram/>. [Accessed: 09-Feb-2020].
- [94] "ASP.NET Core Blazor hosting models | Microsoft Docs." [Online]. Available: <https://docs.microsoft.com/en-us/aspnet/core/blazor/hosting-models?view=aspnetcore-3.1>. [Accessed: 18-Jun-2020].
- [95] "Git." [Online]. Available: <https://git-scm.com/>. [Accessed: 22-May-2020].
- [96] "Bitbucket | The Git solution for professional teams." [Online]. Available: <https://bitbucket.org/>. [Accessed: 22-May-2020].
- [97] "Sourcetree | Free Git GUI for Mac and Windows." [Online]. Available: <https://www.sourcetreeapp.com/>. [Accessed: 22-May-2020].
- [98] "What Is A Non Relational Database | MongoDB." [Online]. Available: <https://www.mongodb.com/scale/what-is-a-non-relational-database>. [Accessed: 27-May-2020].
- [99] "alexandre-spieser/AspNetCore.Identity.MongoDb." [Online]. Available: <https://github.com/alexandre-spieser/AspNetCore.Identity.MongoDbCore>. [Accessed: 26-May-2020].

- [100] “tugberkugurlu/AspNetCore.Identity.MongoDB.” [Online]. Available: <https://github.com/tugberkugurlu/AspNetCore.Identity.MongoDB>. [Accessed: 26-May-2020].
- [101] “matteofabbri/AspNetCore.Identity.Mongo: This is a MongoDB provider for the ASP.NET Core 2 Identity framework.” [Online]. Available: <https://github.com/matteofabbri/AspNetCore.Identity.Mongo>. [Accessed: 25-May-2020].
- [102] “Razor Pages Files | Learn Razor Pages.” [Online]. Available: <https://www.learnrazorpages.com/razor-pages/>. [Accessed: 21-Apr-2020].
- [103] A. Bangor, P. T. Kortum, and J. T. Miller, “An empirical evaluation of the system usability scale,” *Int. J. Hum. Comput. Interact.*, vol. 24, no. 6, pp. 574–594, Aug. 2008.
- [104] K. Finstad, “The System Usability Scale and Non-Native English Speakers,” 2006.
- [105] J. R. Lewis and J. Sauro, “The factor structure of the system usability scale,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2009, vol. 5619 LNCS, pp. 94–103.
- [106] “Usability 101: Introduction to Usability.” [Online]. Available: <https://www.nngroup.com/articles/usability-101-introduction-to-usability/>. [Accessed: 04-Jun-2020].
- [107] “Medindo a usabilidade do seu produto com System Usability Scale (SUS).” [Online]. Available: <https://medium.com/design-contaazul/medindo-a-usabilidade-do-seu-produto-com-system-usability-scale-sus-3956612d9229>. [Accessed: 04-Jun-2020].
- [108] J. Brooke, “SUS-A quick and dirty usability scale,” *Usability Eval. Ind.*, vol. 189, no. 194, pp. 4–7, 1996.
- [109] J. Brooke, “SUS: A Retrospective,” 2013.
- [110] “Page Speed — 2020 Website Best Practices - Moz.” [Online]. Available: <https://moz.com/learn/seo/page-speed>. [Accessed: 22-Jun-2020].
- [111] “Web performance testing: 18 free open-source tools to consider | TechBeacon.” [Online]. Available: <https://techbeacon.com/app-dev-testing/web-performance-testing-18-free-open-source-tools-consider>. [Accessed: 22-Jun-2020].
- [112] “PageSpeed Insights.” [Online]. Available: <https://developers.google.com/speed/pagespeed/insights/>. [Accessed: 22-Jun-2020].
- [113] “Correlation Coefficient: Simple Definition, Formula, Easy Calculation Steps.” [Online]. Available: <https://www.statisticshowto.com/probability-and-statistics/correlation-coefficient-formula/>. [Accessed: 23-Jun-2020].