

**RELATÓRIO DE MODULO 16**

**CURSO TÉCNICO PROFISSIONAL DE GESTÃO E PROGRAMAÇÃO DE SISTEMAS INFORMÁTICOS**

Professores: José Alexandre, Luísa Alves, Joaquim Buinho

André Custódio, L1949

Marco Henriques, L1974

01/06/2021

*O relatório encontra-se em condições para ser apresentado*

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Ciclo de Formação 2018/2021  
Ano Letivo 2020/2021

# Agradecimentos

O nosso grupo tem muito a agradecer aos professores José Alexandre, Joaquim Buinho e Luísa Alves por terem estado sempre ajudar ao longo do projeto.

Gostávamos de agradecer ao professor José Alexandre, por nos ter ajudado na conceção da rede informática do projeto assim como ajuda em vários outros problemas que iam aparecendo

Gostaríamos também de agradecer à professora Luísa Alves pelo gasto do seu tempo pessoal com visa a nos encaminhar no projeto e ajudar na realização do mesmo.

# Índice

[Agradecimentos 1](#_Toc74660684)

[Índice 2](#_Toc74660685)

[Índice de Imagens 4](#_Toc74660686)

[Introdução 6](#_Toc74660687)

[Capítulo I – Cronograma Inicial 7](#_Toc74660688)

[Descrição do Cronograma 8](#_Toc74660689)

[Capítulo II- Conceção do Projeto 9](#_Toc74660690)

[Objetivos do projeto 9](#_Toc74660691)

[Desenvolvimento da rede: 10](#_Toc74660692)

[Equipamentos ativos: 10](#_Toc74660693)

[Tipologia de rede: 10](#_Toc74660694)

[Packet tracer (simulação da rede) 12](#_Toc74660695)

[Desenvolvimento do Website ASP.NET 13](#_Toc74660696)

[Encriptação RSA 13](#_Toc74660697)

[Registo do cliente 14](#_Toc74660698)

[Login do Cliente 17](#_Toc74660699)

[Ver/Editar perfil 19](#_Toc74660700)

[Visualizar as encomendas 21](#_Toc74660701)

[Navegação no Website 22](#_Toc74660702)

[Carrinho 24](#_Toc74660703)

[Query da Base de Dados 28](#_Toc74660704)

[Desenvolvimento da aplicação do Admin 31](#_Toc74660705)

[Tecnologias 31](#_Toc74660706)

[Frameworks 32](#_Toc74660707)

[.NET Framework 32](#_Toc74660708)

[Vantagens de usar um Framework: 32](#_Toc74660709)

[Linguagem de Programação 33](#_Toc74660710)

[C# 33](#_Toc74660711)

[Sql 33](#_Toc74660712)

[*IDE* 33](#_Toc74660713)

[Json 34](#_Toc74660714)

[O Programa 34](#_Toc74660715)

[Abreviaturas: 34](#_Toc74660716)

[CamelCase 34](#_Toc74660717)

[Linq 36](#_Toc74660718)

[*Class* 36](#_Toc74660719)

[Ligações a base de dados 37](#_Toc74660720)

[Try, catch finally 37](#_Toc74660721)

[Notificações 37](#_Toc74660722)

[Stored Procedure 38](#_Toc74660723)

[Auto-Run 38](#_Toc74660724)

[Projeto 40](#_Toc74660725)

[Capítulo IV – Cronograma Final e Justificação de desvios 47](#_Toc74660726)

[Justificação 48](#_Toc74660727)

[Conclusão 49](#_Toc74660728)

[Bibliografia e Web Grafia: 50](#_Toc74660729)

# Índice de Imagens

[Figura 1 - Cronograma Inicial 7](#_Toc74660730)

[Figura 2 - Topologia em estrela 10](#_Toc74660731)

[Figura 3 - Tabela de IPs de Sta. Maria da Feira 11](#_Toc74660732)

[Figura 4 - Tabela de IPs de Setúbal 11](#_Toc74660733)

[Figura 5 - Rede da delegação de Sta. Maria da Feira 12](#_Toc74660734)

[Figura 6 - Rede da delegação de Setúbal 12](#_Toc74660735)

[Figura 7 - Encriptação da password 13](#_Toc74660736)

[Figura 8 - Desencriptação da password 13](#_Toc74660737)

[Figura 9 - Interface para o utilizador se registar 14](#_Toc74660738)

[Figura 10 - Verificar se o username já existe 15](#_Toc74660739)

[Figura 11 - Verificar se as senhas inseridas coincidem e envio do Username e Password 15](#_Toc74660740)

[Figura 12 - Mensagem de passwords diferentes 15](#_Toc74660741)

[Figura 13 - Código que mostra o cliente a ser inserido com sucesso 16](#_Toc74660742)

[Figura 14 - Icon de login 17](#_Toc74660743)

[Figura 15 - Modal do login 17](#_Toc74660744)

[Figura 16 - Código do Login 18](#_Toc74660745)

[Figura 17 - Mensagem de erro 18](#_Toc74660746)

[Figura 18 - Mensagem Sucesso no login 18](#_Toc74660747)

[Figura 19 - Icon para editar perfil 19](#_Toc74660748)

[Figura 20 - Informações pessoais do cliente 19](#_Toc74660749)

[Figura 21 - Código do select da base de dados dos campos pretendidos e afixação dos mesmos 19](#_Toc74660750)

[Figura 22 - Menu para editar as informações pessoais 20](#_Toc74660751)

[Figura 23 - Update das informações pessoais 20](#_Toc74660752)

[Figura 24 - Icon visualizar encomenda 21](#_Toc74660753)

[Figura 25 - GridView das encomendas feitas pelo utilizador 21](#_Toc74660754)

[Figura 26 - Redes sociais da empresa e definições, encomendas e carinho de compras 22](#_Toc74660755)

[Figura 27 - Produto Masculino 22](#_Toc74660756)

[Figura 28 - Código do menu de pesquisa do tipo de produto 22](#_Toc74660757)

[Figura 29 - Seleção do tipo de produto da DB e conversão da imagem do produto 23](#_Toc74660758)

[Figura 30 - Display do item escolhido para comprar 24](#_Toc74660759)

[Figura 31 - Código dos detalhes do produto 24](#_Toc74660760)

[Figura 32 - Insert dos itens no carrinho de compras 25](#_Toc74660761)

[Figura 33 - Carrinho de compras com itens inseridos 25](#_Toc74660762)

[Figura 34 - Select dos dados dos produtos que estão no carrinho assim como o IdCliente 26](#_Toc74660763)

[Figura 35 - Insert dos produtos na tbl\_Encomendas 26](#_Toc74660764)

[Figura 36 - Delete dos itens da listbox 27](#_Toc74660765)

[Figura 37 - .*Net* *Framework* 31](file:///C:\Users\L1949\Documents\GitHub\Project-Mod16-C-Sharp\Relatório%20M16.docx#_Toc74660766)

[Figura 38 - Linguagens de Programação 31](file:///C:\Users\L1949\Documents\GitHub\Project-Mod16-C-Sharp\Relatório%20M16.docx#_Toc74660767)

[Figura 39 - Inicio do .*Net Framework* 32](file:///C:\Users\L1949\Documents\GitHub\Project-Mod16-C-Sharp\Relatório%20M16.docx#_Toc74660768)

[Figura 40 - *C#* 33](file:///C:\Users\L1949\Documents\GitHub\Project-Mod16-C-Sharp\Relatório%20M16.docx#_Toc74660769)

[Figura 41 - *Sql* 33](file:///C:\Users\L1949\Documents\GitHub\Project-Mod16-C-Sharp\Relatório%20M16.docx#_Toc74660770)

[Figura 42 - Json 34](file:///C:\Users\L1949\Documents\GitHub\Project-Mod16-C-Sharp\Relatório%20M16.docx#_Toc74660771)

[Figura 43 - camelCase 34](file:///C:\Users\L1949\Documents\GitHub\Project-Mod16-C-Sharp\Relatório%20M16.docx#_Toc74660772)

[Figura 44 - *Linq* 36](file:///C:\Users\L1949\Documents\GitHub\Project-Mod16-C-Sharp\Relatório%20M16.docx#_Toc74660773)

[Figura 45 - *Class* Cliente 36](file:///C:\Users\L1949\Documents\GitHub\Project-Mod16-C-Sharp\Relatório%20M16.docx#_Toc74660774)

[Figura 46 - Enviar os dados para a *class* 36](file:///C:\Users\L1949\Documents\GitHub\Project-Mod16-C-Sharp\Relatório%20M16.docx#_Toc74660775)

[Figura 47 - Ligação a base de dados 37](#_Toc74660776)

[Figura 48 - comando para mostrar o balão das notificações 37](file:///C:\Users\L1949\Documents\GitHub\Project-Mod16-C-Sharp\Relatório%20M16.docx#_Toc74660777)

[Figura 49 - Balão de Notificações 37](file:///C:\Users\L1949\Documents\GitHub\Project-Mod16-C-Sharp\Relatório%20M16.docx#_Toc74660778)

[Figura 50 - C# mais Stored Procedure 38](#_Toc74660779)

[Figura 51 - Stored Procedure Insert Encomendas 38](file:///C:\Users\L1949\Documents\GitHub\Project-Mod16-C-Sharp\Relatório%20M16.docx#_Toc74660780)

[Figura 52 - Codigo do Auto Run 38](file:///C:\Users\L1949\Documents\GitHub\Project-Mod16-C-Sharp\Relatório%20M16.docx#_Toc74660781)

[Figura 53 - Form quando iniciamos sessão 40](file:///C:\Users\L1949\Documents\GitHub\Project-Mod16-C-Sharp\Relatório%20M16.docx#_Toc74660782)

[Figura 54 - Form dos Administradores 40](file:///C:\Users\L1949\Documents\GitHub\Project-Mod16-C-Sharp\Relatório%20M16.docx#_Toc74660783)

[Figura 55 - Form Adicionar Produtos 41](file:///C:\Users\L1949\Documents\GitHub\Project-Mod16-C-Sharp\Relatório%20M16.docx#_Toc74660784)

[Figura 56 - Formulario de Editar ou Apagar Produtos 41](file:///C:\Users\L1949\Documents\GitHub\Project-Mod16-C-Sharp\Relatório%20M16.docx#_Toc74660785)

[Figura 57 - Quando clicamos no Clientes ou Funcionarios 42](file:///C:\Users\L1949\Documents\GitHub\Project-Mod16-C-Sharp\Relatório%20M16.docx#_Toc74660786)

[Figura 58 - Inicio do Form Cliente e Funcionario 42](file:///C:\Users\L1949\Documents\GitHub\Project-Mod16-C-Sharp\Relatório%20M16.docx#_Toc74660787)

[Figura 59 - Modificar os Dados do Cliente e Funcionario 43](file:///C:\Users\L1949\Documents\GitHub\Project-Mod16-C-Sharp\Relatório%20M16.docx#_Toc74660788)

[Figura 60 - Form Consulta Carrinho 44](file:///C:\Users\L1949\Documents\GitHub\Project-Mod16-C-Sharp\Relatório%20M16.docx#_Toc74660789)

[Figura 61 - Detalhes de Produtos 44](file:///C:\Users\L1949\Documents\GitHub\Project-Mod16-C-Sharp\Relatório%20M16.docx#_Toc74660790)

# Introdução

O projeto do módulo 16, consiste na criação de uma rede informática e um website para o cliente poder aceder aos produtos e poder encomendar, para o Sr. José RoupaJusta da Silva, proprietário da empresa VesteBem Lda.

Este relatório tem como objetivo mostrar o nosso percurso para desenvolver o projeto assim como demonstrar todas as aprendizagens adquiridas neste projeto.

Iremos demonstrar todas as atividades desenvolvidas, os apoios teóricos a que recorremos e explicar como foi desenvolvido o nosso projeto, identificando todas as ferramentas utilizadas ao longo do seu desenvolvimento

# Capítulo I – Cronograma Inicial

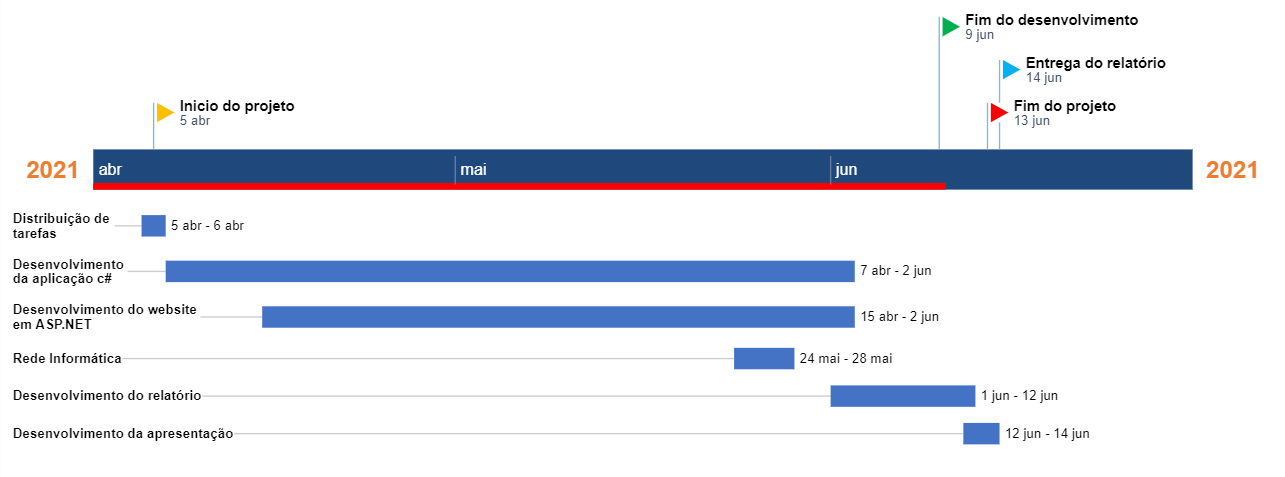


Figura - Cronograma Inicial

## Descrição do Cronograma

No cronograma mostrado acima mostra a nossa expectativa de realização de todas as etapas do projeto até ao dia de entrega do projeto, o projeto começou no dia 05/04/2021 e a data de conclusão irá ser dia 13/06/2021.

Se tudo correr como esperado, no dia 02/06/2021 iremos ter a aplicação e o website já prontos, e a rede informática do projeto já feita também no dia 28/05/2021 faltando terminar o relatório e a apresentação.

# Capítulo II- Conceção do Projeto

## Objetivos do projeto

**Projeto:** VesteBem\_Admin e VesteBem\_Site

**Descrição do Projeto:** O Website tem como objetivo o cliente poder conseguir consultar e encomendar os produtos da empresa Veste Bem. Para o cliente poder efetuar a encomenda terá de fazer login.

A aplicação do administrador, tem como função a manutenção de dados, inserir dados e apagar os mesmos. Para aceder ao aplicativo tem que fazer login primeiro no site e só assim consegue fazer login no aplicativo.

Objetivos:

* Ser acessível para todos os clientes*;*
* Fácil de utilizar;
* Compatível com a plataforma *Windows*;
* Segurança;

## Desenvolvimento da rede:

### Equipamentos ativos:

* 1 servidor DHCP(genérico)
* 2 Routers 5505, um em Setúbal e outro em Sta. Maria da Feira;
* 4 Switches 2960-24TT em Setúbal
* 6 Switches 2960-24TT em Lisboa;

### Tipologia de rede:

A tipologia de rede que o nosso grupo escolheu adotar foi a tipologia em estrela, nós escolhemos usar esta tipologia porque já a tínhamos usado num projeto anterior e tínhamos visto que era a mais segura e a mais utilizada.

A tipologia de rede em estrela faz com que toda a informação deva passar pelo switch, e graças a isso vai tornar a receção de informação mais eficiente, fazendo com que não haja o encaminhamento de informações para todos os utilizadores e sim só para o devido destinatário.

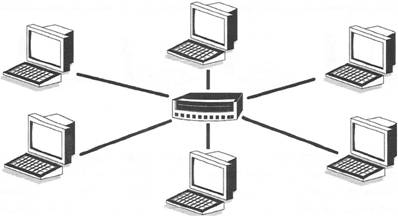


Figura - Topologia em estrela

IPs (DHCP)

O nosso servidor de DHCP tem o Ip: 192.168.1.1 e fornece para Sta. Maria da Feira os IPs para as máquinas entre 192.168.2.1 e 192.168.2.254 enquanto em Setúbal disponibiliza entre 192.168.1.1 e 192.168.1.254. A Subnet mask utilizada nos dois departamentos é 255.255.255.0.

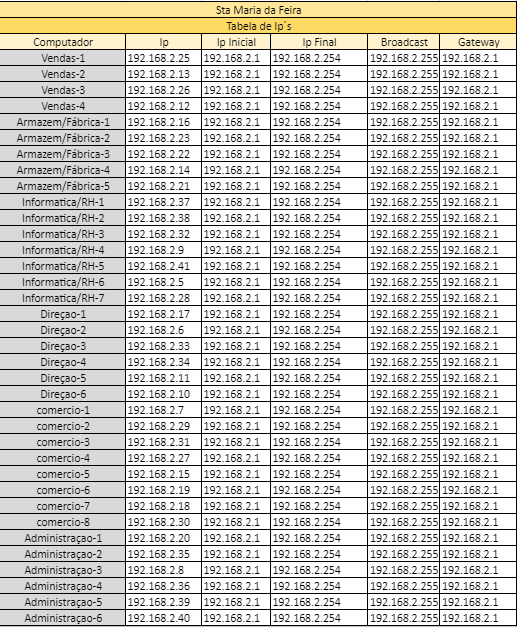


Figura - Tabela de IPs de Sta. Maria da Feira

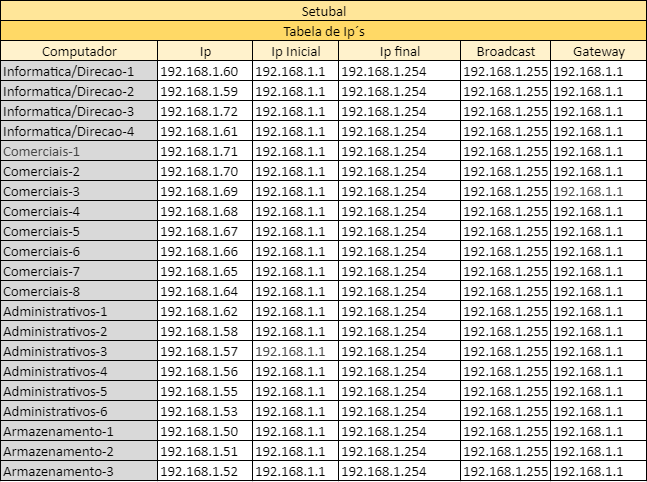


Figura - Tabela de IPs de Setúbal

### Packet tracer (simulação da rede)

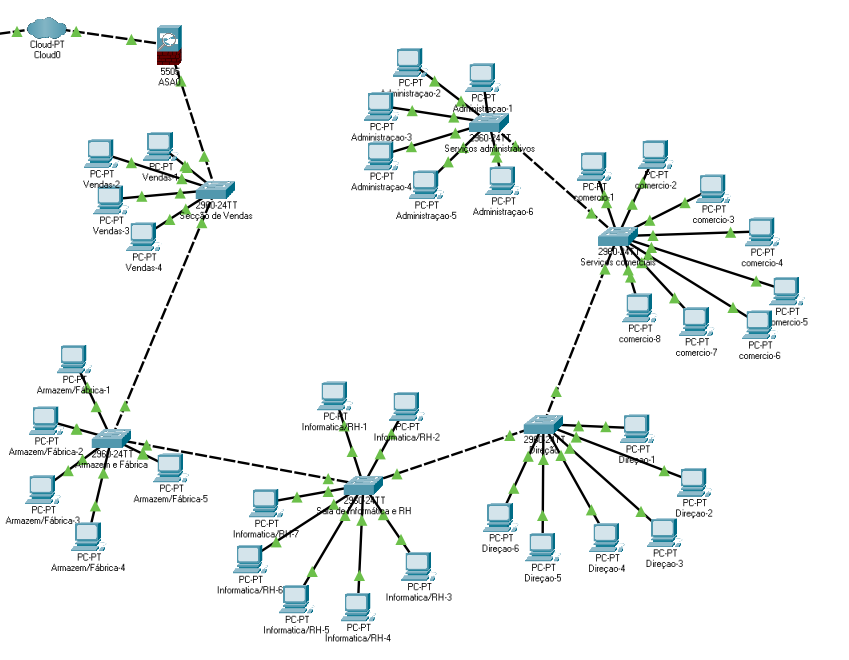


Figura - Rede da delegação de Sta. Maria da Feira

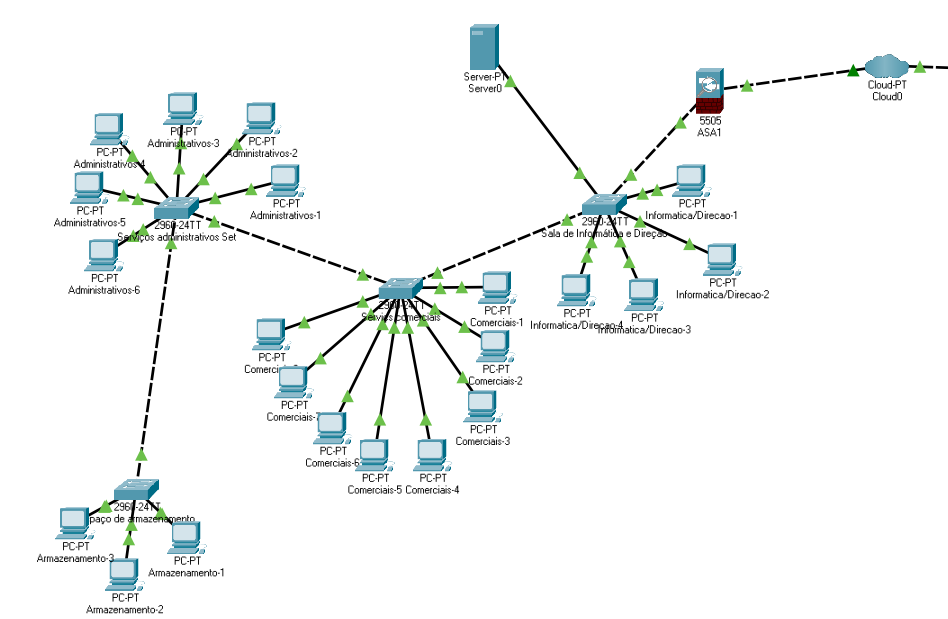


Figura - Rede da delegação de Setúbal

## Desenvolvimento do Website ASP.NET

### Encriptação RSA

A encriptação RSA foi usada no projeto para encriptar as passwords do utilizador fazendo assim com que o registo e o login do utilizador seja mais seguro, o resultado gerado por cada encriptação de cada password vai ser sempre diferente mesmo que a senha seja igual, ou seja, se a password do utilizador “TesteA” for “123” e a password do utilizador “TesteB” for “123” a encriptação gerada para as suas passwords vai ser diferente mesmo sendo elas iguais.

Uma imagem com texto

Descrição gerada automaticamente

Figura - Encriptação da password

#### Uma imagem com texto Descrição gerada automaticamente

Figura - Desencriptação da password

### Registo do cliente

Para os cliente poderem comprar roupa no nosso website os clientes precisam se registar, sem se registarem os clientes conseguem ver os produtos disponiveis na loja através do menu de navegação, mas para poder fazer uma encomenda precisam fazer um registo e um login.

Na nossa página de registo o utilizador tem que nos fornecer todos o seus dados pessoais como o seu nome, data de nascimento, sexo, nif, morada, código postal, localidade, email, telefone, username e a sua password tendo um outro campo para confirmar a password inserida para evitar que o utilizador insira uma password com erros.

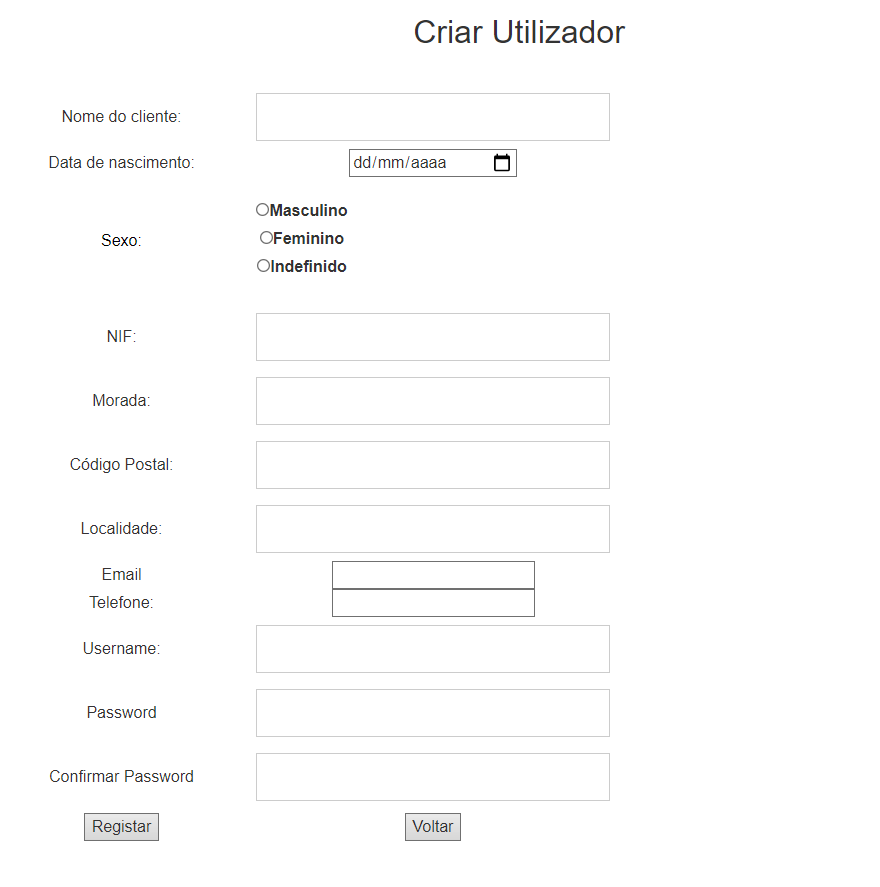


Figura - Interface para o utilizador se registar

.

Uma imagem com texto

Descrição gerada automaticamente

Figura - Verificar se o username já existe

**Nota**: Depois de inserir o username nós decidimos executar o código acima para verificar se o username já existia na base dados, onde se o valor do Count fosse “0” mostraria ao cliente a mensagem “ O username já existe!!”, e se o Count fosse “1” prosseguiria então o registo.



Figura - Verificar se as senhas inseridas coincidem e envio do Username e Password



Figura - Mensagem de passwords diferentes

**Nota**: Nas 2 imagens acima é-nos mostrado como verificamos se as duas passwords são iguais, assim como também o envio dos dados para a base dados.

Uma imagem com texto

Descrição gerada automaticamente

Figura - Código que mostra o cliente a ser inserido com sucesso

**Nota**: Nesta imagem é nos mostrado então depois das duas verificações já faladas, o envio dos dados do utilizador para a base de dados, assim como a encriptação do username, se algum erro inesperado acontecer enquanto o utilizador está a criar conta e que não tenham esses erros a ver com a password e o username, usámos então um catch(Exceprion er) o que esta função faz é escrever na “lblMensagem.text” o tipo de erro que ocorreu na criação do utilizador.

### Login do Cliente

Para o login do utilizador nós decidimos utilizar um modal que fosse mais user friendly, o modal então pede o username, a sua password, tem um botão para cancelar a operação e tem também a opção para criar uma conta nova, onde depois de clicado vai dar ao registo de utilizador.

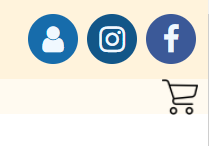


Figura - Icon de login

**Nota**: Para o utilizador aceder ao modal do login precisa então de clicar no icon que é representado na imagem em cima com uma pessoa.

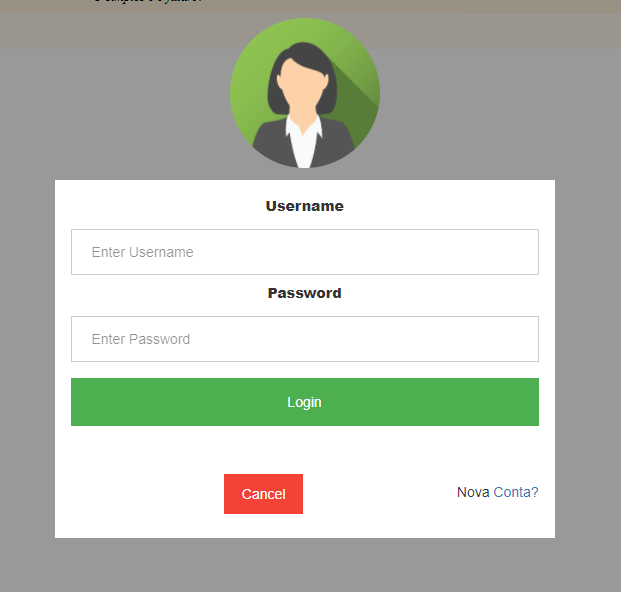


Figura - Modal do login

**Nota**: Depois do utilizador clicar no icon vai dar pop-up deste modal onde o proprio poderá fazer o login inserindo os seus dados .

Uma imagem com texto

Descrição gerada automaticamente

Figura - Código do Login

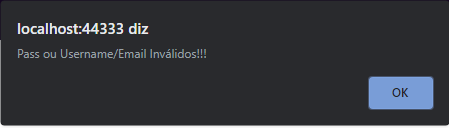


Figura - Mensagem de erro

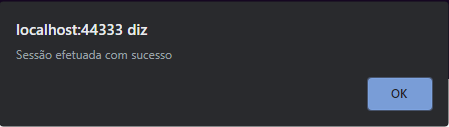


Figura - Mensagem Sucesso no login

**Nota**: Quando o cliente clicar no botão login o programa retira o valor que está na textbox do username e verifica se esse username existe na base de dados, seguidamente se o username existir vai então descodificar a password e vai verificar se a password inserida coincide com a password que está guardada para aquele utilizador na base de dados, se a password não coincidir com a que está guardada na base de dados vai então aparecer uma mensagem de erro a dizer que a password está errada.

## Ver/Editar perfil



Figura - Icon para editar perfil

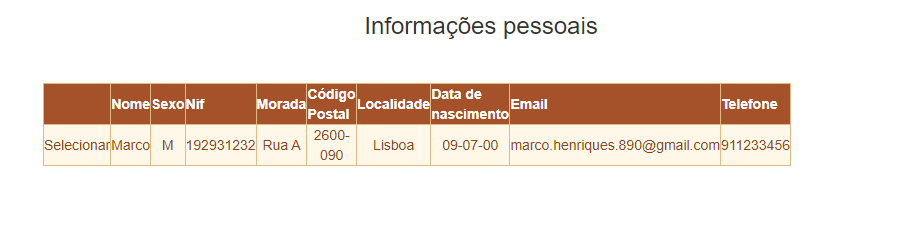


Figura - Informações pessoais do cliente

**Nota**: Quando o cliente clicar no botão de editar perfil a gridview vai buscar a session[Idcliente] e vai afixar os seus dados pessoais na gridview.



Figura - Código do select da base de dados dos campos pretendidos e afixação dos mesmos



Figura - Menu para editar as informações pessoais

**Nota**: Depois de clicar no selecionar vai aparecer este menu onde o cliente irá poder mudar todas as suas informações pessoais com a excessão da sua data de nascimento, o cliente também irá poder voltar para trás clicando no botão voltar, botão este que o encaminhará para a homepage do website.

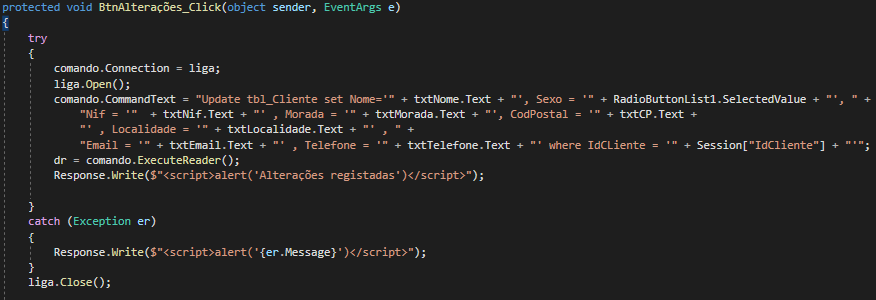


Figura - Update das informações pessoais

**Nota**: Com este comando vai ser dado o update das informações pessoais que o cliente escolheu alterar.

## Visualizar as encomendas



Figura - Icon visualizar encomenda

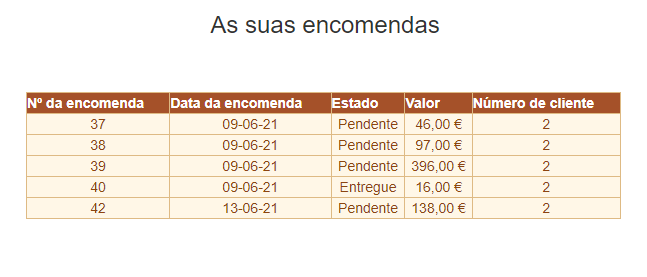


Figura - GridView das encomendas feitas pelo utilizador

**Nota**: Para que o utiliador veja as suas encomendas efetuada ele terá que clicar no icon da caixa, seguidamente irá então aparecer uma gridview com as informações das suas encoemndas efetuadas.

### Navegação no Website

Depois do cliente então fazer o login ele terá a posibilidade de navegar no website e de fazer compras, para fazer isso tem duas categorias que é homem e mulher, depois de carregar num dos sexos vai ser então redirecionado para a pagina de items correspondente ao sexo escolhido pelo utilizador.



Figura - Redes sociais da empresa e definições, encomendas e carinho de compras



Figura - Produto Masculino

**Nota**: Quanto o utilizador clicar na secção desejada, irá então ser mostrado a roupa que está nessa categoria, no exemplo da imagem em cima nos casacos.

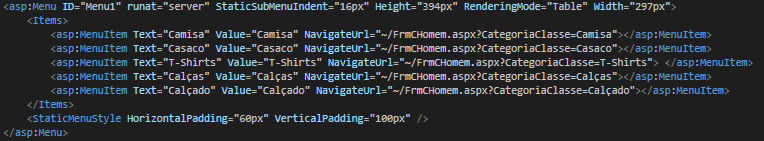


Figura - Código do menu de pesquisa do tipo de produto

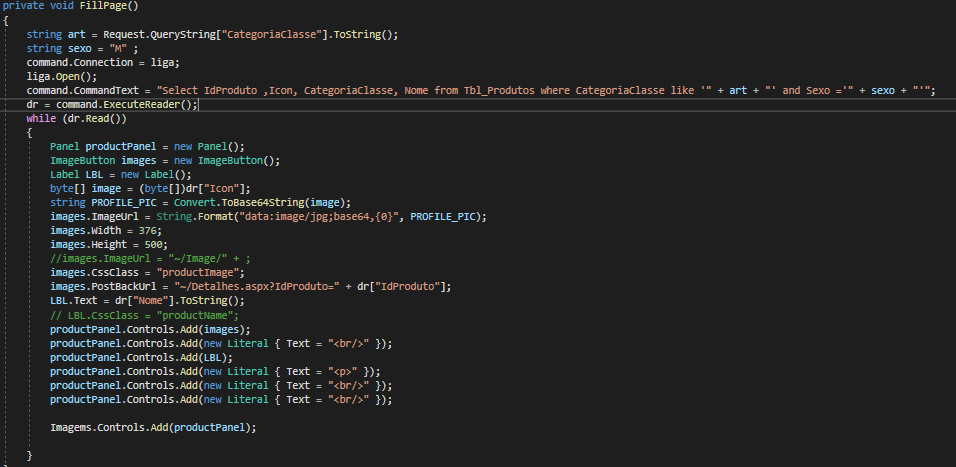


Figura - Seleção do tipo de produto da DB e conversão da imagem do produto

**Nota**: Quando o utilizador escolher um produto que tem definido uma “CategoriaClasse” é enviado esse valor à base de dados e a base de dados retorna os valores que foram escolhidos pelo utilizador e então é mostrada a roupa pretendida na página. (Página do sexo feminino feito da mesma maneira)

### Carrinho

Quando um utilizador clicar num item ele será enviado para o form Detalhes que vai então possibilitar a compra do mesmo assim como examinar o item melhor.

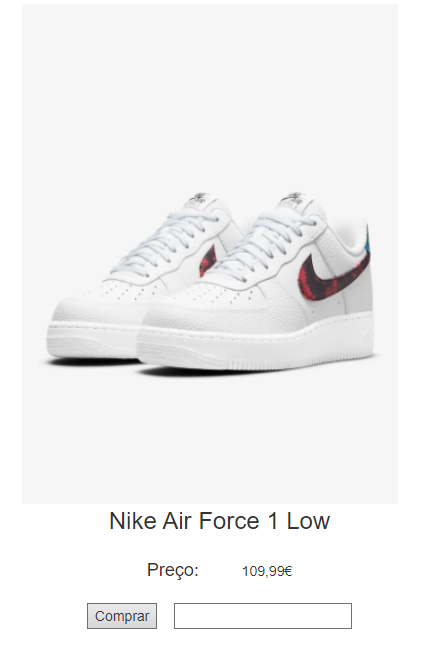


Figura - Display do item escolhido para comprar

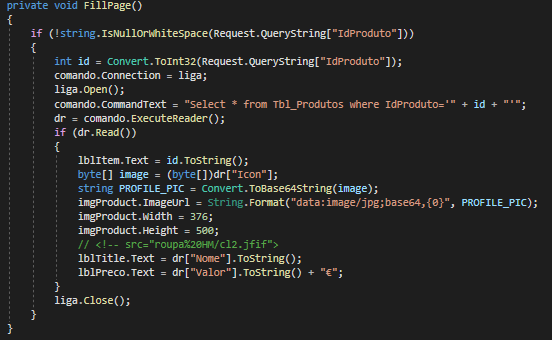


Figura - Código dos detalhes do produto

**Nota**: Este é código por trás do display do item no from Detalhes que vai fazer com que sejam afixadas as informações do produto desejado

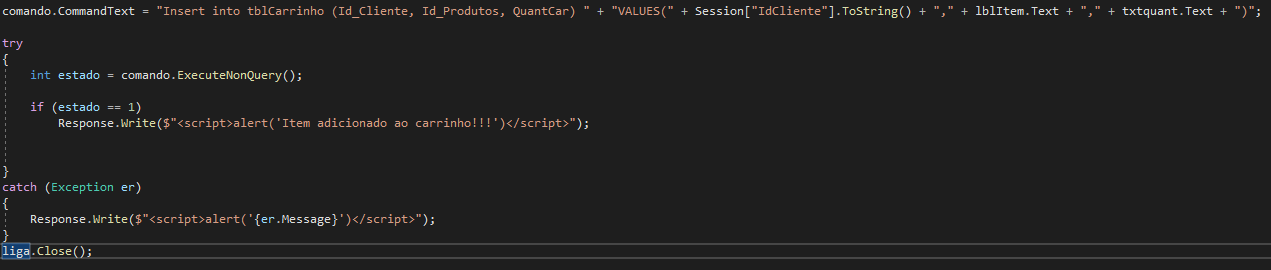


Figura - Insert dos itens no carrinho de compras

**Nota**: Quando é cliado o botão comprar, este código é executado e insere os dados do cliente e do item inserido na tblCarrinho..



Figura - Carrinho de compras com itens inseridos

**Nota**: Nesta página vão então aparecer todos os produtos inseridos na tblCarrinho com o idCliente do utilizador que está logado no website, nesta pa´gina tem também a opção de continuar as compras, apagaros items do carrinho de compras e finalizar a encomenda, sendo a morada de entrega aquela que o cliente inseriu no registo.

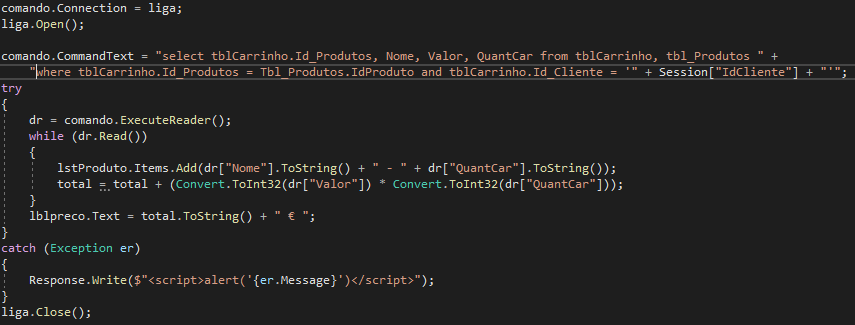


Figura - Select dos dados dos produtos que estão no carrinho assim como o IdCliente

**Nota**: É graças a este código que aparece os items no carrinho pois ele dá um select aos items que foram postos no carrinho por um determinado IdCliente e depois vai buscar à DB os valores da tblCarinho que correspondem a esse IdCliente e afixa essas valores na listbox.

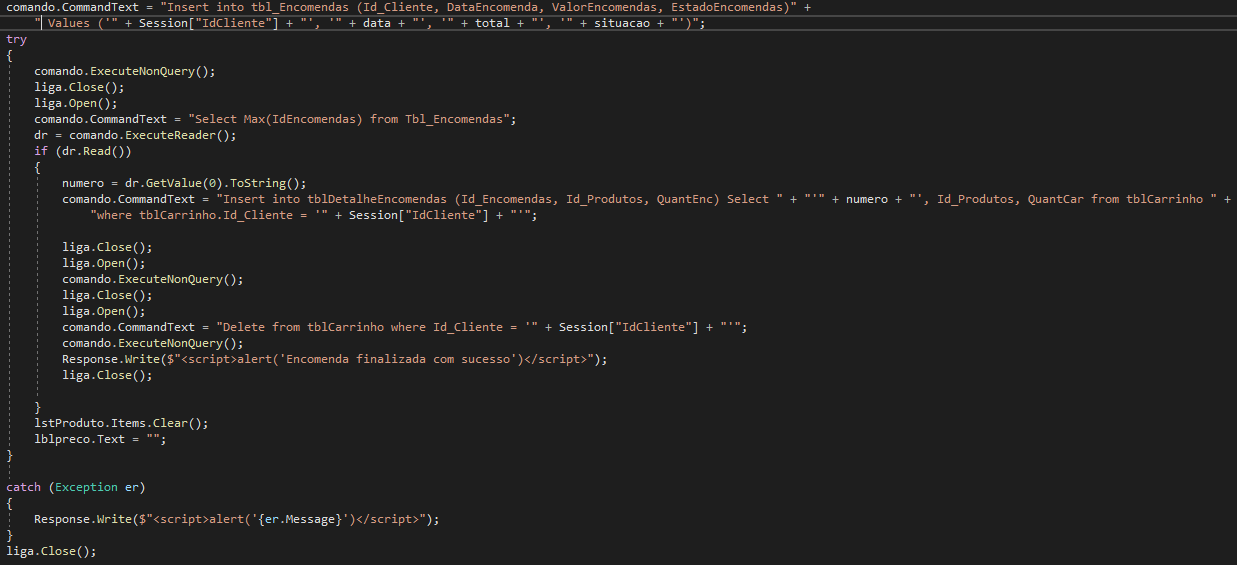


Figura - Insert dos produtos na tbl\_Encomendas

**Nota**: Com este código são etão enviados os valores da listbox para a tabela tbl\_Encomendas, e são tambem inseridos os detalhes da encomenda estando assim então a encomenda concluida.

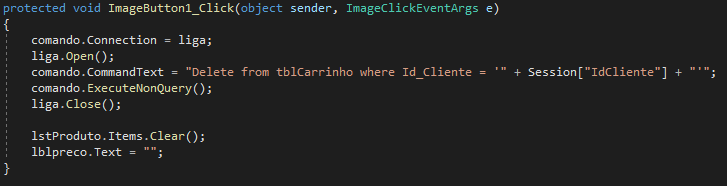


Figura - Delete dos itens da listbox

**Nota**: Este código é usado para apagar os items que estão no carrinho.

## Query da Base de Dados

Create table tbl\_login(

IdLogin int identity(1,1) primary key,

Usern varchar(max) not null,

Passw varchar(max) not null,

Funcionario varchar(1) not null

);

Create table tblFuncao(

IdFuncao int identity(1,1) primary key,

Funcao varchar(100), --Nome da funcao

);

Create table tbl\_Funcionario(

IdFuncionario int identity(1,1) primary key,

Id\_Funcao int Foreign key references tblFuncao(idFuncao),

Nome varchar(100) not null,

Telemovel varchar(9) not null,

Id\_Login int Foreign key references tbl\_login(idLogin)

);

Create table tblEstado(

IdEstado int identity(1,1) primary key,

Estado varchar(100) not null, --Em loja, Nos ctts, ...

);

Create table tbl\_Cliente(

IdCliente int identity(1,1) primary key,

Nome varchar(100) not null,

Sexo varchar(1) not null,

Nif varchar(9) not null,

Id\_Login int foreign key references tbl\_login(idLogin) not null,

Morada varchar(250) not null,

CodPostal varchar(8) not null,

Localidade varchar(100) not null,

DataNasc date not null,

Email varchar(300) not null,

Telefone varchar(9) not null,

Icon Image

);

Create table tbl\_Produtos(

IdProduto int identity(1,1) primary key,

Nome varchar(100) not null,

Valor Decimal(7,2) not null,

NomedaEmpresa varchar(100) not null,

CategoriaClasse varchar(100), -- Setor Cal�as, Sapatos, Acess�rios

CategoriaSubClasse varchar(100), -- Tenis, botas, roupa desportiva

Sexo varchar(1) not null,

Icon Image not null

);

Create table tbl\_Encomendas(

IdEncomendas int identity(1,1) primary key,

ValorEncomendas Decimal(7,2) not null,

EstadoEncomendas int foreign key references tblEstado(IdEstado),

DataEncomenda date not null,

Id\_Cliente int foreign key references tbl\_Cliente(idCliente)

);

Create table tblDetalheEncomendas(

Id\_Encomendas int foreign key references tbl\_Encomendas(IdEncomendas),

Id\_Produtos int foreign key references tbl\_Produtos(IdProduto),

QuantEnc int not null,

Primary key(Id\_Encomendas, Id\_Produtos)

);

Create table tblCarrinho(

IdCarrinho int identity(1,1) primary key,

Id\_Cliente int foreign key references tbl\_Cliente(idCliente),

Id\_Produtos int foreign key references tbl\_Produtos(IdProduto),

QuantCar int not null,

);

## Desenvolvimento da aplicação do Admin

Este aplicativo tem como objetivo o funcionário, fazer atualização dos dados, eliminar os mesmos ou inserir. O mesmo tem um aspeto visual simples e fácil de ser utilizado

### Tecnologias

O nosso projeto do Admin utilizamos uma frameworks (**.Net Framework**) e duas linguagem de programação (**C# e Sql**).

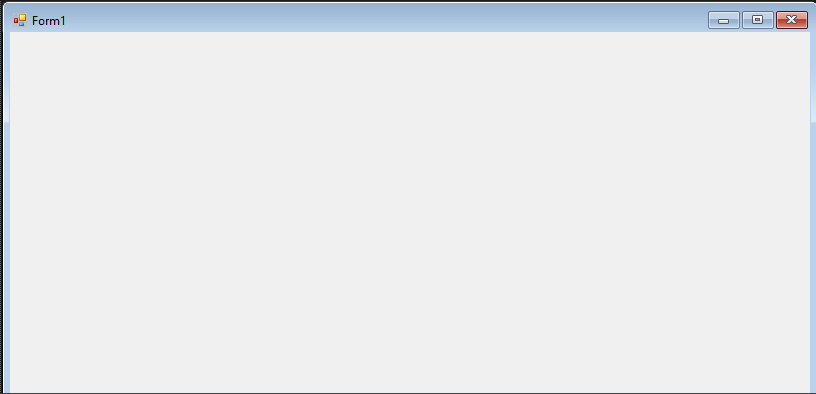
Figura - .*Net* *Framework*



Figura - Linguagens de Programação

## Frameworks

Um framework consiste numa abstração que une códigos entre vários projetos de software, fornecendo uma funcionalidade sem pormenores. Trata-se de um conjunto de classes que colaboram para realizar uma responsabilidade para um domínio de um subsistema da aplicação.

Ao contrário das bibliotecas é o framework quem dita o controlo da aplicação.

### .NET Framework

Figura - Inicio do .*Net Framework*

*.NET Framework* como o nome diz é um *Framework* *open-source*, desenvolvida pela *Microsoft*. Qualquer código .*NET* pode ser executada em todos os dispositivos que tenham um *Framework* da plataforma. A sua versão mais atual e estável é o *4.8* lançada em 04/18/2019.

### Vantagens de usar um Framework:

Uma das grandes vantagens de um *framework* é a padronização (auxiliar na maximização da compatibilidade, reprodutibilidade, segurança e qualidade) do desenvolvimento. Por termos um conjunto já definido de classes e/ou funções, somos “forçados” a trabalhar conforme a ferramenta de escolha.

## Linguagem de Programação

### Resultado de imagem para c#C#

Foi criada pela *Microsoft* em 2001. É simples, moderna, orientada por objetos (conceito de “objetos”, que podem conter dados na forma de campos) e flexível para empreendedores modernos, pois é um recurso para criar *software* que não só funcionará hoje, mas também será aplicável futuramente.

Figura - *C#*

### Sql

Em 12 de junho de 1988, a microsoft uniu-se com o Ashton-Tate e Sybase para criar um software que iria guardar os dados todos digitalmente, pois antes disso os dados eram todos guardados em papel.

Figura - *Sql*

Edgar Frank Codd desenvolveu o modelo relacional, que consiste na maneira como os dados são guardados e que teriam ligação entre elas próprias.

Mais tarde a Sybase e a Microsoft separaram-se e cada uma criou o seu negocio e os seus modelos O Sql tenta fazer a redundância de dados (evitar erros, poupar espaço de memória, evitar perda de tempo e facilitar na pesquisa de dados) e é case sensitivity, aceita tanto maiúsculas tanto minúsculas no código.

## *IDE*

Integrated Development Environment ou Ambiente de Desenvolvimento Integrado é uma software que auxilia os desenvolvedores a desenvolverem o programa. Normalmente os IDEs facilitam a técnica de RAD (Rapid Application Development, Desenvolvimento Rápido de Aplicações) a criar um debug muito rapidamente. Com o uso da IDE é mais fácil de encontrar o erro e de o corrigir, sem ter de ler o código todo, uma grande vantagem é que os IDEs são orientados a objetos, completa alguns linhas de código, tem texto de atalho (como o *cw*, que fica *Console*.*Writeline*), podemos importar alguns dlls publicas para o nosso projeto, etc.

No nosso projeto usamos o Microsoft SQL Management Studio e Visual Studio 2019.

### Json

*Json* ou *Java Script Object Notation* é uma extensão de ficheiro, neste caso *\*.json*. Temos de o instalar no “*newtonsoft.com/json*”, para conseguirmos escrever e ler objetos. Para o conseguirmos usar no programa tem de ser convertido.

Figura - Json

## O Programa

### Abreviaturas:

Para deixarmos a codificação mais fácil e sabermos qual é o objeto que estamos a chamar usamos abreviaturas; quase todas as abreviaturas são utilizadas por todos os programadores. Vou demonstrar os utilizados no programa:

Backgroundworker -> bgw...

Button -> btn...

ComboBox -> cmb...

Label -> lbl...

NotifyIcon -> icn...

OpenFileDialog -> ofd...

Panel -> pnl...

PictureBox -> pic\_...

TextEdit -> txt...

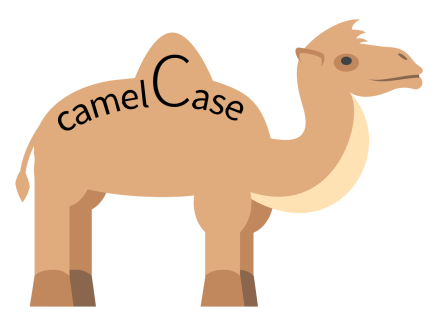
NumericUpDown -> rdo...

DataTimePicker -> dtp...

FlowLayoutPanel -> flp...

Form -> frm...

Timer -> tmr...

Onde estão as reticências, colocamos o nome da utilização do objeto, por exemplo.: *btnGuardar*, *btnLogin*, *txtUserName* .

### CamelCase

Para identificarmos o objeto e a sua função usamos a abreviatura dos objetos mais a sua função. Existe várias técnicas, há mais comum de os desenvolvedores usarem é a *CamelCase*, a primeira palavra pode ser escrita em *UpperCamelCase* ou *lowerCamelCase*, mas o resto tem de ser escrito em maiúscula.

Figura - camelCase

Optamos por usar o *lowerCamelCase*, por ser mais fácil e versátil.

### Linq

*Language* *Integrated* *Query* ou consulta integrada à linguagem, é uma tecnologia, e um mecanismo para facilitar consulta dos dados organizados da lista.

O *LINQ* é mais comum de ser usado nas consultas de bancos de dados usando o *Entity* *Framework*.

Figura - *Linq*

### *Class*

Uma *Class* ajuda-nos a retornar valores e a enviar valores para aquela variável. Nas listas associamos as classes como o tipo de variável.

Figura - *Class* Cliente

Por exemplo para enviarmos os valores para as variável da class temos de escrever o nome da classe, neste caso Cliente. Depois para chamar a class usamos o Cli.Nome.



Figura - Enviar os dados para a *class*

### Ligações a base de dados

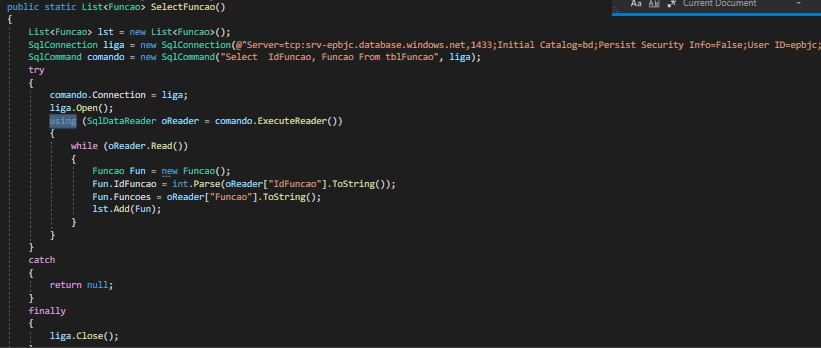


Figura - Ligação a base de dados

Na imagem 47 temos o comando *Select* IdFuncao e Funcao para conseguir identificar a função do funcionário e o seu id. Com o *SqlCommand* faço o comando que quero que seja executado, o *SqlConnection* é o nome do servidor para tentar ligar ao mesmo. Ligo o comando.connection a ligação do sql e abro a ligação. Mando os dados para o *SqlDataReader* para certificar se existe dados e envia todos os dados do *DataReader* para uma lista criada. Após a sua finalização ele retorna os valores.

### Try, catch finally

Está ferramenta ajuda muito na executação do codigo. Tenta executar aqueles comandos, se aparecer algum erro, entra no *catch*. Após finalizar o *try* ou o *catch* ele vai para o *finally*, onde nós metemos a liga.*Close*();

### Notificações

O 1 é a duração que o balão das notificações aparece, o 2 é o título da mensagem, o 3 é o texto e o ponto 4 é o icon da notificação.

Figura - comando para mostrar o balão das notificações

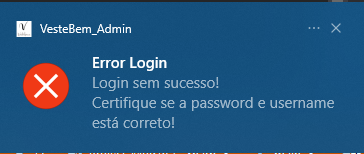


Figura - Balão de Notificações

### Stored Procedure

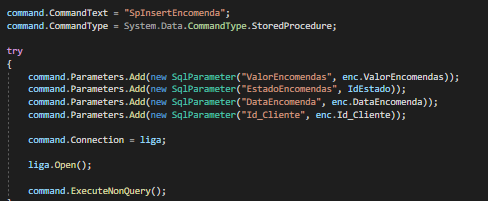


Figura - C# mais Stored Procedure

A *Stored* *Procedure* ajuda muito na execução de código SQL. Em vez de escrever muitas vezes o mesmo código(“*SELECT* \* *FROM* tbl...”), basta só chamar a *Stored* *Procedure* e enviar os seus campos para ela e após isso exetuar e os dados são enviados com sucesso.

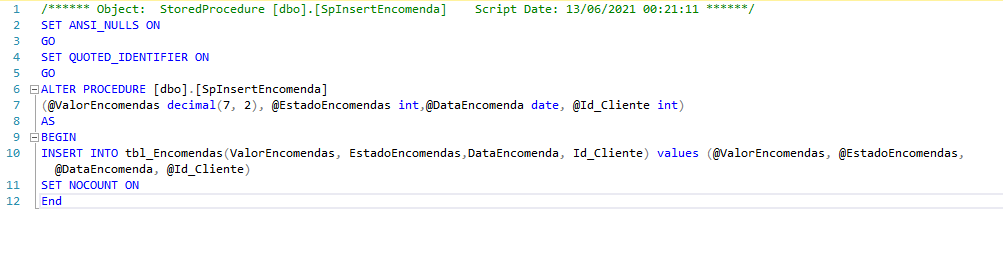
Criamos variáveis temporárias que são associadas as do nosso projeto e podemos inserir na tabela.

Figura - Stored Procedure Insert Encomendas

### Auto-Run

Figura - Código do Auto Run

O *RegistryKey* é uma variável onde podemos enviar informação para a centrar do sistema operativo. Normalmente todos os processos do Windows estão guardados no *regedit*, onde podemos ver todos os eventos do Windows, até o do nosso projeto.

### Projeto

Figura - Form quando iniciamos sessão

Quando arrancamos com a aplicação deparamos com este layout. Para o *textbox* do *username* e a *textbox* da *password* estarem completos, precisamos de fazer login no site primeiro e só depois conseguimos fazer login no aplicativo.

 (Form Administrador)

Após efetuar o login com sucesso poderá o funcionário aceder a este *form*, onde poderá adicionar produtos, Editar ou Apagar Produtos, Modificar Clientes e Funcionário, Fazer Encomendas, Consultar Carrinho e Terminar Sessão.

Figura - Form dos Administradores

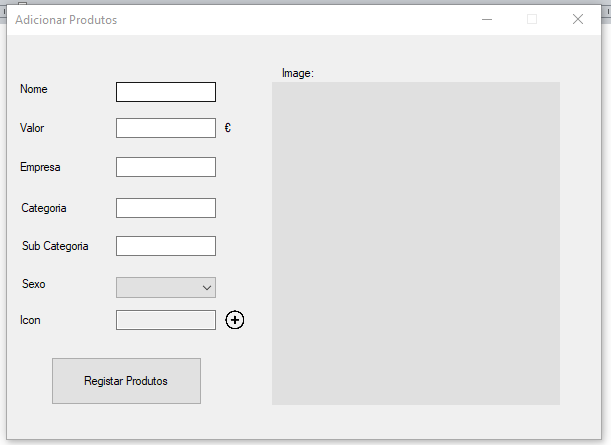
Se clicar no Adicionar Produtos, ele irá mostrar outro *form* para adicionar um novo Produto. Algumas *textbox*, só aceitam números e outras só aceitam texto. Um exemplo disso é a *textBoxValor* só números e os pontos.

Figura - Form Adicionar Produtos

Após adicionar os dados todos, irá aparecer uma notificação do programa a dizer que os dados foram adicionados com sucesso. E fecha o *form* e o *formAdministrador* aparece novamente.

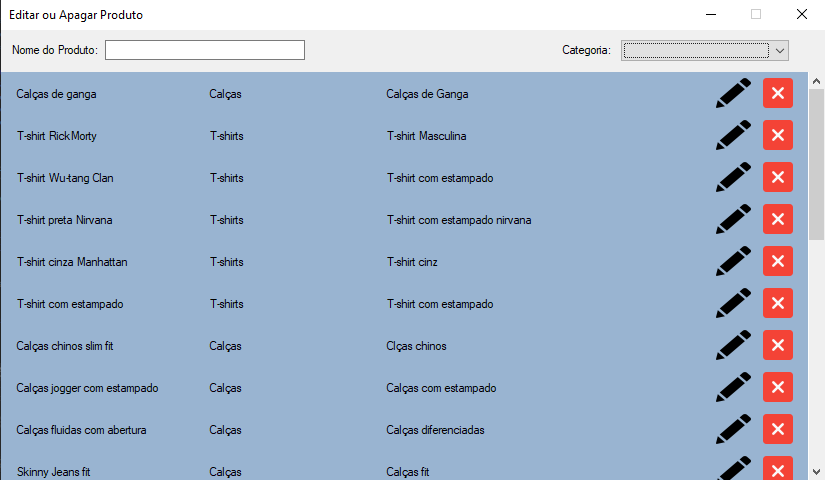
Para consultar os produtos todos temos de clicar no Editar ou Apagar Produto, onde o Funcionário poderá apagar o produto ou editar o mesmo, que levará ao *form* anterior com os dados daquele produto. Poderá procurar todos os produtos ou pelo nome ou pela categoria.

Figura - Formulário de Editar ou Apagar Produtos

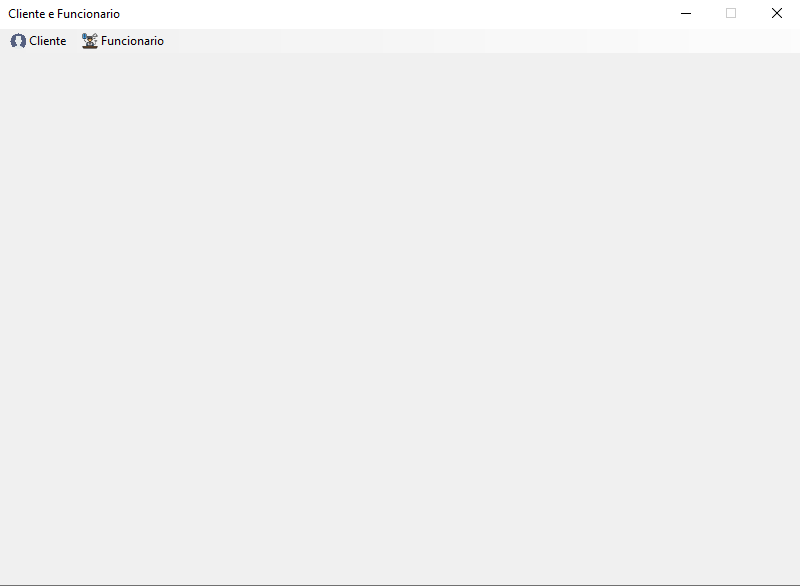
No *form* Cliente e Funcionário conseguimos consultar os funcionários e clientes. Para consultar todos os clientes, temos de clicar em cima do *toolStripMenuItem* onde diz Cliente, e aparece todos os clientes da Base de Dados. Poderá procurar todos os clientes com aquele nome, modificar os seus dados e eliminar. Mesma coisa para o funcionário, mas o funcionário poderá adicionar

Figura - Inicio do *Form* Cliente e Funcionário

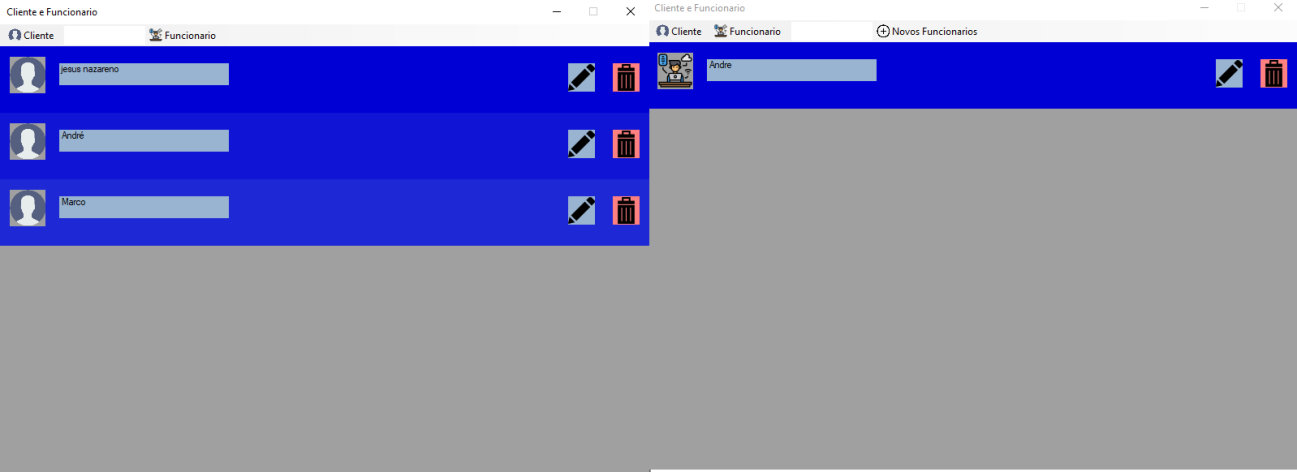
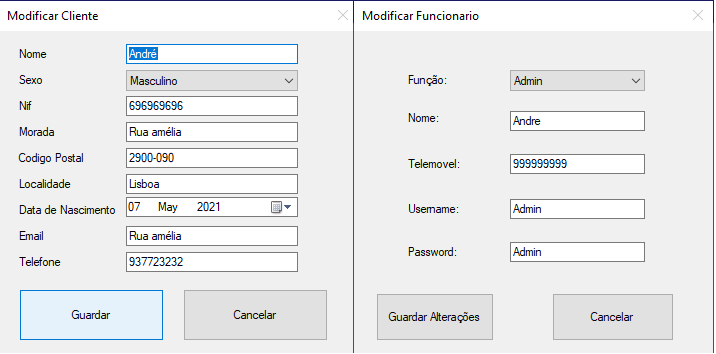
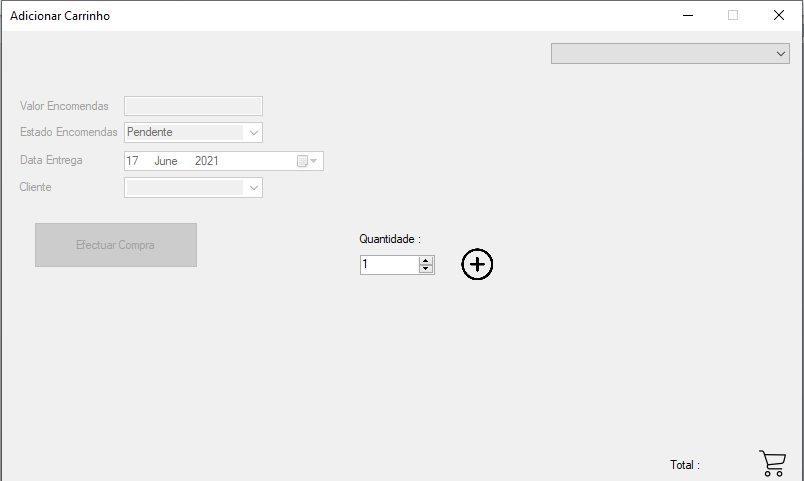


Figura - Quando clicamos no Clientes ou Funcionários

Como no adicionar produtos, as *textbox* só aceitam certos caracteres, no *Nif* só aceitam números e no telemóvel. No nome só aceita texto como na localidade. Após as alterações feitas o funcionário clica no guardar e vai voltar para o outro *form* e este é atualizado automaticamente.

Figura - Modificar os Dados do Cliente e Funcionário

Desenvolvi este *form*, o Adicionar Carrinho, caso o cliente vá a loja efetuar a encomenda de um produto.

O Funcionário seleciona um produto na *combobox* e depois escreve a quantidade e depois clica na *picturebox* com um sinal de mais e o produto será adicionado a lista de compras. Quando achar que acabou de efetuar a compra, clica no carrinho e os dados serão enviados para o lado esquerdo, depois tem de selecionar o cliente onde será efetuado a compra e mais ou menos a data de entrega. Quando tiver finalizado é só clicar efetuar Compra.

(Form de Adicionar Carrinho)

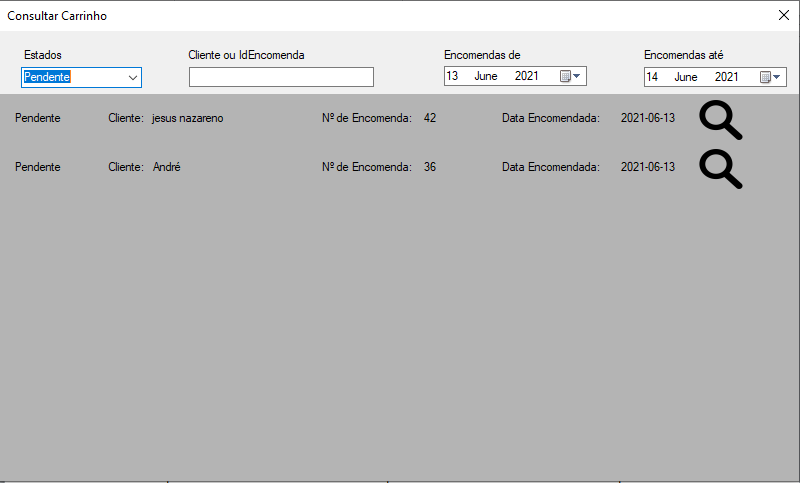
O funcionário consegue consultar todos os carrinhos disponíveis no *form* Consultar Carrinho Conseguem pesquisar pelo nome de cliente, Id da Encomenda e selecionar as encomendas naquele período de tempo. Para ver o que os clientes compraram tem de clicar na lupa e irá aparecer todos os produtos do carrinho.

Figura - Form Consulta Carrinho

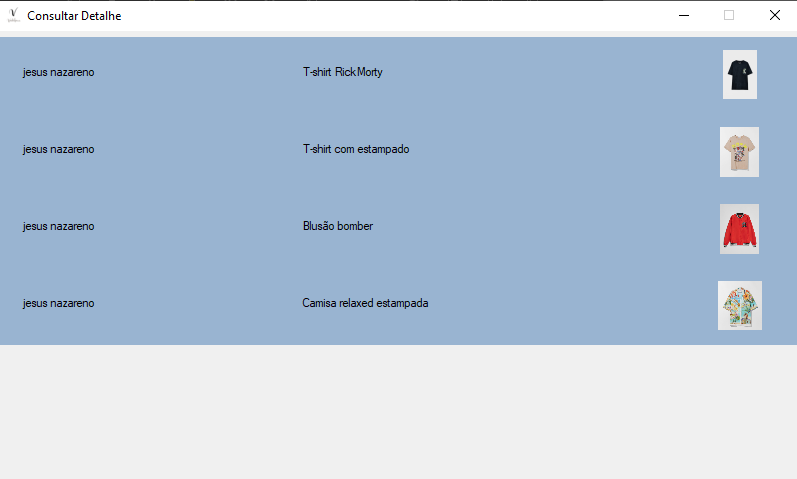
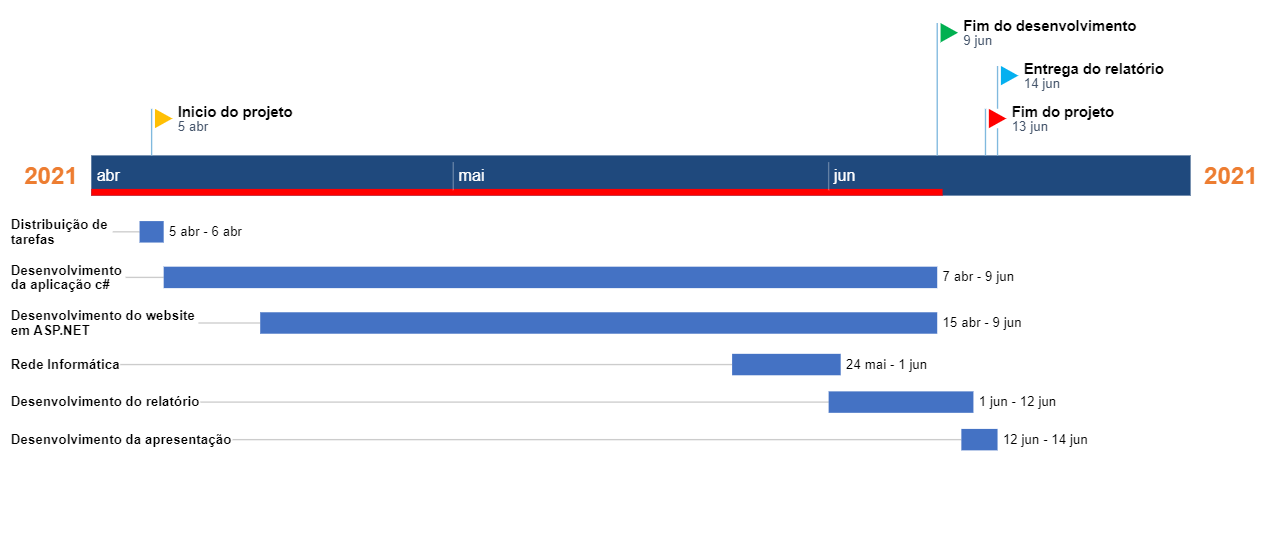
Nas definições só tem uma *checkBox* onde ativa o programa com o arranque do windows e por final temos o terminar sessão onde e volta para o *form* de Login. Capítulo IV – Recursos Utilizados

Figura - Detalhes de Produtos

|  |  |
| --- | --- |
| Resultado de imagem para visual studio 2015 logo | **Visual Studio –** Software utilizado para a programação da aplicação. |
| Resultado de imagem para office timeline logo  Resultado de imagem para powerpoint logo | **Power Point & Office Timeline –** Software utilizado na concessao da apresentação e no cronograma |
| Resultado de imagem para word logo | **Word –**Utilizado para realizar o relatório |
|  | **Opera–** Navegador de internet utilizado para pesquisar informação e para esclarecer dúvidas. |
|  | **Cisco Packet Tracer** – Utilizado para fazer Diagrama da Rede. |
|  | **Microsoft SQL server management studio –** Sistema gestor de base dados utilizado na concessão da nossa base de dados |
| File:Github-desktop-logo-symbol.svg - Wikimedia Commons | **Git Hub –** Plataforma onde se guarda os projetos graças ao *Git*, que envia tudo para o *GitHub*. Usamos o *GitHub* para enviarmos o projeto um para o outro. |

# Capítulo IV – Cronograma Final e Justificação de desvios



Cronograma Final

## Justificação

O cronograma inicial não foi cumprido por alguma falta de conhecimento e alguns contratempos, assim como a aprendizagem respetiva aos conhecimentos necessários tendo assim terminado a concessão da aplicação e do website no dia 09/06/2021 e a rede informática no dia 01/06/2021, enquanto que o fim do relatório e da apresentação se realizaram no prazo esperado para tal.

## Conclusão

Após terminarmos este relatório concluímos que este projeto foi bastante longo, pois eram bastantes coisas a serem feitas durante a realização do mesmo.

É claro que tivemos dificuldade com o projeto pois haviam certas técnicas e conhecimentos que nunca tínhamos usado como a Encriptação da palavra-passe durante o registo do Cliente e a sua desencriptação durante o Login, entre outros fatores.

Após terminarmos este projeto sentimo-nos mais aliviados pois numa certa altura estávamos bastantes cansados, não só com o projeto como também com outras disciplinas

É também com os erros que aprendemos e nos tornamos melhores pessoas e, no nosso caso, melhores programadores. Adquirimos bastantes conhecimentos e métodos de trabalho.

## Bibliografia e Web Grafia:

GitHub:

- <https://github.com/>

Gmail:

- <https://www.google.com/intl/pt-PT/gmail/about/>

StackOverFlow:

- <https://stackoverflow.com/>

Youtube:

- <http://youtube.com>