



IPG Politécnico
| da Guarda
Escola Superior
de Tecnologia e Gestão

RELATÓRIO DE ESTÁGIO

Curso Técnico Superior Profissional
em Desenvolvimento de Aplicações Informáticas

Rui Pedro Santos Costa

outubro | 2021



Relatório de Estágio

Rui Pedro Santos Costa

IPG

CTeSP em Desenvolvimento de Aplicações Informática



UTC DE INFORMÁTICA
Escola Superior de Tecnologia e Gestão

Outubro, 2021

CTeSP em Desenvolvimento de Aplicações Informática.

Candidato: Rui Pedro Santos Costa, Nº 1703532, 1703532@sal.ipg.pt

Orientação Científica: Paulo Alexandre de Andrade Vieira, pavieira@ipg.pt

Empresa: MestreClique - Sistemas de Informação, Lda.

Orientador na Empresa: Rui Pereira, rui@mestreclique.pt



UTC DE INFORMÁTICA

Escola Superior de Tecnologia e Gestão

Endereço: Av. Dr. Francisco Sá Carneiro 50, 6300-559 Guarda

Outubro, 2021

Elementos Identificativos

Aluno

Nome: Rui Pedro Santos Costa

Número de aluno: 1703532

Curso: CTeSP Desenvolvimento de Aplicações Informáticas

Ano Letivo: 2020/2021

E-mail: 1703532@sal.ipg.pt

Estabelecimento de Ensino

Estabelecimento de ensino: Escola Superior de Tecnologia e Gestão - Instituto Politécnico da Guarda

Morada: Av. Dr. Francisco Sá Carneiro, 50 – 6300-559 Guarda

Telefone: +351 271 220 100 E-mail: ipg@ipg.pt

Orientador: Professor Doutor Paulo Alexandre de Andrade Vieira

E-mail: paulovieira@ipg.pt

Local de Estágio

Empresa de acolhimento: MestreClique - Sistemas de informação Lda

Morada: Av. Monsenhor Mendes do Carmo, N.º12 A e B, 6300-586 Guarda

Telefone: 271 227 354 | 927 407 402

E-mail: geral@mestreclique.pt

Website: mestreclique.pt

Supervisor na empresa de acolhimento: Engenheiro Rui Pereira

Período do Estágio: 02 de Março a 8 de Julho

Duração do Estágio: 750 horas

Agradecimentos

Gostaria de agradecer de modo geral, a todos aqueles que direta ou indiretamente tiveram influência positiva durante o Curso CTesp Desenvolvimento de Aplicações Informáticas dos anos letivos 2019/2020 e 2020/2021.

Agradeço ao Professor Paulo Vieira pela disponibilidade que mostrou e toda a ajuda que proporcionou ao longo deste percurso, não só como orientador, mas também como professor. Aproveito por deixar também aqui um agradecimento sincero a todos os professores que de uma maneira ou de outra contribuíram para me tornar um profissional e ser humano melhor.

Agradeço do fundo do coração a todos os colaboradores da empresa Mestre-Clique, ao Engenheiro Rui Pereira e ao Engenheiro Ricardo Dias por me disponibilizarem este estágio, e por me guiarem sempre da melhor maneira possível. Não destacando nenhuma colaborador em específico, agradeço de igual forma aos colaboradores António Carvalho, Cristina Martins, Fátima Andrade, Leonardo Lourenço, toda a aprendizagem e motivação que me proporcionaram ao longo desta experiência maravilhosa, posso com toda a certeza dizer que ganhei nesta empresa amigos para a vida.

Resumo

O estágio na empresa MestreClique, não só me proporcionou aprendizagens a níveis técnicos como programador e informático, mas também como ser humano, atendimento ao público, e outros fatores que fazem de mim hoje um profissional e técnico melhor. Ao longo deste percurso tive oportunidade de trabalhar no mundo das API's, desenvolvendo integrações com empresas como a Moloni, a E-goi, a Easy-Pay, a Microsoft, a Mailchimp, a API Recaptcha da Google, entre outras, usando ferramentas como o VB, ASP.net e SQL.

Ainda houve tempo para ajudar os outros colaboradores da MestreClique, que me proporcionaram aprendizagens no ramo da gestão documental, apoio ao cliente, reparações de hardware e software, e ainda instalações da plataforma MestreClique nas máquinas dos clientes.

Com o tempo que sobrou colaborei ainda na reestruturação dos sites que a MestreClique proporciona aos seus clientes, usando o Bootstrap para esse efeito, melhorámos a responsivo e a rapidez com que o site responde aos pedidos dos clientes, assim como o torná-mos mais bonito e moderno.

Palavras-Chave: MestreClique, VB, Asp.net, SQL, API's.

Abstract

The internship at MestreClique, not only gave me the opportunity to learn at a technical level as a programmer, but also as a human being, customer services, and other factors that make me today a much better professional and technical. During this I had the opportunity to work with API's, developing integrations with companies such as Moloni, E-goi, EasyPay, Microsoft, Mailchimp, the API Recaptcha from Google, and many others, using tools such as VB, ASP.net and SQL.

There was still time to help the MestreClique contributors, which gave me so much knowledge in document management area, customer services, hardware and software, and MestreClique platform installations on customer's computers.

With the remaining time I collaborated on the reconstruction of the MestreClique Websites that are sold to the customers, using Bootstrap to do it, we made the Website faster and more enjoyable for the customers.

Keywords: MestreClique, VB, Asp.net, SQL, API's.

Índice

Lista de Figuras	v
Lista de Símbolos	vii
1 Introdução	1
1.1 Objetivos	1
1.2 Calendarização	1
1.3 Organização do Relatório	1
2 MestreClique Lda	3
2.1 A Empresa	3
2.2 Colaboradores da empresa	3
2.3 Plataforma MestreClique	4
2.4 Caracterização do espaço e ambiente	5
3 Tecnologia utilizada	6
3.1 Visual Studio 2019:	6
3.2 Visual Basic:	6
3.3 Asp.Net:	7
3.4 SQL Manager:	7
3.5 GitHub:	8
3.6 Postman:	8
3.7 Bootstrap:	8
3.8 Recaptcha:	9
3.9 Ferramentas Microsoft:	9
3.10 Overleaf:	9
4 Tarefas realizadas	11
4.1 Descrição das principais tarefas desenvolvidas durante o estágio: . . .	11
4.1.1 Integração com a API Moloni:	11
4.1.2 Integração com a API Mailchimp:	12
4.1.3 Integração com a API E-goi:	12
4.1.4 Integração do PackWeb com a API Recaptcha do Google: . . .	12

4.1.5	Atualização para a mais recente versão da API EasyPay da plataforma MestreClique:	13
4.1.6	Integração da Framework Bootstrap no PackWeb MestreClique:	13
4.1.7	Desenvolvimento de website em VB/asp.net:	13
4.1.8	Criação de apresentações PowerPoint apartir de um tema (visual basic):	14
4.1.9	Atendimento ao cliente e acesso remoto para instalação e manutenção da plataforma MestreClique:	14
4.1.10	Gestão documental:	15
5	Desenvolvimento	16
5.1	Api Moloni	16
5.2	Api E-goí	21
5.3	Api EasyPay	24
5.4	Api Recaptcha	28
5.5	Bootstrap	29
5.6	Apresentações em Powerpoint	33
5.7	Outras funções desempenhadas	37
6	Conclusões	38
	Referências	39

Lista de Figuras

3.1	Logotipo Visual Studio	6
3.2	Logotipo Visual Basic	7
3.3	Logotipo ASP.NET	7
3.4	Logotipo SQL	8
3.5	Logotipo Github	8
3.6	Logotipo Postman	8
3.7	Logotipo Bootstrap	9
3.8	Logotipo Recaptcha	9
3.9	Logotipo Word	9
3.10	Logotipo Excel	10
3.11	Logotipo Powerpoint	10
3.12	Logotipo Overleaf	10
4.1	Logotipo Moloni	11
4.2	Logotipo Mailchimp	12
4.3	Logotipo E-Goi	12
4.4	Logotipo Recaptcha	13
4.5	Logotipo Easypay	13
4.6	Logotipo Bootstrap	13
4.7	Logotipo Visual Basic	14
4.8	Logotipo ASP.NET	14
4.9	Logotipo Powerpoint	14
4.10	Logotipo MestreClique	15
4.11	Logotipo Word	15
4.12	Logotipo Excel	15
4.13	Logotipo Powerpoint	15
5.1	Pedido de autenticação Api	17
5.2	Resposta do pedido de autenticação Api	18
5.3	criação das variáveis do cliente	18
5.4	criação do pedido com as variáveis	19
5.5	criação das variáveis para a fatura/recibo	19
5.6	criação do pedido com as variáveis	20

5.7	Link do download do ficheiro, https://www.moloni.pt//downloads/?h=f9b8f1f48467e5687d0051cf6373c3b9&d=433562911	20
5.8	Exemplo newsletter	22
5.9	Imports do projeto e-goi	22
5.10	Variáveis global do projeto e-goi	22
5.11	Download da string em Html e substituição de caracteres	23
5.12	Envio de pedido da Api e-goi	23
5.13	E-mail recebido através da E-goi	23
5.14	Formulário do protótipo	24
5.15	Pedido criação pagamento simples	25
5.16	Imports	26
5.17	Variáveis globais de autenticação	26
5.18	Pedido para a api	26
5.19	Função conversão do json em tabelas	27
5.20	Tabela "method" da dataset	27
5.21	Resultado final pedido api EasyPay	28
5.22	Javascript da Api	28
5.23	Função OnSubmit	28
5.24	Estrutura e botão	29
5.25	Resultado final	29
5.26	Ações de Formação PackWeb antigo	30
5.27	Ações de Formação PackWeb novo	30
5.28	Painel inicial de uma formação PackWeb antigo	31
5.29	Painel inicial de uma formação PackWeb novo	31
5.30	Área pessoal PackWeb antigo	31
5.31	Área pessoal PackWeb novo	32
5.32	Exemplo tabelas responsivas ambiente desktop	32
5.33	Exemplo tabelas responsivas ambiente mobile	32
5.34	Importes das bibliotecas e inicialização da variável global	34
5.35	Page_load	35
5.36	Primeira parte da função colocaimagem	35
5.37	Segunda parte da função colocaimagem	36
5.38	Função colocatexto	36
5.39	Resultado final	37

Lista de Símbolos

Símbolo	Descrição	
<i>API</i>	Application Programming Interface	API
<i>CSS</i>	Cascading Style Sheets	CSS
<i>HTML</i>	HyperText Markup Language	HTML
<i>SQL</i>	Structured Query Language	SQL
<i>VB</i>	Visual Basic	VB

Capítulo 1

Introdução

1.1 Objetivos

O objectivo geral deste estágio é colaborar no desenvolvimento da plataforma MestreClique, que conta com quase 20 anos de desenvolvimento, aprendendo e adquirindo o máximo de experiência possível pela jornada fora, colocando também em prática os ensinamentos adquiridos ao longo do ano e meio de aulas que tivemos. Várias integrações foram implementadas na plataforma da MestreClique, funcionalidades que melhoram o sistema e facilitam o uso do utilizador, mantendo também a segurança na informação que circula na plataforma, e formas de avaliar se o utilizador é humano ou não, foram objectivos realizados durante este estágio.

1.2 Calendarização

Início a dia 2 de Março de 2021. Finalizado a dia 14 de Julho de 2021. Horário das 09:30 à 13:00 e 14:00 às 18:30.

1.3 Organização do Relatório

O capítulo 1 é a introdução a este relatório, onde são relatados os objetivos, o plano de trabalho, e a calendarização do estágio.

O capítulo 2 é dedicado à empresa que me acolheu neste estágio, a MestreClique. Este capítulo é um resumo do que é a MestreClique e dos seus colaboradores, tal como a plataforma e a caracterização do espaço e ambiente em si.

O capítulo 3 é um resumo de todas as ferramentas e aplicações que foram usadas durante este estágio.

O capítulo 4 define as tarefas que foram realizadas durante todo o estágio.

O capítulo 5 desenvolve todas as tarefas que foram feitas durante o estágio.

O capítulo 6 é o capítulo da conclusão deste longo trabalho e deste relatório de estágio.

Capítulo 2

MestreClique Lda

2.1 A Empresa

A Mestreclique – Sistemas de Informação, Lda, foi constituída por Ricardo Dias e Rui Pereira em 2002 e desde então tem marcado a diferença no mundo tecnológico cada vez mais exigente.

O plano concebido para constituição da empresa, desde a primeira hora, passava pela criação de Software para gerir formação. Aliando o "Know-How" de ambos os sócios nas duas áreas fulcrais (Gestão da Formação com o Desenvolvimento de Software), nasceu a Plataforma Mestreclique.

Durante os primeiros anos, para além da evolução da Plataforma, a Mestreclique expandiu a sua atividade para o mercado das TI, com espaço comercial aberto ao público para venda de hardware, software genérico, de gestão comercial e outros serviços.

Estabeleceu parcerias com grandes referências do mercado: XD, SAGE, Microsoft, Kaspersky, etc.

Atualmente com dezenas de clientes a gerirem formação com a Plataforma Mestreclique em todo o território nacional e Angola, a empresa está segmentada nos departamentos:

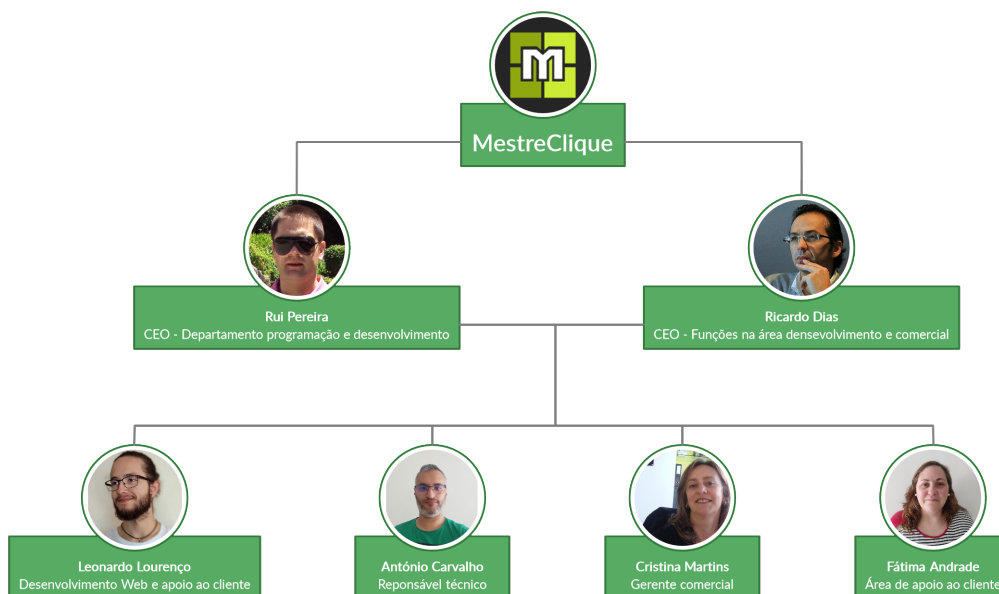
- I&D (Programação e Web Design);
- Comercial (outdoor/indoor);
- Serviços HelpDesk/Pós-Venda.

A Plataforma Mestreclique está desenvolvida em tecnologia “.net”, permitindo a gestão da formação nas componentes pedagógica, financeira e web.

2.2 Colaboradores da empresa

No momento em que este estágio decorreu a equipa da MestreClique contava com 9 colaboradores, na liderança da empresa e fundadores da mesma, o Dr. Eng. Ricardo

Dias e o Dr. Eng. Rui Pereira. No atendimento e gestão comercial, a Eng. Cristina Martins, no atendimento ao público, a Eng. Fátima Andrade, na programação e desenvolvimento web e apoio ao cliente o Eng. Leonardo Lourenço, e o responsável técnico da empresa o Eng. António Carvalho. A empresa contava ainda com mais 3 estagiários, Rui Costa, o próprio, André Godinho, o estagiário do Instituto Politécnico da Guarda do Curso Tesp de Cybersegurança, e o Gonçalo Carneiro, o estagiário do curso Técnico de Informática Instalação e Gestão de Redes da Ensi-guarda Escola Profissional da Guarda. Perto do fim do estágio entrou também para a equipa outro estagiário da parte do Instituto Politécnico da Guarda da Licenciatura de Engenharia Informática o aluno Vital Gonçalves.



2.3 Plataforma MestreClique

A plataforma MestreClique divide-se em dois projetos principais, GestForm e PackWebForm. GestForm é a aplicação desktop da MestreClique que permite gerir Entidades, Acções, Relatórios e muitas outras coisas dentro do mundo da Formação, facilitando o típico serviço de escritório e tornando-o mais eficiente. PackWebForm é a parte website da plataforma MestreClique, que facilita também o serviço, e organiza principalmente o trabalho e formações. A plataforma tem ainda mais dois projectos, GestConsult e PackWebConsult, que são a extensão de ambos os projectos mencionados anteriormente.

2.4 Caracterização do espaço e ambiente

Uma empresa deste calibre necessita das melhores condições possíveis para os seus colaboradores. A MestreClique é constituída por uma junção de duas lojas. Situada no rés-do-chão de um prédio, a loja A, loja principal, é onde está situada a secretaria da loja, um espaço amplo com duas cadeiras para os clientes, e uma mesa e cadeira de escritório para a secretária Eng. Cristina Martins. Atrás desta divisão está a sala onde se encontra a Eng. Fátima Andrade, é a sala mais pequena, e é onde se faz o atendimento ao público telefónico. Depois vem o escritório do Dr. Eng. Ricardo Dias, um escritório com duas mesas e várias cadeiras, porém nesta divisão está apenas o Dr. Eng. Ricardo Dias e é também onde se realizam as reuniões. Depois encontra-se o corredor que dá acesso às casas de banho, à segunda loja e ainda à divisão do Eng. António Carvalho. Essa divisão é usada pelo mesmo para as reparações e arrumação. É também neste corredor que se encontra a nossa área de lazer, que inclui tudo que é necessário para as nossas pausas e convívio. Depois na loja B existem duas salas amplas, uma que foi usada pelos estagiários, com duas mesas enormes e várias cadeiras, antigamente servia como sala de reuniões com os clientes, e existe ainda outra sala que é a sala de desenvolvimento, onde o Eng. Leonardo Lourenço e o Dr. Eng. Rui Pereira têm os seus momentos de concentração para continuar a desenvolver os seus trabalhos. É um espaço acolhedor com quadros em vários sítios para esclarecimento de dúvidas ou realização de "brainstormings".

Capítulo 3

Tecnologia utilizada

3.1 Visual Studio 2019:

Microsoft Visual Studio é um ambiente de desenvolvimento integrado (IDE) da Microsoft para desenvolvimento de software especialmente dedicado ao .NET Framework e às linguagens Visual Basic (VB), C, C++, C# (C Sharp) e F# (F Sharp). Também é um produto para o desenvolvimento na área web, usando a plataforma do ASP.NET, como websites, aplicativos web, serviços web e aplicativos móveis.

Optou-se por usar o Visual Studio 2019 como ambiente de desenvolvimento integrado devido á experiência já adquirida durante as aulas e também por ser o IDE utilizado para desenvolvimento na MestreClique.



Figura 3.1: Logotipo Visual Studio

3.2 Visual Basic:

O Visual Basic é uma linguagem de programação produzida pela empresa Microsoft, e é parte integrante do pacote Microsoft Visual Studio. A sua versão mais recente faz parte do pacote Visual Studio .NET, voltada para aplicações .Net. Sua versão anterior fez parte do Microsoft Visual Studio 6.0, e é ainda muito utilizado atualmente por aplicações legadas. Embora o estagiário estivesse mais familiarizado com C# e Java, optou-se por programar a maior parte dos projetos em VB, por ser a linguagem de programação predominante na empresa. Sendo o VB parecido com C# houve poucas dificuldades no que toca ao uso de linguagem de programação, recorrendo também a sites de conversão no início para facilitar o trabalho.



Figura 3.2: Logotipo Visual Basic

3.3 Asp.Net:

O ASP.NET é a plataforma da Microsoft para o desenvolvimento de aplicações Web e é o sucessor da tecnologia ASP. Permite, através de uma linguagem de programação integrada na .NET Framework, criar páginas dinâmicas. Não é nem uma linguagem de programação como a VBScript, o PHP, nem um servidor web como IIS ou Apache.

O ASP.NET é baseado no Framework .NET e herda todas as suas características, por isso, como qualquer aplicação .NET, as aplicações para essa plataforma podem ser escritas em várias linguagens, como C#, F# e Visual Basic.NET.

Uma solução viável porque já se encontra entregue na plataforma desde os primeiros dias de vida da empresa, e foi o intuito de aprender um pouco mais sobre esta linguagem, que continuámos a trabalhar com esta ferramenta..



Figura 3.3: Logotipo ASP.NET

3.4 SQL Manager:

A Structured Query Language, ou Linguagem de Consulta Estruturada ou SQL, é a linguagem de pesquisa declarativa padrão para base de dados relacional (base de dados relacional). Muitas das características originais do SQL foram inspiradas na álgebra relacional.

Uma ferramenta essencial para um software que faz gestão de formação, SQL é o software de eleição que gere a base de dados da MestreClique.



Figura 3.4: Logotipo SQL

3.5 GitHub:

O GitHub é uma plataforma de hospedagem de código-fonte e arquivos com controle de versão usando o Git. Ele permite que programadores, utilitários ou qualquer usuário cadastrado na plataforma contribuam em projetos privados e/ou Open Source de qualquer lugar do mundo. O GitHub é amplamente utilizado por programadores para divulgação dos seus trabalhos ou para que outros programadores contribuam para o projeto, além de promover fácil comunicação através de recursos que relatam problemas ou mesclam repositórios remotos (issues, pull request).



Figura 3.5: Logotipo Github

3.6 Postman:

Postman é a plataforma perfeita para o desenvolvimento com API's, utilizando esta plataforma é possível fazer pedidos a api's e analisar a informação que nos é remetida.



Figura 3.6: Logotipo Postman

3.7 Bootstrap:

Bootstrap, uma framework que facilita o trabalho dos programadores Html, Css e JavaScript, baseado em modelos de design para a tipografia, melhora a experiencia do utilizador nos sites, pois torna-o um site mais responsável e amigável de usar.



Figura 3.7: Logotipo Bootstrap

3.8 Recaptcha:

O Recaptcha é um serviço de segurança que protege os sites de fraudes e abusos, usando uma análise avançada de riscos, o recaptcha protege os sites de software malignos e abusivos, bloqueando todos os utilizadores que forem avaliados como falsos ou maliciosos.



Figura 3.8: Logotipo Recaptcha

3.9 Ferramentas Microsoft:

Foram também utilizadas várias ferramentas microsoft, mais propriamente ferramentas de processamento e edição de texto.



Figura 3.9: Logotipo Word

3.10 Overleaf:

O Overleaf é um editor LaTeX colaborativo baseado em nuvem usado para escrever, editar e publicar documentos científicos. Tem parceria com uma ampla gama



Figura 3.10: Logotipo Excel

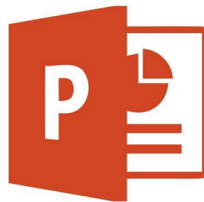


Figura 3.11: Logotipo Powerpoint

de editores científicos para fornecer modelos LaTeX de revistas científicas e links de submissão direta. É esta a ferramenta utilizada para o desenvolvimento deste relatório de estágio.



Figura 3.12: Logotipo Overleaf

Capítulo 4

Tarefas realizadas

4.1 Descrição das principais tarefas desenvolvidas durante o estágio:

4.1.1 Integração com a API Moloni:

A Moloni é uma empresa portuguesa que fornece um software de gestão comercial, porém a nossa abordagem com a mesma é apenas para utensílio da sua API de facturação, que permite através de pedidos efectuar a gestão do stock, facturas, recibos, e todo o tipo de documentos de facturação. A integração com a Moloni foi sem dúvida o maior desafio deste estágio. É preciso fazer uma formatação detalhada e especifica dos dados para estabelecer um pedido correctamente, qualquer tipo de erro nos dados irá gerar uma falha no pedido e cancelar assim o mesmo pedido. A Moloni é extremamente exigente na utilização de dados, o que faz sentido pois é uma empresa que trabalha directamente com facturação. Foi também a primeira abordagem que tive com Api's, o que tornou um pouco difícil o trabalho e demoroso pois exigiu um nível bastante elevado de pesquisas e estudo.



Figura 4.1: Logotipo Moloni

4.1.2 Integração com a API Mailchimp:

Mailchimp é uma empresa das mais conhecidas a nível de marketing e comunicação via e-mail e SMS. Foi feita uma abordagem com a mailchimp a fim de encontrar uma solução para os nossos clientes poderem enviar as suas "newsletters" e e-mails mais facilmente, mas os custos desta Api e as suas limitações a níveis de números de e-mails diários/mensais tornaram esta Api uma solução pouco provável para o nosso problema, e, após uma reunião decisiva, optámos por dar como finalizada esta integração, e optar por outras soluções.

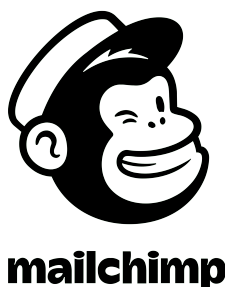


Figura 4.2: Logotipo Mailchimp

4.1.3 Integração com a API E-goi:

A E-goi é também uma empresa de marketing porém não tão conhecida como a Mailchimp. Apesar da Mailchimp ser uma empresa maior, a E-goi trouxe-nos mais vantagens a nível de integração com outras APIs, e também a níveis monetários, acabando assim por se tornar na nova solução para envios de e-mails e Newsletter com corpo em Html.



Figura 4.3: Logotipo E-Goi

4.1.4 Integração do PackWeb com a API Recaptcha do Google:

Recaptcha permite fazer análises ao utilizador que efectua as acções, impedindo assim software malicioso de fazer ataques/pedidos de informação abusivos. Qualquer empresa sofre vários ataques, com intuito de retirar alguma informação que possam usar posteriormente ou para outros fins maliciosos, é aqui que a Recaptcha da Google torna-se uma solução viável, monitorizando as acções da pessoa que faz o pedido, e avaliando o perigo que esse pedido representa.



Figura 4.4: Logotipo Recaptcha

4.1.5 Atualização para a mais recente versão da API EasyPay da plataforma MestreClique:

A EasyPay é a instituição pioneira em Portugal de pagamentos online, permitindo todo o tipo de pagamentos online, e a prestações. A EasyPay é a empresa número um em Portugal no ramo.



Figura 4.5: Logotipo Easypay

4.1.6 Integração da Framework Bootstrap no PackWeb MestreClique:

Bootstrap é uma framework para Html que facilita o trabalho de um programador em todos os aspetos possíveis, pois já dispõe das suas bibliotecas Css sendo assim extremamente fácil de integrar em qualquer ambiente.



Figura 4.6: Logotipo Bootstrap

4.1.7 Desenvolvimento de website em VB/asp.net:

A MestreClique usa Visual Basic e Asp.net para programar a maior parte dos seus projectos.



Figura 4.7: Logotipo Visual Basic



Figura 4.8: Logotipo ASP.NET

4.1.8 Criação de apresentações PowerPoint a partir de um tema (visual basic):

A plataforma da MestreClique já constrói os formulários/modelos da plataforma em Word e Excel, mas devido a vários pedidos dos clientes para esta nova abordagem com o PowerPoint, desenvolvemos uma maneira de criar apresentações Powerpoint através do template fornecido pelos clientes. Resumidamente o pedido dos clientes seria para usar o template que eles fornecem e colocar logótipos e texto com informação específica das acções que estão a instruir, facilitando assim o trabalho deles entregando um modelo Powerpoint pronto para o cliente usar como entender, com os logótipos e texto sobre a acção.

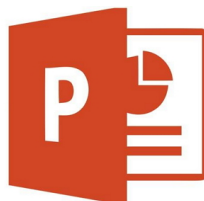


Figura 4.9: Logotipo Powerpoint

4.1.9 Atendimento ao cliente e acesso remoto para instalação e manutenção da plataforma MestreClique:

Uma empresa como a MestreClique tem uma vasta carteira de clientes. Por vezes exercemos também o atendimento a cliente para resolver problemas menores, e também para fazer a instalação da nossa plataforma no computador dos clientes.



Figura 4.10: Logotipo MestreClique

4.1.10 Gestão documental:

Sendo a MestreClique uma plataforma para gerir formação, o uso de software da Microsoft como "Word", "Excel", entre outros, torna-se inevitável e é mesmo um ponto chave. Assim desta forma por vezes deparamos-nos com situações de actualizar algum documento que já esteja na produção durante algum tempo, muita das vezes a pedido do cliente.



Figura 4.11: Logotipo Word



Figura 4.12: Logotipo Excel



Figura 4.13: Logotipo Powerpoint

Capítulo 5

Desenvolvimento

Devido à plataforma MestreClique não ser um software Open Source, vejo-me um pouco limitado no que posso ou não mostrar neste relatório. Foram então criados alguns "protótipos" de exemplo para mostrar algum do trabalho que foi feito durante este estágio.

5.1 Api Moloni

A Api da Moloni está mais desenvolvida do que qualquer outro tipo de empresa que tive até hoje oportunidade de trabalhar. Este foi o primeiro projeto que desenvolvi na MestreClique, e foi sem dúvida o mais difícil de fazer. As minhas noções sobre Api's eram poucas, e nesta altura estava a trabalhar em tele-trabalho e havia um fluxo enorme de trabalho na empresa, o que fez com que o meu supervisor tivesse pouco tempo para me disponibilizar. Posso dizer sem dúvida alguma que os primeiros dias foram apenas de pesquisa e estudo sobre o funcionamento das Api's, tipos de pedidos existentes, etc.

Fico grato por a minha primeira experiência com Api's tenha sido com a da Moloni, esta empresa dispõe de uma documentação incrível no que toca a desenvolvimento e configurações da Api.

Existe toda uma documentação que pode ser acedida no seguinte site:

<https://www.moloni.pt/dev/endpoints/>

Existe ainda também um postman com todos os pedidos possíveis que existem para esta Api: <https://documenter.getpostman.com/view/1519683/SVfNwq34> .

Este Postman está disponível em vinte e sete linguagens de código diferentes. Para além disso, a Moloni ainda disponibiliza uma sandbox para os programadores de aplicações fazerem os testes necessários sem problema algum. A MestreClique conseguiu ainda uma conta de testes para que eu conseguisse fazer todo o tipo de testes necessário para garantir a integração da moloni com a MestreClique.

O que se pretende com esta implementação é fornecer uma forma fácil de os nossos clientes obterem documentos de faturação/recibos através da nossa Api. Para isso, é necessário criar uma conta moloni e configurar a conta para poder usar a parte Api. No menu configurações existe uma página dedicada às configurações de api e plugins, é aqui que se ativa a versão Api da conta, e que se configura todo o tipo de dados necessários para fazer os pedidos. A moloni tem ainda integração com a E-goí, o que ainda não foi desenvolvido mas será uma integração muito interessante pois vai facilitar que os recibos sejam diretamente enviados por e-mail para os contactos da E-goí quando for efetuado uma nova fatura/recibo.

Claro que a Moloni só tem este nível de excelência, porque também exige um nível de excelência no que toca aos critérios de dados que são enviados na Api.

Como não me é possível mostrar o código fonte da aplicação da MestreClique, vamos então fazer um processo simples de facturar um curso a um cliente, exemplificando com exemplos retirados do Postman os pedidos necessários para fazer uma fatura/recibo.

A Moloni funciona com tokens para fazer a autenticação, para além de ser necessário também o "Secret-id" da conta, e o "Developer ID", estes, podem ser obtidos através das configurações de programador no site da moloni, para podermos fazer qualquer tipo de pedido será necessário fazer um pedido de autenticação primeiro.

Este será o primeiro pedido a fazer à Api apenas para autenticar e obter o token para fazer as outras chamadas.

Authentication

GET grant/password

```
{{access_url}}v1/grant?grant_type=password&client_id={{client_id}}&client_secret={{client_secret}}&username={{username}}&password={{password}}
```

Private

HEADERS

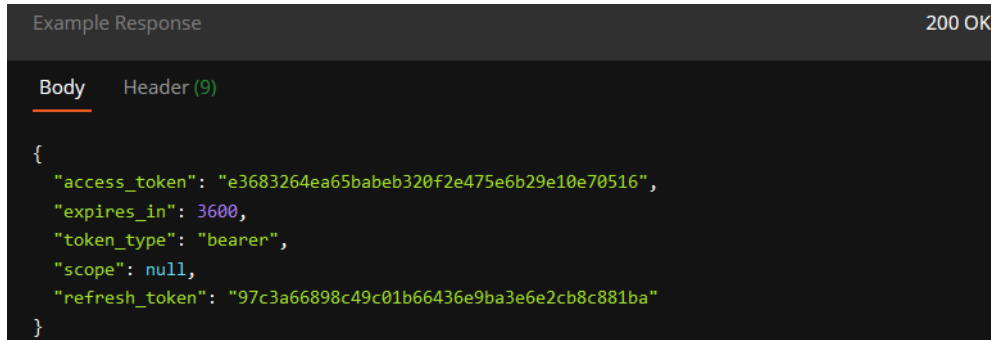
Content-Type	application/x-www-form-urlencoded
---------------------	-----------------------------------

PARAMS

grant_type	password
client_id	{{client_id}}
client_secret	{{client_secret}}
username	{{username}}
password	{{password}}

Figura 5.1: Pedido de autenticação Api

A resposta vem em json e retorna dois tokens, um token que será utilizável por 3600 segundos, e outro token que será necessário para fazer um novo pedido de autenticação caso o primeiro token seja expirado.



```

Example Response 200 OK
Body Header (9)
{
  "access_token": "e3683264ea65babeb320f2e475e6b29e10e70516",
  "expires_in": 3600,
  "token_type": "bearer",
  "scope": null,
  "refresh_token": "97c3a66898c49c01b66436e9ba3e6e2cb8c881ba"
}

```

Figura 5.2: Resposta do pedido de autenticação Api

Depois da autenticação feita e com o token na nossa pose, podemos fazer qualquer pedido que assim desejarmos. Neste caso para a realização de uma fatura/recibo, primeiro temos de criar um categoria de produto, caso esta ainda não exista, e criar um produto dentro dessa categoria. Temos também de criar um cliente para a atribuir na fatura e só depois disso conseguimos retirar a fatura com sucesso.

Vamos agora analisar um pedido de introdução de cliente no sistema da moloni.

Criamos primeiro as variáveis, como ilustrado na figura 5.3, e só depois atribuímos as variáveis aos parâmetros do pedido, figura 5.4.

```

Dim company_id As Integer = 172616
Dim vat As String = "999999990"
Dim number As String = "01"
Dim name As String = "Novo Cliente"
Dim language_id As Integer = 1
Dim address As String = "Rua da rua"
Dim zip_code As String = "6300-000"
Dim city As String = "Guarda"
Dim country_id As Integer = 1
Dim email As String = "novocliente@gmail.com"
Dim website As String = "https://www.NovoCliente.pt"
Dim phone As String = "+351 910 000 000"
Dim fax As String = "+351 920 000 000"
Dim contact_name As String = "experiencia"
Dim contact_email As String = "experiencia@gmail.com"
Dim contact_phone As String = "999999999"
Dim notes As String = "experiencia"
Dim salesman_id As Integer = 0
Dim price_class_id As Integer = 1
Dim maturity_date_id As Integer = 1
Dim payment_day As Integer = 1
Dim discount As Single = 0
Dim credit_limit As Single = 0
Dim document_type_id As Integer = 1
Dim copies As Integer = 1
Dim payment_method_id As Integer = 1196215
Dim delivery_method_id As Integer = 0
Dim field_notes As String = "experiencia"

```

Figura 5.3: criação das variáveis do cliente

Depois fazemos o pedido usando as variáveis.

```

Try
    Dim client = New RestClient(access_url & "v1/customers/insert/?access_token=" + access_token)
    client.Timeout = -1
    Dim request = New RestRequest(Method.POST)
    request.AddHeader("Content-Type", "application/x-www-form-urlencoded")
    request.AddParameter("company_id", company_id)
    request.AddParameter("vat", vat)
    request.AddParameter("number", number)
    request.AddParameter("name", name)
    request.AddParameter("language_id", language_id)
    request.AddParameter("address", address)
    request.AddParameter("zip_code", zip_code)
    request.AddParameter("city", city)
    request.AddParameter("country_id", country_id)
    request.AddParameter("email", email)
    request.AddParameter("phone", phone)
    request.AddParameter("salesman_id", salesman_id)
    request.AddParameter("maturity_date_id", maturity_date_id)
    request.AddParameter("payment_day", payment_day)
    request.AddParameter("discount", discount)
    request.AddParameter("credit_limit", credit_limit)
    request.AddParameter("copies[0][document_type_id]", document_type_id)
    request.AddParameter("copies[0][copies]", copies)
    request.AddParameter("payment_method_id", payment_method_id)
    request.AddParameter("delivery_method_id", delivery_method_id)
    Dim response As IRestResponse = client.Execute(request)
    Console.WriteLine(response.Content)
    Label2.Text = response.Content
Catch __unusedException1__ As Exception
    Throw
End Try

```

Figura 5.4: criação do pedido com as variáveis

Os outros pedidos são bastante semelhantes, por isso vamos passar logo para o ultimo pedido, o pedido que nos irá devolver o link para aceder-mos á fatura/recibo que pretendemos.

Vamos mostrar de novo os dados das variáveis e a atribuição das variáveis ao pedido.

```

Dim company_id As Integer = 172616
Dim data As String = "2021-08-21"
Dim expiration_date As String = "2021-10-30"
Dim document_set_id As Integer = 385846
Dim customer_id As Integer = 43693429
Dim alternate_address_id As Integer = 0
Dim our_reference As String = "Nossa referência"
Dim your_reference As String = "Referência Cliente"
Dim financial_discount As Single = 0
Dim eac_id As Integer = 0
Dim salesman_id As Integer = 0
Dim salesman_commission As Single = 0
Dim special_discount As Single = 0
Dim product_id As Integer = 79335444
Dim name As String = "Curso de culinária"
Dim summary As String = "Curso de culinária"
Dim qty As Single = 1
Dim price As Single = 100
Dim discount As Single = 0
Dim deduction_id As Integer = 0
Dim order As Integer = 1
Dim warehouse_id As Integer = 0
Dim tax_id As Integer = 2155418
Dim value As Single = 23
Dim tax_order As Integer = 0
Dim cumulative As Integer = 1
Dim notes As String = "Pago por MB XXXXX"
Dim status As Integer = 1

```

Figura 5.5: criação das variáveis para a fatura/recibo


```

Dim client = New RestClient(access_url & "v1/invoiceReceipts/insert/?access_token=" + acess_token)
client.Timeout = -1
Dim request = New RestRequest(Method.POST)
request.AddHeader("Content-Type", "application/x-www-form-urlencoded")
request.AddParameter("company_id", company_id)
request.AddParameter("date", data)
request.AddParameter("expiration_date", expiration_date)
request.AddParameter("document_set_id", document_set_id)
request.AddParameter("customer_id", customer_id)
request.AddParameter("your_reference", your_reference)
request.AddParameter("salesman_id", salesman_id)
request.AddParameter("products[0][product_id]", product_id)
request.AddParameter("products[0][name]", name)
request.AddParameter("products[0][summary]", summary)
request.AddParameter("products[0][qty]", qty)
request.AddParameter("products[0][price]", price)
request.AddParameter("products[0][deduction_id]", deduction_id)
request.AddParameter("products[0][order]", order)
request.AddParameter("products[0][taxes][0][tax_id]", tax_id)
request.AddParameter("products[0][taxes][0][value]", value)
request.AddParameter("notes", notes)
request.AddParameter("status", status)
Dim response As IRestResponse = client.Execute(request)
Console.WriteLine(response.Content)

```

Figura 5.6: criação do pedido com as variáveis

Após a conclusão do pedido, será devolvido o Id do novo documento criado, em seguida é só fazer um pedido para obter o link do download do documento e o resultado final encaminhará para uma página onde conseguimos ver qual a empresa que faturou, para quem faturou, e o link para conseguirmos descarregar o documento.

Na figura 5.7 conseguimos ver a página de download do documento, na legenda está um link válido até dia 30 de Outubro de 2021 que nos permite descarregar o ficheiro PDF da fatura/recibo.

MESTRECLIQUE
SISTEMAS DE INFORMAÇÃO Lda

Av. Monsenhor Mendes do Carmo, Nº12 A e B
6300-586 Guarda
Portugal

Telefone: 271 227 354
E-mail: geral@mestrelclique.pt
Internet: https://mestrelclique.pt/

Contribuinte: MESTRECLIQUE_TESTES_API

Novo Cliente
Rua da rua
6300-000 Guarda
Portugal

Contribuinte: 999999990

Documentos para download

Tipo de Documento	Número	Data de Emissão	Data Vencimento	V/ Refº	Valor
Fatura/Recibo	M/3	21-08-2021	30-10-2021	Referência Cliente	123,00€

[Download do ficheiro](#)

© 2021 Powered by Moloni - Software de Faturação Online
Certificado pela Autoridade Tributária Nº 2860

Figura 5.7: Link do download do ficheiro, <https://www.moloni.pt/downloads///?h=f9b8f1f48467e5687d0051cf6373c3b9&d=433562911>

5.2 Api E-goi

O objetivo desta implementação com a E-goi é fornecer aos nossos clientes uma maneira fácil e eficaz de enviar mensagem de e-mail, telemóvel, mas principalmente mensagens de e-mail com as newsletters que os clientes já têm. Após um estudo de mercado a implementação com a E-goi pareceu uma resposta muito melhor do que a mailchimp, que apesar de ter sido implementada, acabou por decidir-se que não seria uma opção válida, devido a um preço extremamente alto comparado com a e-goi, e com menos envios de e-mail e mensagem.

A e-goi permite enviar estas newsletter com um pedido simples da api de envio de e-mail, mas para isso o corpo da newsletter tem de estar em HTML. Aqui é que nos deparamos com o primeiro problema, a maneira como estão escritas estas newsletters. Algumas já são bastante antigas, e contêm caracteres desnecessários, como por exemplo espaços a mais entre campos, parágrafos, entre outros.

Como mais uma vez não me é permitido mostrar o código fonte da MestreClique, foi criado um protótipo apenas para esclarecimento. Este é bastante simples e mostra apenas o envio de uma newsletter através de um pedido simples de envio de e-mail.

Antes de tudo, é preciso criar uma conta E-goi. É necessário também disponibilizar um domínio para a E-goi conseguir usar os nossos dados para enviar uma mensagem como se fossemos nós mesmos. Após a conta criada a primeira coisa a fazer é ir ao menu de utilizador criar um remetente para os envios.

Irá ser necessário primeiro autenticar o e-mail do remetente que estamos a tentar criar. Será enviada uma mensagem para o e-mail de confirmação e autenticação.

Seguidamente, a e-goi pede que sejam criados Cname no nosso serviço DNS, e para isso, eles têm três opções:

- A primeira opção é sermos nós mesmos a fazer, a e-goi irá dizer-nos o que fazer e o que temos de criar para ser autenticado o nosso domínio;

- A segunda opção permite que a e-goi envie instruções de como fazer o mesmo processo a técnicos. Este caso é importante se for uma empresa que tem alguém a tratar da parte da informática;

- A terceira e última opção, permite que sejam eles a fazer isso por nós, fornecendo as credenciais de acesso ao painel de domínios para que um dos técnicos da e-goi consiga fazer essa autenticação.

O colaborador António Carvalho faz essa gestão, pedi então ao técnico que, através da opção um, criasse o que fosse preciso para autenticar a e-goi nos nossos domínios, o que apesar de ser bastante fácil, demorou cerca de 2 dias para propagar esse serviço e conseguir por a e-goi a funcionar sem problema algum.

Como exemplo vamos mostrar o envio de uma newsletter de um cliente, figura 5.8.



Figura 5.8: Exemplo newsletter

Como ilustrado na figura 5.9, este projeto necessita de 4 imports.

```
Imports System.Text
Imports RestSharp
Imports System.Web
Imports System.IO
```

Figura 5.9: Imports do projeto e-goi

As variáveis globais que existem neste projeto são as variáveis para os parâmetros de envio do pedido da Api, figura 5.10, e uma outra variável do tipo "System.Net.WebClient" que iremos usar para fazer o download da string em HTML da newsletter que queremos enviar, figura 5.11.

```
Dim Name As String = "Rui"
Dim email As String = "ruicosta17@live.com.pt"
Dim subject As String = "E-mail MestreClique"
Dim Message As String = "teste"
Dim Message1 As String = "teste"
Dim w As New System.Net.WebClient
```

Figura 5.10: Variáveis global do projeto e-goi

Após o download da string, é necessário formatar de acordo com os critérios da e-goi, e para isso teremos de tratar de tudo que seja quebras de linhas, espaços a mais entre outros, e de seguida, com a string formatada como deve ser, podemos proceder ao envio do pedido da Api, figura 5.12.

```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    w.Encoding = Encoding.UTF8
    w.Encoding = UTF8Encoding.UTF8
    Message = w.DownloadString("https://pt/newscurso.aspx?id=11&acao=338")
    Dim myString AS String = Message.Replace(System.Environment.NewLine, "").Replace(vbTab, "").Replace(Chr(10), "").Replace(Chr(13), "")
    Message1 = myString
    SendMail()
End Sub
```

Figura 5.11: Download da string em Html e substituição de caracteres

O envio deste pedido é feito de uma maneira diferente, a e-go exige que o pedido da Api seja apenas o link do pedido que estamos a efetuar. Exige também que conteha os headers "content-type" e um outro chamado "ApiKey", onde introduzimos a chave que nos foi fornecida antes da api. E no final exige que todo o corpo do pedido seja apenas um parâmetro do tipo "ParameterType.RequestBody". Este parâmetro tem de ser feito em json, um método um pouco invulgar para Api's nestes tempos.

```
Sub SendMail()
    Dim client = New RestClient("https://sigatoh.epilago.com/api/v2/email/messages/action/send")
    Client.Timeout = 5
    Dim request = New RestRequest(Method.POST)
    request.AddHeader("ApiKey", [redacted])
    request.AddHeader("Content-Type", "application/json")
    request.AddHeader("Host: sigatoh.epilago.com", "[redacted]")
    Dim response As IRestResponse = client.Execute(request)
    Dim body = response.Content
    Label1.Text = response.Content
End Sub
```

Figura 5.12: Envio de pedido da Api e-go

Desta maneira conseguimos enviar não só mensagens simples, mas também as pedidas newsletters que os nossos clientes tanto pedem para enviar.

E-mail **MestreClique**

MestreClique Lda <info@mestreclique.pt>
06/05/2021 11:30
Para: fulcoosta7@live.com.pt

Se não visualizar corretamente este e-mail, [clique aqui](#)





REABILITAÇÃO DE DENTES ENDODONCIADOS

Formadores: [redacted] 15 e 16 maio '21

- Com este curso pretende-se capacitar os formandos com competências para executarem diagnóstico e planeamento.
- Dominar as diferentes opções e técnicas de restauração de dentes endodoncizados tendo em conta as suas indicações, vantagens e desvantagens.

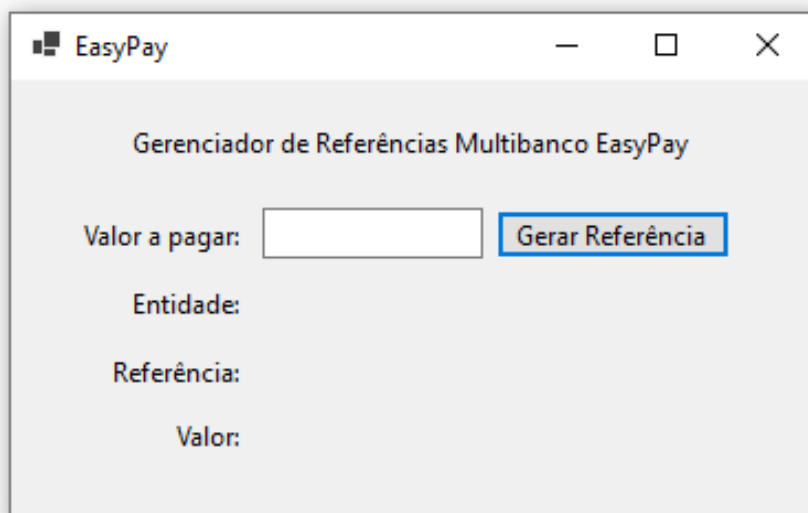
Visite o nosso site [inscreva-se aqui](#)

Se não puder visualizar as imagens corretamente, [clique aqui](#)

Figura 5.13: E-mail recebido através da E-go

5.3 Api EasyPay

Esta implementação foi apenas uma atualização da Api, já implementada na plataforma da Mestreclique, para a ultima versão, e como não me é permitido mostrar o código fonte da plataforma MestreClique, criei então um protótipo simples para demonstração. Este protótipo contém apenas um formulário para indicação do valor que pretende gerar a referência de multibanco, figura 5.14.



The image shows a browser window with the title 'EasyPay'. The main content area has a title 'Gerenciador de Referências Multibanco EasyPay'. Below the title, there is a form with the following elements: a label 'Valor a pagar:' followed by a text input field, a button labeled 'Gerar Referência', a label 'Entidade:', a label 'Referência:', and a label 'Valor:'.

Figura 5.14: Formulário do protótipo

Relembrando que a MestreClique já dispõe de várias formas de pagamento através de outros serviços de Api's, esta foi apenas mais uma tarefa que me foi atribuída de maneira a aprender uma outra forma de fazer chamadas a Api's. Enquanto que por exemplo as chamadas da Api Moloni são feitas com "add.parameters" de maneira a facilitar o controlo de cada parâmetro, no caso EasyPay fazemos a atribuição dos parâmetros diretamente na criação do pedido RestClient.

Foram feitas várias integrações de pedidos, como por exemplo de cancelamento de pedido de pagamento, ou uma subscrição, mas para este exemplo vamos apenas mostrar a criação de um pedido simples de pagamento.

Estes são os critérios que necessitamos para fazer este pedido de pagamento simples com referência multibanco.

Create single payment

AUTHORIZATIONS: **accountId AND apiKey**

REQUEST BODY SCHEMA: **application/json**

Payment configuration object to generate a payment

type	string Default: "sale" Enum: "sale" "authorisation" Optional - only applicable to method cc or mbw
expiration_time	string <Y-m-d H:i> Optional
currency	string Default: "EUR" Enum: "EUR" "BRL"
customer >	object (schemas)
key	string <= 50 characters Merchant identification key
value required	number <double> >= 0.5 Value will be rounded to 2 decimals
method required	string (Payment_Single_Method_Available) Enum: "mb" "cc" "mbw" "dd" "bb" "vi"
sdd_mandate >	object (schemas) Only required if 'method' is 'dd'
capture >	object Only required if 'type' is 'sale'

Figura 5.15: Pedido criação pagamento simples

A maior parte destes parâmetros são de critério opcional. Neste pedido vamos apenas usar:

Type - Se é uma venda ou uma autorização de pagamento, neste caso vamos usar - "sale";

Name - Nome da pessoa a quem vamos passar a referência de pagamento (apenas obrigatório para pagamentos MBWay) - "Rui Costa";

Email - E-mail da pessoa a quem vamos passar a referência de pagamento (apenas obrigatório para pagamentos MBWay) - "Ruicosta17@live.com.pt";

Phone - Telemóvel da pessoa a quem vamos passar a referência de pagamento (apenas obrigatório para pagamentos MBWay) - "911234567";

Phone_indicative - Indicativo do número de telemóvel (apenas obrigatório para pagamentos MBWay) - "+351"

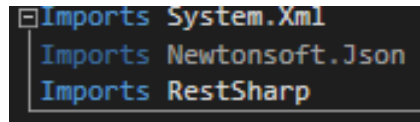
Value - valor a definir pelo utilizador

Method - Método de pagamento do pedido que vamos criar - "MB"

Descriptive - (Obrigatório) Descrição que irá aparecer nos detalhes da transferência do banco ou nos detalhes MBWay se for o caso - "Exemplo teste"

Depois de termos todos os parametros definidos, e o layout do programa definido, podemos começar a trabalhar no back-end do programa. Tudo que irá ser apresentado aqui é feito na versão Sandbox da Api EasyPay, uma versão da Api desenvolvida apenas para testes e desenvolvimento.

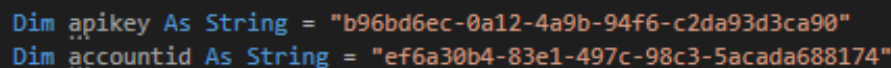
Vamos precisar de três importes neste projeto para a sua execução:



```
Imports System.Xml
Imports Newtonsoft.Json
Imports RestSharp
```

Figura 5.16: Imports

Vamos precisar também de declarar 2 variáveis globais que irão ser necessárias para autenticar a conta que está a fazer o pedido:



```
Dim apikey As String = "b96bd6ec-0a12-4a9b-94f6-c2da93d3ca90"
Dim accountid As String = "ef6a30b4-83e1-497c-98c3-5acada688174"
```

Figura 5.17: Variáveis globais de autenticação

A criação do pedido para a Api neste caso é feita num clique de um botão, e como mencionado anteriormente, ao contrário da moloni este pedido é feito praticamente todo no diretamente na criação do pedido junto com o link que identifica qual é a chamada da Api a ser executada. Este pedido é feito da seguinte maneira:

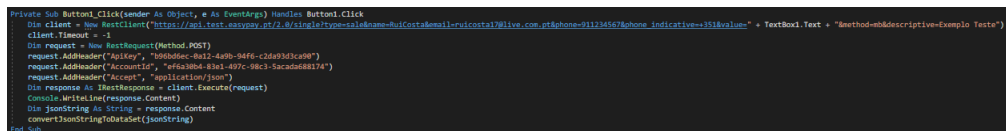
Primeiro vem a parte que esclarece qual o pedido a ser feito;

Depois vem o parametro e separado por o simbolo "=" vem o valor que queremos enviar nesse parametro. Ex.: "type=sale";

Cada vez que queremos introduzir um parametro novo é necessário indicá-lo com o simbolo "&".

Depois necessitamos de indicar também três "headers". Os "headers" servem para indicar credenciais de autenticação, o tipo do contexto que vamos receber e enviar, entre outros.

E por fim apenas precisamos de enviar o pedido e trabalhar a resposta conforme entendermos.



```
Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
    Dim client = New RestClient("https://api.test.easypay.pt/2.0/single?type=sale&nome=huicosta@live.com.pt&phone=91123457&phone_indicativo=351801101" + TextBox1.Text + "&method=mbdescriptive=Exemplo Teste")
    client.Timeout = 1
    Dim request = New RestRequest(Method.POST)
    request.AddHeader("apikey", "b96bd6ec-0a12-4a9b-94f6-c2da93d3ca90")
    request.AddHeader("accountid", "ef6a30b4-83e1-497c-98c3-5acada688174")
    request.AddHeader("accept", "application/json")
    Dim response As IRestResponse = client.Execute(request)
    Console.WriteLine(response.Content)
    Dim jsonString As String = response.Content
    convertJsonStringToDataSet(jsonString)
End Sub
```

Figura 5.18: Pedido para a api

Esta resposta irá ser um json, porém este json constitui uma série de informação desnecessária para nós, então, iremos utilizar uma função para converter o json numa "DataSet" de modo a facilitar o nosso trabalho em termos de organização de dados.

Uma "Dataset" não é nada mais nada menos do que várias tabelas introduzidas numa só. Esta dataset irá ser dividida em três tabelas:

-rootNode - tabela onde ficará gravada o resultado do pedido e as especificações do mesmo;

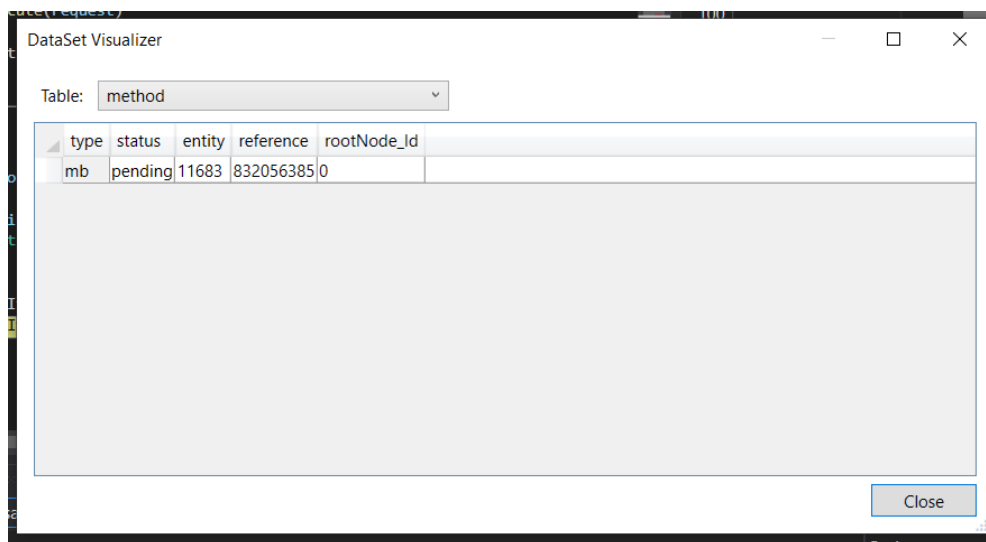
-method - tabela onde ficará gravada a informação que iremos devolver ao utilizador;

-customer - tabela onde ficará gravada o "id" do pedido que fizemos.

```
Private Function convertJsonStringToDataSet(jsonString As String) As DataSet
    Dim xd As New XmlDocument()
    jsonString = "{ "rootNode": {" + jsonString.Trim().TrimStart("{").TrimEnd("}") + " } }"
    xd = DirectCast(Newtonsoft.Json.JsonConvert.DeserializeXmlNode(jsonString), XmlDocument)
    Dim ds As New DataSet()
    ds.ReadXml(New XmlNodeReader(xd))
    Label1.Text = ds.Tables("method").Rows(0).Item("entity").ToString
    Label6.Text = ds.Tables("method").Rows(0).Item("reference").ToString
    Label7.Text = TextBox1.Text + "€"
    Return ds
End Function
```

Figura 5.19: Função conversão do json em tabelas

Desta maneira podemos trabalhar os dados que nos interessam que são apenas os dados da tabela "method".



The screenshot shows a window titled "DataSet Visualizer" with a dropdown menu set to "Table: method". Below the dropdown is a table with the following data:

type	status	entity	reference	rootNode_Id
mb	pending	11683	832056385	0

A "Close" button is visible at the bottom right of the window.

Figura 5.20: Tabela "method" da dataset

Eis o resultado final.

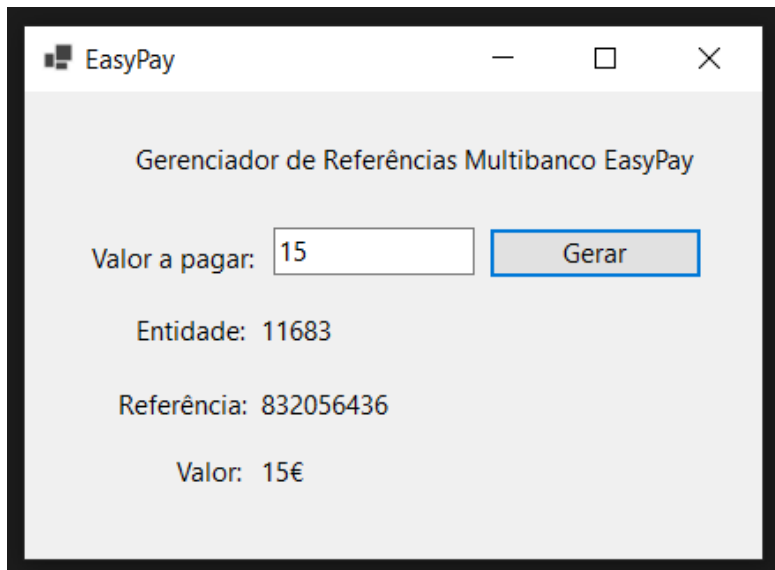


Figura 5.21: Resultado final pedido api EasyPay

5.4 Api Recaptcha

O Recaptcha é algo fácil de implementar, devido à boa documentação que existe por parte da Google, a empresa que fornece o serviço. Baseando-se neste site "<https://developers.google.com/recaptcha/docs/v3>" podemos ver que para conseguirmos implementar o recaptcha em qualquer site apenas precisamos de 3 passos após a criação da conta. O primeiro passo baseia-se no carregamento do JavaScript da API

```
<script src="https://www.google.com/recaptcha/api.js"></script>
```

Figura 5.22: Javascript da Api

De seguida, criamos a função "onSubmit" que iremos chamar no nosso botão assim que desejarmos uma avaliação do Recaptcha sobre o utilizador que está a fazer o pedido.

```
<script>
  function onSubmit(token) {
    document.getElementById('gRecaptchaResponse').value = token;
    document.getElementById('Button1').click();
  }
</script>
```

Figura 5.23: Função OnSubmit

E por último criamos o botão que irá desencadear a ação "onSubmit" para fazer a avaliação das ações que o utilizador tomou.

Algumas passaram apenas por mudanças de Css e icons mais bonitos, cores mais apelativas que facilitam ao utilizador entender o estado das ações.

Sessão de Esclarecimento - Avaliação ao Aviso N°24-2020-08					
Estado	Início	Fim	Duração	Sigla	Local
● Concluído(a)	23/04/2021	23/04/2021	1.0 horas	SEAA24	Online

Sessão de Esclarecimento - Acerca da Versão 2021					
Estado	Início	Fim	Duração	Sigla	Local
● Concluído(a)	26/03/2021	26/03/2021	2.0 horas	SE-V21	Online

Figura 5.26: Ações de Formação PackWeb antigo

Área de formador > Ações de Formação > Todas					
Todas A iniciar A decorrer Concluída E-Learning					
	Processamento de texto2	Estado - Concluída	Início - 01-08-2021	Fim - 30-08-2021	Duração - 50.0 horas
	Processamento de texto	Estado - E-Learning	Início - 01-07-2021	Fim - 30-07-2021	Duração - 50.0 horas
	Processamento de texto	Estado - Planeada	Início - 01-07-2021	Fim - 30-07-2021	Duração - 50.0 horas

Figura 5.27: Ações de Formação PackWeb novo

Outra grande inovação que ainda está em implementação, é a transformação do menu quando entramos numa formação, que apenas mostrava o menu de lado esquerdo, logo com os conteúdos a lado direito, agora irá ser uma dashboard que permitirá os formandos ou formadores verem todo os tipos de dados que são mais importantes. Como dito antes, esta parte ainda está em implementação, e ainda estamos a estudar quais os campos que deviam ser visualizados aqui.

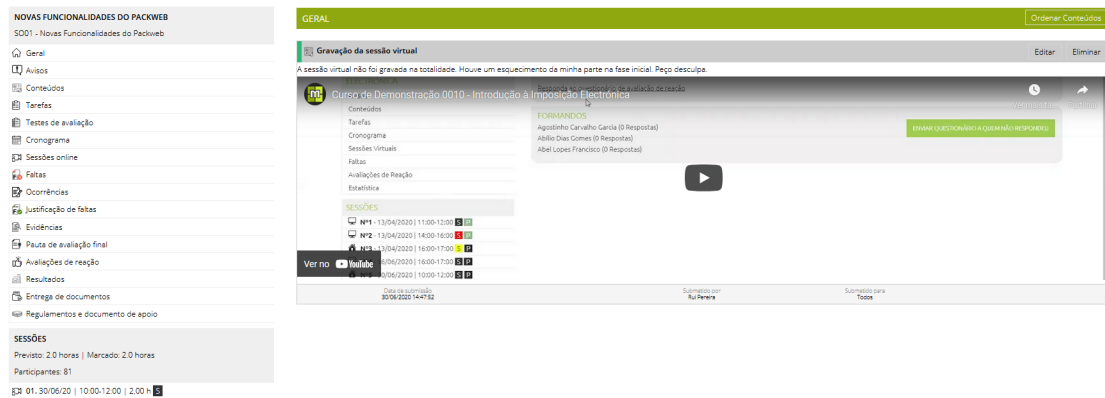


Figura 5.28: Painel inicial de uma formação PackWeb antigo

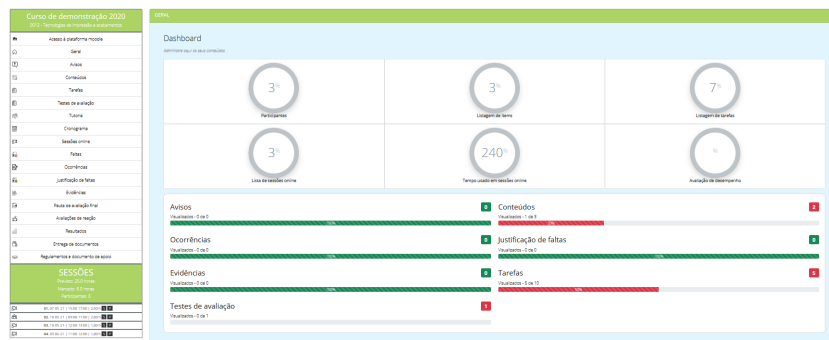


Figura 5.29: Painel inicial de uma formação PackWeb novo

A área pessoal do utilizador levou uma completa mudança, onde antes era apenas um painel de boas vindas, estamos a trabalhar para que seja algo parecido como uma rede social, mais uma vez estes aspetos ainda estão a ser estudados e trabalhados, logo isto é apenas uma fase inicial.

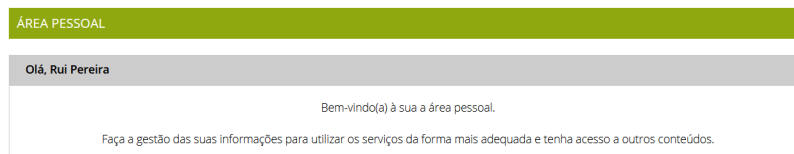


Figura 5.30: Área pessoal PackWeb antigo

Estes são apenas alguns dos exemplos do que foi melhorado no PackWeb da Mestreclique. No final, esta parte ajudou-me bastante a desenvolver as minhas competências em Html/Bootstrap. Nunca tendo mexido em bootstrap foi um pouco difícil para mim no início, mas no final mostrou-se uma solução maravilhosa e de fácil utilização.

5.6 Apresentações em Powerpoint

A produção de uma maneira eficaz de facilitar a vida aos nossos clientes é sempre um bom motivo para desenvolver uma aplicação, e esta é apenas mais uma maneira de facilitar o trabalho dos nossos clientes. Quando um novo cliente entra no universo da MestreClique, um dos muitos modelos que a nossa plataforma dispõe é um PowerPoint para facilitar as apresentações das aulas dos nossos formadores, mas dependendo do formador e do tipo de formação, os textos variam e os logótipos também, fazendo assim com que o formador seja obrigado a colocar esses dados à mão. É aqui que a minha solução é usada.

Quando esse modelo é dado ao cliente ele já foi anteriormente trabalhado por nós, mas o que nós entregamos ao cliente é apenas um modelo "base" para ele conseguir construir conteúdo de maneira idêntica a outro formador da mesma empresa.

Dentro de cada ação a MestreClique dispõe de uma serie de relatórios, os quais vão diretamente buscar os dados à base de dados e são automatizados para que o cliente tenha de fazer apenas o indispensável, e foi com esse pensamento que criamos esta solução também. Assim, o formador quando entra na ação nós já sabemos quem está a tentar retirar a apresentação, qual a ação e o curso, e os logótipos que devemos colocar.

Como disse, o modelo já foi trabalhado por nós antes de ser entregue para a plataforma, para além das mudanças feitas relativamente a cores de cada cliente, é apenas preciso colocar 3 caixas para indicar onde deverão ficar os logotipos e as caixas que forem precisas de texto. Assim, quando o formador tentar tirar o relatório o algoritmo irá verificar quais são as caixas que deve preencher se existirem, e quais os textos que terá de ir buscar à base de dados para alterar.

Para este projeto foi desenvolvido um protótipo que irei deixar em anexo no GitHub completamente funcional.

Ao dispor do formador está também o modelo "base", o qual ele poderá alterar as caixas ou adicionar os textos que deseja aparecer, para isso o formador terá de mexer no modelo global diapositivos do powerpoint, toda a "magia" é feita nesta parte, pois desta forma qualquer coisa que seja feita aqui, irá aparecer em todos os diapositivos do modelo geral que o utilizador fizer como se fosse o "background".

O algoritmo respeitará os limites definidos pelas caixas, e irá inserir as imagens por cima dessas caixas sem ultrapassar os limites ou desfocar a imagem. O texto

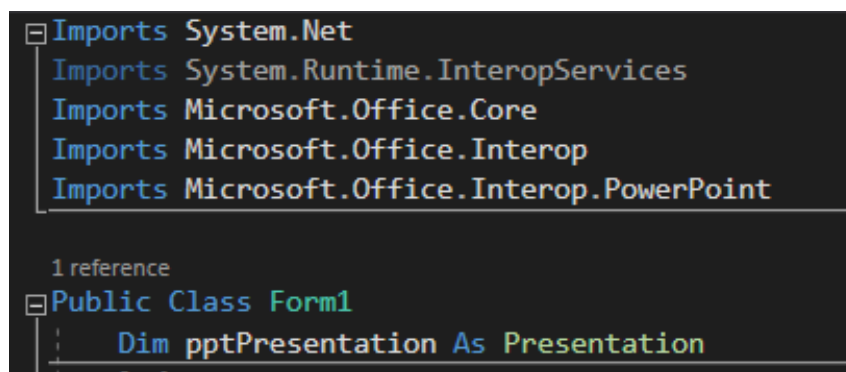
que deve ser alterado deve estar limitado pelos símbolos "«"no início e "»"no fim do texto pretendido, e no meio deve estar escrito a palavra-chave do que é pretendido ir buscar à base de dados. Neste protótipo criamos um texto pré-definido que irá ser substituído por um texto padrão, e damos também ao utilizador opção de escolher imagens que estejam num local específico, ou uma ligação da Internet. As caixas para as imagens têm um ID específico:

"HL- Header Left - cabeçalho superior esquerdo

"HR- Header Right - cabeçalho superior direito

"FC- Footer Bottom - rodapé central

Após vários estudos de bibliotecas disponíveis, acabamos por optar mesmo pelo Microsoft.Office e o Microsoft.Office.Interop. Abaixo está uma imagem que identifica os "Imports"necessários e as declarações de variáveis globais para este projeto.



```
Imports System.Net
Imports System.Runtime.InteropServices
Imports Microsoft.Office.Core
Imports Microsoft.Office.Interop
Imports Microsoft.Office.Interop.PowerPoint

1 reference
Public Class Form1
    Dim pptPresentation As Presentation
```

Figura 5.34: Importes das bibliotecas e inicialização da variável global

Como isto é um protótipo apenas de demonstração, fazemos as ações diretamente no "page_load", assim só temos de correr o programa para que ele faça o algoritmo de introdução das imagens e texto.

Inicializamos uma nova aplicação e criamos uma variável para aceder ao Slide Master ou Modelo global como mencionado anteriormente. Aplicamos um tema ao Modelo Global e chamamos a função para adicionar as imagens, neste caso fazemos apenas três imagens, mas se houver necessidade de adicionar mais, basta criar mais uma linha de texto com o ID e o caminho da imagem pretendida. Chamamos também a função para colocar texto, e finalmente guardamos a nossa apresentação com o nome que desejar. Para mudar o nome que queremos para a nossa nova apresentação, basta indicar no final do caminho, neste caso a apresentação vai chamar-se "Nova Apresentação.pptx"

```

Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
    Dim pptApplication As Application = New Application()
    Dim master As Microsoft.Office.Interop.PowerPoint.Master
    pptPresentation = pptApplication.Presentations.Add(MsoTriState.msoTrue)
    master = pptPresentation.SlideMaster
    master.ApplyTheme("C:\Users\ruico\Documents\GitHub\powerpointcreation\tema.potx")
    colocaimagem("hl", "https://www.mestreclique.pt/img/logo.png")
    colocaimagem("hr", "C:\Users\ruico\Desktop\Mestreclique\Imagens\indice.jpg")
    colocaimagem("fc", "https://www.qualiforma.pt/modelos/fc_2.jpg")
    colocatexto("Texto exemplo 3")
    pptPresentation.SaveAs("C:\Users\ruico\Desktop\Nova Apresentação.pptx", Microsoft.Office.Interop.PowerPoint.PpSaveAsFileType.ppSaveAsDefault, MsoTriState.msoTrue)
    Close()
End Sub

```

Figura 5.35: Page_load

A primeira parte da função "Colocaimagem" irá identificar se a imagem é um caminho ou um link para uma imagem. Antes disso necessitamos de 3 variáveis para utilizar no algoritmo de maneira a encaixar a imagem e não a desfocar.

cimage - caminho da imagem - variável do tipo Double;

aimage - altura da imagem - variável do tipo Double;

image - a própria imagem em si - variável do tipo Object.

```

Sub colocaimagem(id As String, ficheiro As String)
    Dim cimage As Double
    Dim aimage As Double
    Dim imagem As Object
    If (ficheiro.Contains("http")) Then
        Dim request As WebRequest = WebRequest.Create(ficheiro)
        Dim response As WebResponse = request.GetResponse()
        imagem = Image.FromStream(response.GetResponseStream())
        cimage = imagem.Width * 0.75
        aimage = imagem.Height * 0.75
    Else
        imagem = New Bitmap(ficheiro)
        cimage = imagem.Width * 0.75
        aimage = imagem.Height * 0.75
    End If
End Sub

```

Figura 5.36: Primeira parte da função colocaimagem

De seguida, resumidamente, temos o algoritmo que nos irá colocar a imagem na caixa com o "ID" indicado, vai também tirar medidas da caixa e fazer algumas contas para saber qual o tamanho que a imagem deve ter, e a localização que a imagem deve ficar, para que não exceda os limites da caixa e fique no sitio indicado dependendo do "ID" da mesma. Por exemplo, sabemos que se o "ID" da imagem for o "HL" que a imagem deve inicializar a partir do ponto superior esquerdo da caixa, assim como se o "ID" for "HR" sabemos que a imagem para não ficar desfocada e para que não ultrapasse o limite da caixa, deve inicializar a partir do ponto superior direito da caixa menos a largura da imagem, para que desta maneira fique alinhada à direita da caixa. O mais difícil foi criar uma maneira para que a imagem com o "ID" "FC" ficasse centrada no meio da caixa. Para isso o que fazemos é dizer que o ponto de inicialização da imagem deve ser o ponto superior esquerdo, mais metade da largura da caixa (ou seja ficando posicionado no meio da caixa), menos metade da largura da imagem, e dessa maneira conseguimos colocar a imagem mesmo no centro da caixa.


```

Try
    Dim shape1 As PowerPoint.Shape = TryCast(pptPresentation.SlideMaster.Shapes(id), PowerPoint.Shape)
    Dim left1 = shape1.Left
    Dim top1 = shape1.Top
    Dim width1 As Double = shape1.Width
    Dim height1 As Double = shape1.Height
    Dim pcump As Double = width1 / cimage
    Dim palt As Double = height1 / aimage
    Dim pfinal As Double
    If pcump < palt Then
        pfinal = pcump
    Else
        pfinal = palt
    End If
    Dim cfinal As Double = cimage * pfinal
    Dim afinal As Double = aimage * pfinal
    Dim pleft As Double
    Select Case id
        Case "h1"
            pleft = left1
        Case "h2"
            pleft = left1 + width1 - cfinal
        Case "fc"
            pleft = left1 + (width1 / 2.0) - (cfinal / 2.0)
    End Select
    pptPresentation.SlideMaster.Shapes.AddPicture(FileName:=ficheiro, LinkToFile:=MsoTriState.msoTrue, SaveWithDocument:=MsoTriState.msoTrue, pleft, top1, cfinal, afinal)
Catch ex As Exception
End Try
End Sub

```

Figura 5.37: Segunda parte da função colocaimagem

A função "Colocatexto" é a mais simples, verifica se existe o símbolo "«" e substitui esse texto pelo texto indicado na variável. Aqui fazemos esta ação três vezes pois pode haver situações em que o formador pretenda ter o mesmo texto várias vezes.

```

Sub colocatexto(texto As String)
    Dim i As Integer
    Dim shapes As PowerPoint.Shapes = pptPresentation.SlideMaster.Shapes
    'Dim dtcampos As Data.DataTable
    For i = 1 To shapes.Count
        Dim shape As PowerPoint.Shape = TryCast(shapes(i), PowerPoint.Shape)
        If shape IsNot Nothing Then
            Try
                If shape.TextFrame.TextRange.Text.Contains("«") Then
                    'For j = 0 To dtcampos.Columns.Count - 1
                    shape.TextFrame.TextRange.Replace("«texto»", texto)
                    shape.TextFrame.TextRange.Replace("«texto»", texto)
                    shape.TextFrame.TextRange.Replace("«texto»", texto)
                    'Next
                    ' shape.TextFrame.TextRange.Replace("«texto»", texto)
                    ' shape.TextFrame.TextRange.Replace("«texto»", texto)
                End If
            Catch ex As Exception
            End Try
        End If
    Next
End Sub

```

Figura 5.38: Função colocatexto

Eis o resultado final do powerpoint.

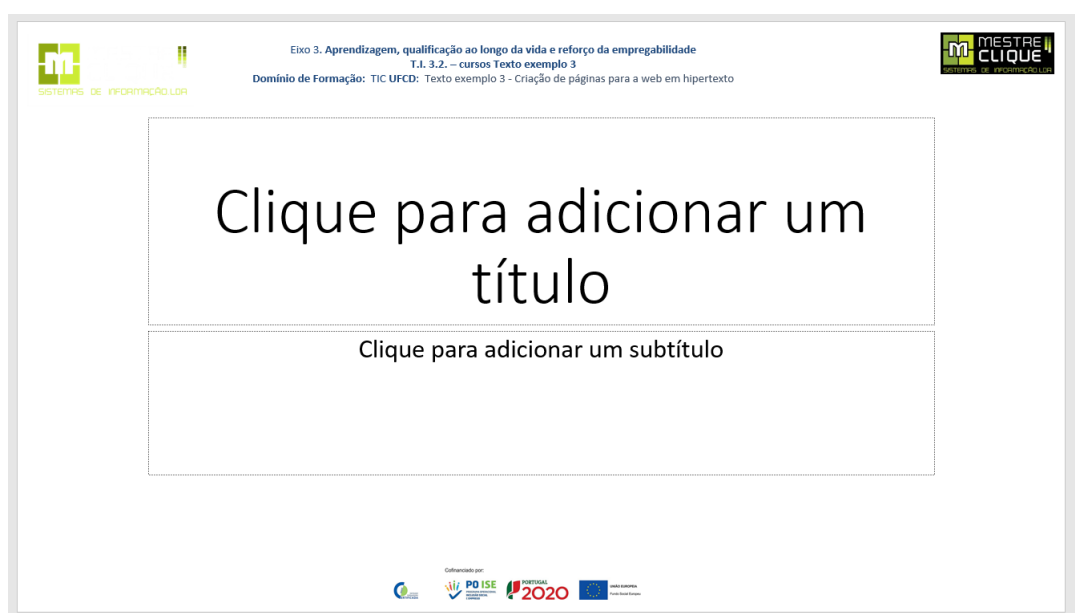


Figura 5.39: Resultado final

5.7 Outras funções desempenhadas

Para além de todos estes desenvolvimentos, trabalhei também com os outros colaboradores da MestreClique. Por vezes enquanto estava parado no desenvolvimento à espera do Supervisor o Eng. Rui Pereira, procurava algo mais para fazer. Por vezes ajudava o colega António Carvalho com alguma reparação de um cliente, ou alguma assistência que fosse necessária. Outras vezes ajudava a colega Fátima Andrade com algumas mudanças que fossem necessárias fazer nos nossos formulários/modelos para utilização dos clientes. Mas a maior parte de tempo livre foi utilizado para atendimento ao público e ajuda na realização de tarefas ou bugs existentes na plataforma MestreClique. Estou agradecido por este estágio não passar apenas por programação, pois com todas estas tarefas extras, sinto que hoje sou um profissional mais completo.

Capítulo 6

Conclusões

Graças a este estágio na MestreClique, consegui melhorar competências já adquiridas durante o ano e meio letivo que tive de aulas, aprendi também muito acerca de outras áreas que não tinha conhecimento, tais como as APIs, a gestão documental, o Bootstrap, entre outros.

Com este estágio concluo que ainda tenho muito para aprender, esta foi uma experiência incrível que me proporcionou não só aprendizagens nas áreas lecionadas mas como amigos para o resto da vida, pessoas que me motivaram para continuar a crescer a nível técnico e humano, este estágio foi muito mais do que apenas um aprimorar de conhecimentos adquiridos durante o ano letivo, acabou por se tornar algo memorável para a minha vida, acordar de manhã e ter motivação e gosto para ir trabalhar, foi um dos principais motivos que me fez mudar de vida e ingressar neste curso, foi com o intuito de um dia ter uma vida melhor que me aventurei por este caminho, e tanto o curso, como esta empresa mostrou-me que realmente é possível isso acontecer, e motivou-me ainda mais para continuar a lutar e a crescer como ser humano e técnico de informática.

Referências

MestreClique - A empresa:

<https://www.mestreclique.pt/ficha/4> - Consultado em Setembro 2021

Tecnologia utilizada - Visual Studio 2019:

https://pt.wikipedia.org/wiki/Microsoft_Visual_Studio - Consultado em Setembro 2021

Tecnologia utilizada - Visual Basic:

https://pt.wikipedia.org/wiki/Visual_Basic - Consultado em Setembro 2021

Tecnologia utilizada - ASP.NET:

<https://pt.wikipedia.org/wiki/ASP.NET> - Consultado em Setembro 2021

Tecnologia utilizada - SQL Manager:

https://pt.wikipedia.org/wiki/SQL_Server_Management_Studio - Consultado em Setembro 2021

Tecnologia utilizada - GitHub:

<https://pt.wikipedia.org/wiki/GitHub> - Consultado em Setembro 2021

Tecnologia utilizada - Bootstrap:

[https://pt.wikipedia.org/wiki/Bootstrap_\(framework_front-end\)](https://pt.wikipedia.org/wiki/Bootstrap_(framework_front-end)) - Consultado em Setembro 2021

Tecnologia utilizada - Recaptcha:

<https://pt.wikipedia.org/wiki/ReCAPTCHA> - Consultado em Setembro 2021

Tecnologia utilizada - Overleaf:

<https://en.wikipedia.org/wiki/Overleaf> - Consultado em Setembro 2021