IPV- Instituto Politécnico de Viseu

ESTV- Escola Superior de Tecnologia de Viseu

Departamento de Informática



Relatório do Projeto de ABD

Licenciatura em Engenharia Informática

Realizado por:

Jorge Filipe Marques da Silva, nº11594

Cláudia Filipa Ferreira Trigo, nº11190

Bruno Rafael Costa Ribeiro, nº12919

Docentes:

Filipe Caldeira

Paulo Tomé

Carlos Cunha

Viseu, 2015

IPV- Instituto Politécnico de Viseu

ESTV- Escola Superior de Tecnologia de Viseu

Departamento de Informática

Por:

Jorge Filipe Marques da Silva, nº11594

Cláudia Filipa Ferreira Trigo, nº11190

Bruno Rafael Costa Ribeiro, nº12919

Docentes:

Filipe Caldeira

Paulo Tomé

Carlos Cunha

Viseu, 2015

**Índice**

[1. Introdução 1](#_Toc411547315)

[2. Estrutura da Base de dados 2](#_Toc411547316)

[2.1 Criação das Tabelas 2](#_Toc411547317)

[2.2 Modelos de Dados 3](#_Toc411547318)

[2.3 Sequencias 6](#_Toc411547319)

[2.4 Funções 7](#_Toc411547320)

[2.5 Triggers 7](#_Toc411547321)

[2.6 Procedimentos e Packages 8](#_Toc411547322)

[2.7 Vistas 9](#_Toc411547323)

[2.8 Controlo de Concorrência 10](#_Toc411547324)

[3. Estrutura da Aplicação 11](#_Toc411547325)

[3.1 Login 11](#_Toc411547326)

[3.2 Desenvolvimento da Aplicação 12](#_Toc411547327)

[4. Conclusão 20](#_Toc411547328)

**Índice de Figuras**

[Ilustração 2.1 - Estrutura tabela Erros 3](file:///C:\Users\Utilizador\Dropbox\ABD\2014-2015\Projecto%20de%20ABD\Trabalho%20final\Relatório\Relatório%20trabalhoABD.docx#_Toc411547151)

[Ilustração 2.2 - Modelo Er 4](#_Toc411547152)

[Ilustração 2.3 - Modelo PDM 5](#_Toc411547153)

[Ilustração 2.4 - Tratamento de Erros 10](file:///C:\Users\Utilizador\Dropbox\ABD\2014-2015\Projecto%20de%20ABD\Trabalho%20final\Relatório\Relatório%20trabalhoABD.docx#_Toc411547154)

[Ilustração 3.1 - Menu Administrador 12](#_Toc411547155)

[Ilustração 3.2 - Inserir Vendedor 13](#_Toc411547156)

[Ilustração 3.3 - Listar Vendedores 14](#_Toc411547157)

[Ilustração 3.4 - Listar e alterar das vendas 14](#_Toc411547158)

[Ilustração 3.5 - Ver ganhos taxados 15](#_Toc411547159)

[Ilustração 3.6 - Alterar taxa do vendedor 15](#_Toc411547160)

[Ilustração 3.7 - Login com Tratamento de Erros 16](#_Toc411547161)

[Ilustração 3.8 - Apresentação da Loja a Visitantes 16](#_Toc411547162)

[Ilustração 3.9 - Registar Cliente 17](#_Toc411547163)

[Ilustração 3.10 - Loja apresentada ao Cliente 18](#_Toc411547164)

[Ilustração 3.11 - Área do Cliente 18](#_Toc411547165)

[Ilustração 3.12 - Edição de dados do Cliente 19](#_Toc411547166)

[Ilustração 3.13 - Carrinho 19](#_Toc411547167)

**Índice de Tabelas**

[Tabela 3.1 - Login do Cliente 11](#_Toc411547213)

[Tabela 3.2 - Login do Vendedor 11](#_Toc411547214)

# Introdução

O objectivo deste trabalho tem como base o desenvolvimento para a empresa N&N uma aplicação que dê suporte a de produtos de diversos tipos através da Internet, onde pretende comercializar duas classes de produtos: produtos da empresa e produtos de outras lojas com as quais mantém protocolos.

Cada cliente da loja tem de estar registado na aplicação e efetua compras através do elemento vulgar nestas situações: o carrinho de compras. Por sua vez cada compra pode ser paga através de diferentes formas: Visa, Serviço e compras, Transferência bancária e Vale CTT.

O esquema do lado de servidor do *software* foi desenvolvido numa base de dados Oracle, em PL/SQL.

Em relação à parte cliente foi desenvolvida em ASP.NET com o *software* Visual Studio 2013.

# Estrutura da Base de dados

Neste capítulo será demonstrado todos os passos que foram elaborados para a estrutura da base de dados.

Criação das Tabelas

Quanto á criação das tabelas, podemos afirmar que foram criadas onze tabelas na base de dados, cada uma com a informação necessária para a empresa. Esta é a nossa estrutura ao nível das tabelas, para a Base de Dados. As tabelas que estão alojadas na base de dados são:

* Cliente
* Fornecedor
* Vendedor
* Produto
* Carrinho
* Encomendas
* Tipo de Estado
* Tipo de Pagamento
* Venda
* Itens Venda

A tabela erros é usada para armazenar os erros que vão ocorrendo no sistema, o código do erro, a descrição do erro e a data da ocorrência. A estrutura da tabela erros será a seguinte:

Ilustração 2.1 - Estrutura tabela Erros

Create table ERROS  
(  
Id\_Erro number(10,0) constraint PK\_Erros primary key,  
Codigo number(10,0),  
Descricao varchar(100),  
Datas date   
);

Modelos de Dados

Nesta parte iremos mostrar os modelos que utilizamos no nosso trabalha, em especial os que deram origem á criação das tabelas bem como os seus devidos relacionamentos. Assim iremos ter a informação necessária na base de dados para depois se proceder a elaboração da aplicação. Elaboramos o modelo ER, que se pode ser na Ilustração 2.2, e o modelo PDM que se pode ser na Ilustração 2.3

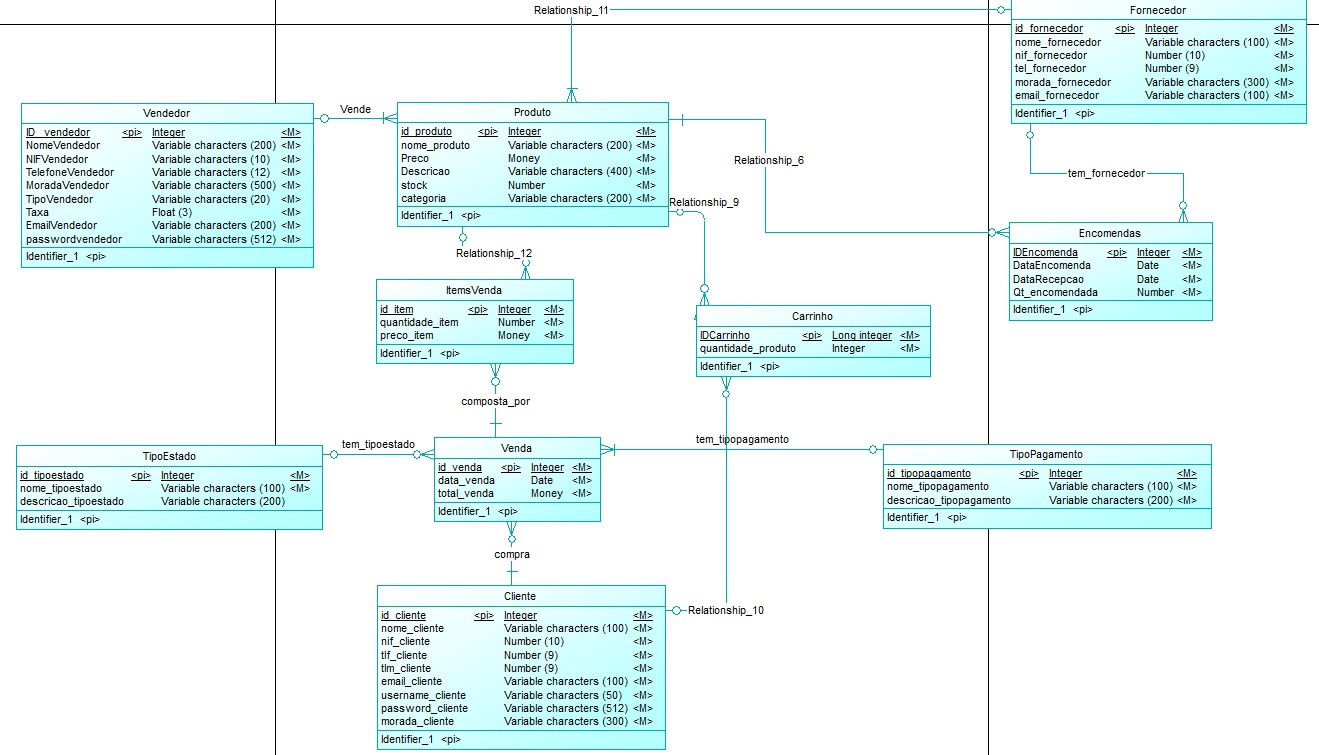


Ilustração 2.2 - Modelo Er

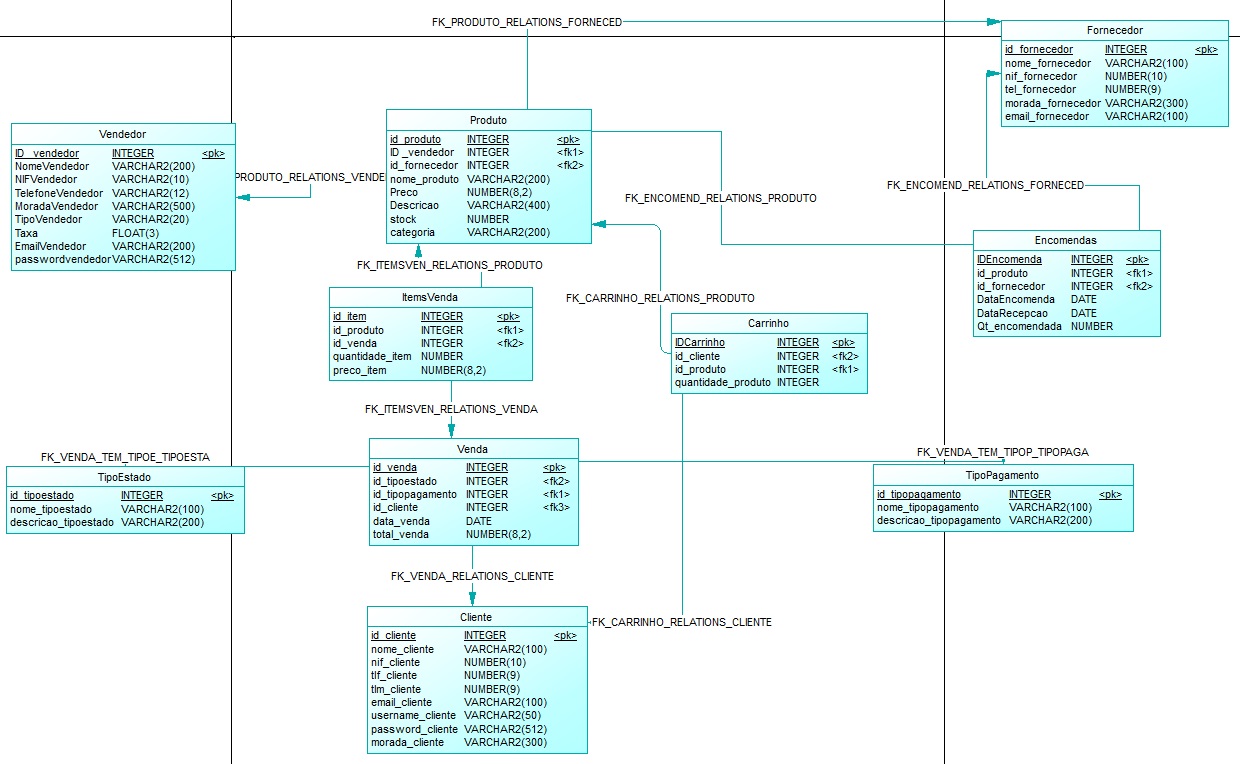


Ilustração 2.3 - Modelo PDM

Sequencias

Uma sequência é um objecto que permite gerar números inteiros únicos. Mesmo que vários utilizadores acedam à sequência, é garantido que nenhum obtém um número que outro já tenha obtido. O principal uso de sequências é para a geração automática de chaves primárias.

Na nossa base de dados existem as seguintes:

* Erros
* Carrinho
* Cliente
* Encomendas
* Fornecedor
* Itens venda
* Produto
* Tipo estado
* Tipo pagamento
* Venda
* Vendedor

Funções

Um programa em PL/SQL pode ser armazenado no servidor como um stored function. Isto torna a utilização de programas em PL/SQL pelas aplicações cliente muito mais eficiente. Foram elaboradas algumas funções neste trabalho, que são as seguintes:

* Função para o login do cliente
* Função para o login do vendedor
* Função para o total do carrinho
* Função para devolver a taxa
* Função para devolver os ganhos taxados

Triggers

Um trigger é um bloco PL/SQL que executa implicitamente sempre que um evento particular acontece.

Existem três tipos de execução: Before, After ou Instead of. Foram utilizados alguns triggers na estrutura da base de dados como:

* Trigger para a actualização do stock
* Trigger para a actualização do stock de venda cancelada
* Trigger para criar encomenda do stock baixo

Procedimentos e Packages

Um programa em PL/SQL pode ser armazenado no servidor como um stored procedure. Isto torna a utilização de programas em PL/SQL pelas aplicações cliente muito mais eficiente, visto o procedimento ficar armazenado no servidor já compilado e pronto a ser executado por qualquer utilizador ou aplicação que tenha recebido privilégios para execução do procedimento.

Um package é um agrupamento lógico de vários procedimentos e funções, tipos ou até mesmo variáveis e constantes.

Vantagens de criação de pacotes são o agrupamento lógico de procedimentos e funções que estão relacionados entre si, a criação de um repositório comum para o código: o pacote é armazenado como uma unidade distinta no servidor, o aumento do desempenho: o pacote é todo carregado em memória após a primeira execução, o encapsulamento: possibilidade de definir subprogramas que apenas são visíveis dentro do pacote, apenas utilizáveis pelos programas do pacote.

Na nossa estrutura da base de dados pode-se dizer que foram usados vários procedimentos dentro dos packages que foram criados para gerir a aplicação. Foram criados alguns packages utilizados como:

* Package do vendedor
* Package do cliente
* Package do produto
* Package do carrinho
* Package das vendas
* Package finaliza compra
* Package do fornecedor

Vistas

Uma vista é um objecto básico do schema de uma base de dados Oracle, sendo um modo de acesso aos dados de uma ou mais tabela. As vistas são usadas geralmente para estabelecer níveis adicionais de segurança para dados da tabela, derivar a informação de dados base da tabela e simplificar o desenvolvimento de aplicações. Criando Vistas alteráveis, podemos usar comandos para introduzir, actualizar, e suprimir dados.

Foram usadas algumas vistas na estrutura da Base de Dados.

* Vista dos detalhes do produto
* Vista dos itens por pagar
* Vista dos itens pagos
* Vista dos itens cancelados

Controlo de Concorrência

O controlo de concorrência serve para bloquear o acesso a determinados dados que estão a ser editados, afim de não serem editados duas vezes, ou de se perder a informação que se tentou editar.

No nosso trabalho utilizamos como recurso a opção “update nowait” fez-se o controlo e a concorrência para editar os dados de um dado cliente, assim quando o cliente vai editar os seus dados há um bloqueio e não pode ser feita edição noutra sessão até que o cliente terminar ou cancelar a edição dos seus dados.

No que diz respeito ao tratamento de erros o utilizador é informado que os dados serão bloqueados e que deve tentar mais tarde, daí a existência da tabela erros que foi criada juntamente com as outras tabelas. Na Ilustração 2.4 abaixo é demostrado os tratamentos de erros utilizados no trabalho.

Ilustração 2.4 - Tratamento de Erros

20001 - 20100 -> package Fornecedor

20101 - 20200 -> package Vendas

20201 - 20300 -> package Produto

20301 - 20400 -> package Vendedor

20401 - 20500 -> package Cliente

20501 - 20600 -> package Finaliza compra

20601 - 20700 -> package carrinho

20701 - 20800 -> funções

20801 - 20900 -> package encomendas

# Estrutura da Aplicação

Para a elaboração da aplicação utilizou-se o programa Microsoft Visual Studio 2013.

Na aplicação elaborada para a empresa N\_N podemos referir que foram criadas três interfaces, ou seja, o vendedor, o cliente e o administrador.

Login   
Na nossa aplicação podemos referir dois tipos de login o do cliente e o do vendedor. Para o perfil cliente foi produzido um login com diferenciação de tipo de utilizador como se vê na Tabela 3.1 abaixo e o mesmo foi feito para o vendedor como se vê pela análise da Tabela 3.2.

|  |  |
| --- | --- |
| **Username** | **Password** |
| Jorgefms | Passwordjorge |
| Filipatrigo | Passwordfilipa |
| Rafa | Passwordrafa |
| Tnobre | passwordnobre |
| Susana | passwordsusana |

Tabela 3.1 - Login do Cliente

|  |  |
| --- | --- |
| **Username** | **Password** |
| geral@N\_N.pt | 1234 |
| geral@xpto.pt | 1234 |
| geral@infosoft.pt | 1234 |

Tabela 3.2 - Login do Vendedor

Desenvolvimento da Aplicação

Nesta parte iremos demonstrador alguns dos formulários realizados na nossa aplicação para a empresa N\_N.

Os formulários que serão apresentados dizem respeito ao administrador onde existe um menu administrador como se pode ver pela Ilustração 3.1 abaixo.



Ilustração 3.1 - Menu Administrador

É possível inserir um vendedor com os seguintes campos apresentados na Ilustração 3.2 abaixo.

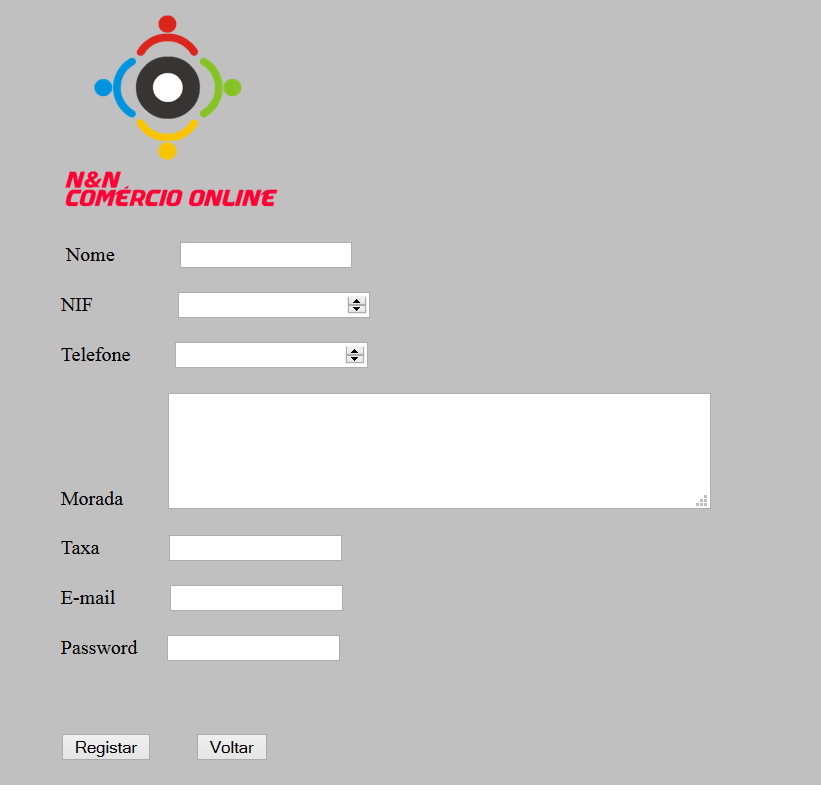


Ilustração 3.2 - Inserir Vendedor

.

Uma das funcionalidades que o administrador pode fazer é listar os vendedores existentes na aplicação, como se pode ver pela análise detalhada da Ilustração 3.3.

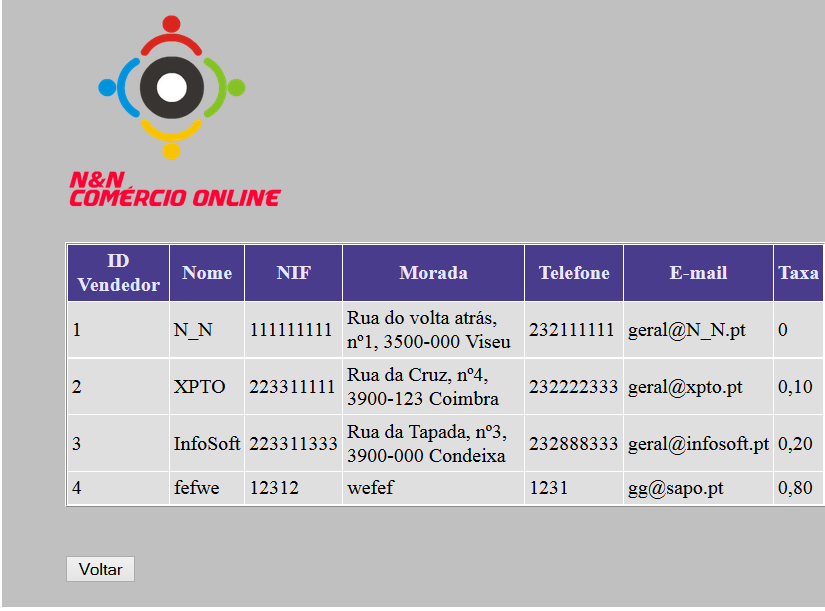


Ilustração 3.3 - Listar Vendedores

Também é possível visualizar na nossa aplicação o listar e alterar o estado de vendas, com a observação da Ilustração 3.4 abaixo podemos ver isso.

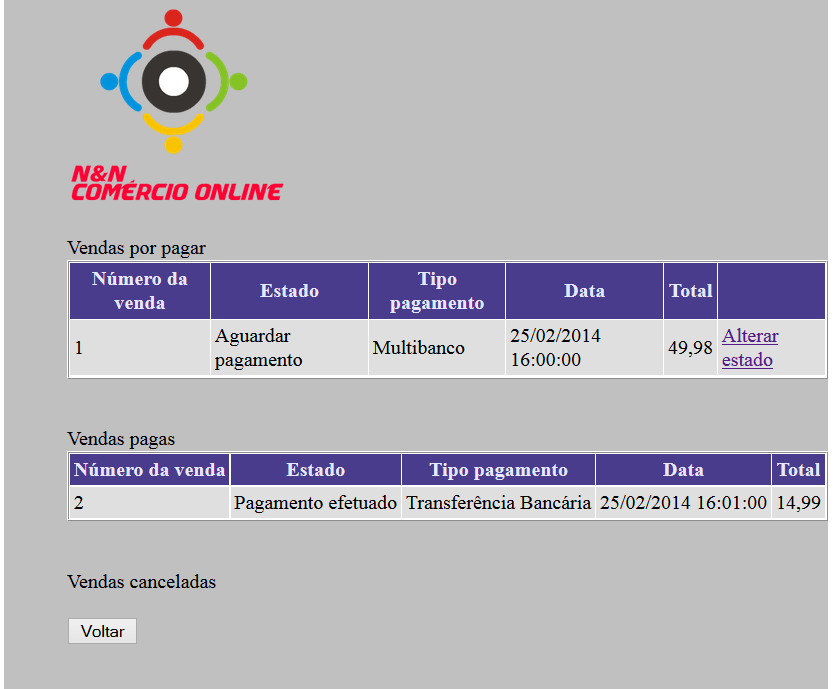


Ilustração 3.4 - Listar e alterar das vendas

O administrador também funcionalidade de ver os ganhos taxados, ver Ilustração 3.5, e alterar a taxa do vendedor, ver Ilustração 3.6.



Ilustração 3.5 - Ver ganhos taxados

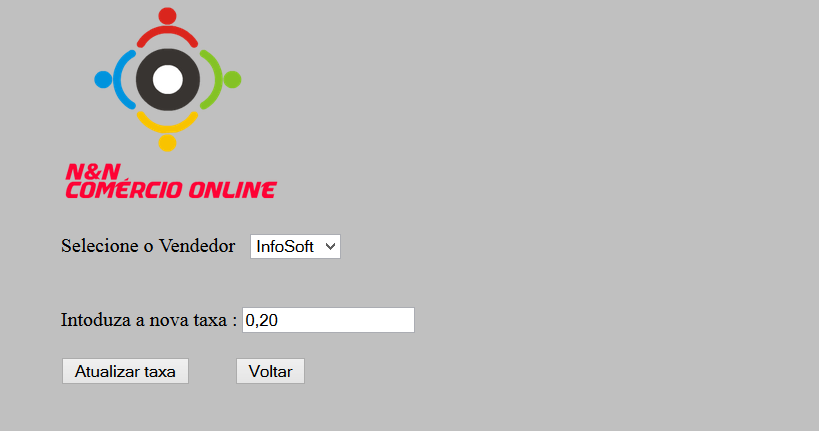


Ilustração 3.6 - Alterar taxa do vendedor

Os formulários que serão apresentados dizem respeito ao cliente, de seguida será demostrado o login respectivo com o devido tratamento de erros, ver Ilustração 3.7.

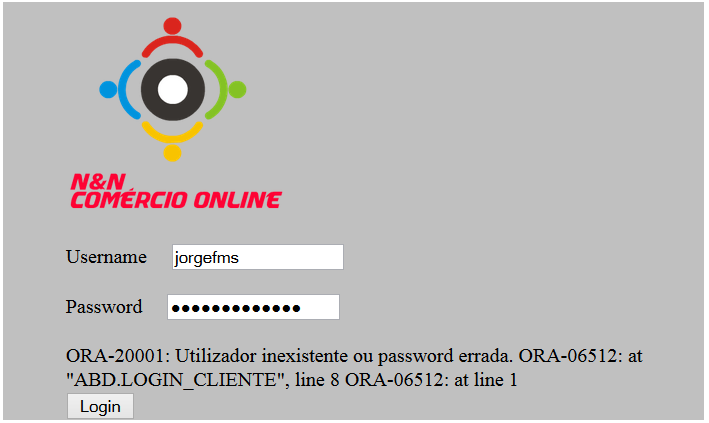


Ilustração 3.7 - Login com Tratamento de Erros

Nesta aplicação é possível fazer uma apresentação da loja aos visitantes como se vê pela Ilustração 3.8 abaixo.



Ilustração 3.8 - Apresentação da Loja a Visitantes

Uma das funcionalidades que se pode fazer na aplicação é o registo de um novo cliente como se na Ilustração 3.9.

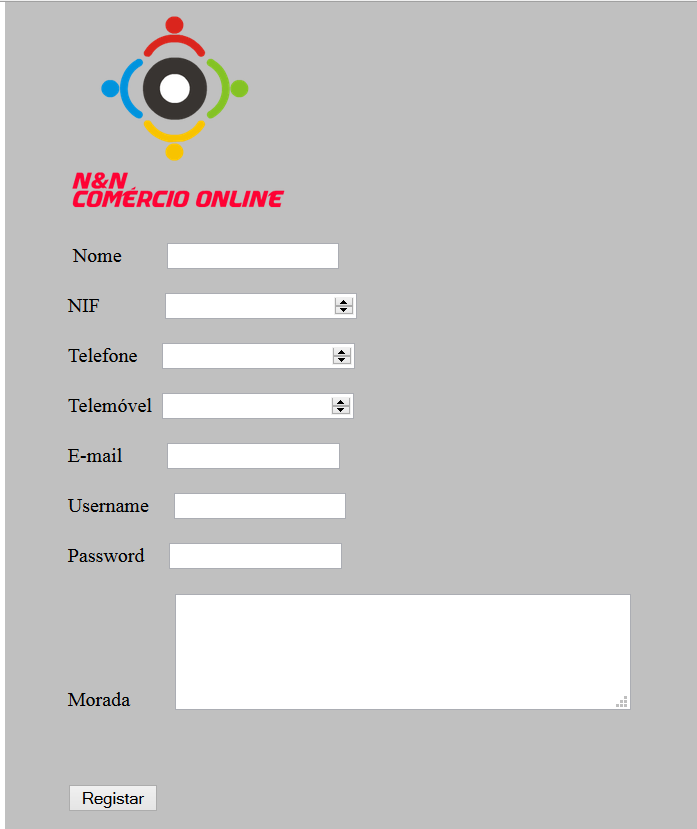


Ilustração 3.9 - Registar Cliente

Também é possível que a loja seja apresentada ao cliente com opção de ver o carrinho, adicionar produtos ao carrinho, ver área pessoal e consultar dados completos dos produtos, com a análise da Ilustração 3.10.

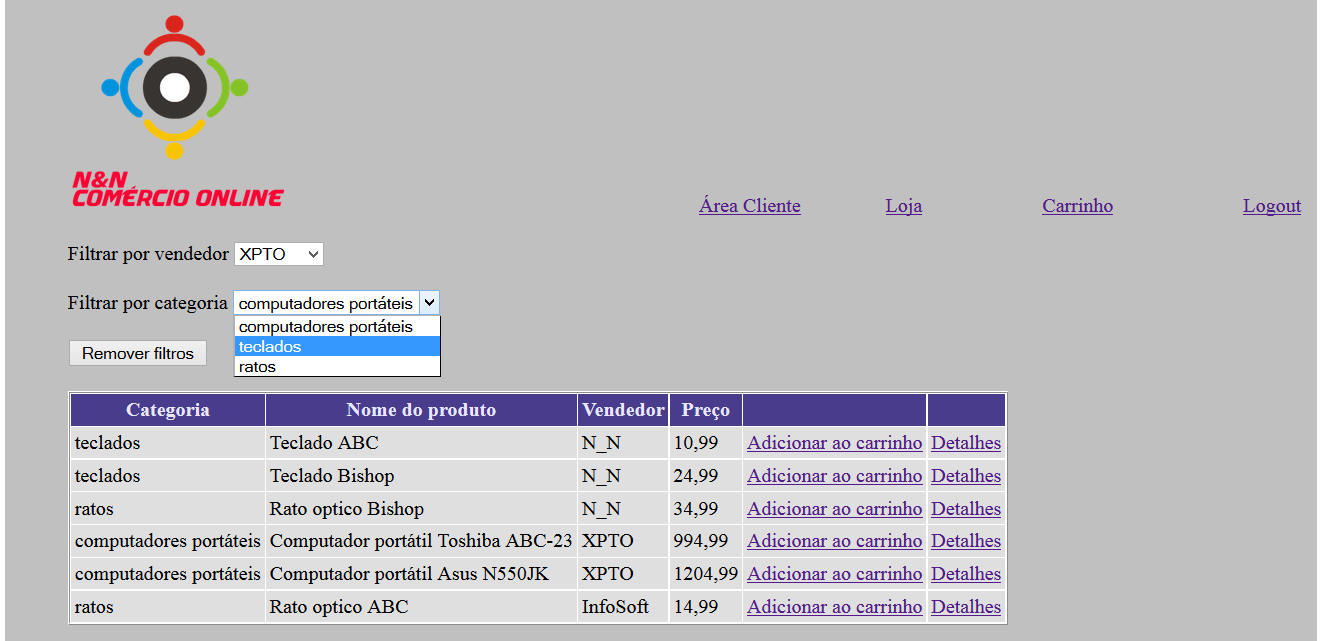


Ilustração 3.10 - Loja apresentada ao Cliente

No que diz respeito á área de cliente, podem-se ver os dados pessoais do cliente e histórico de compras, como se pode ver pela Ilustração 3.11.

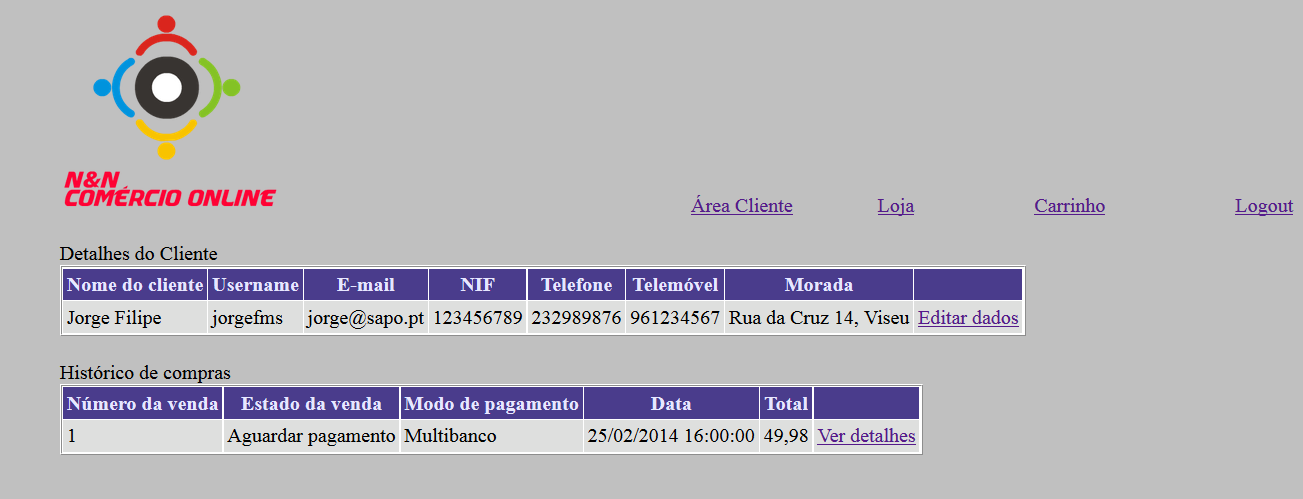


Ilustração 3.11 - Área do Cliente

Quanto á edição de dados do cliente foi feita com o uso do controlo de concorrência, ver Ilustração 3.12.

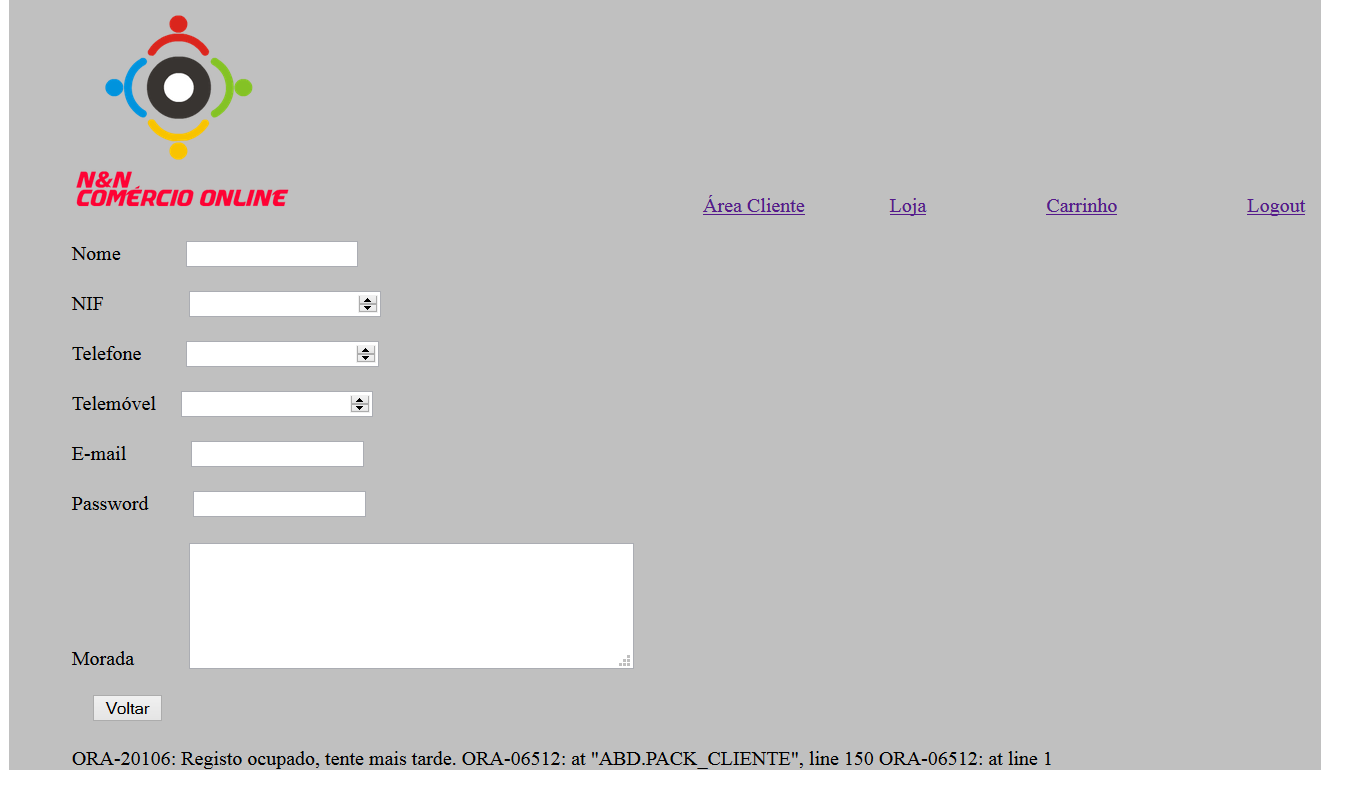


Ilustração 3.12 - Edição de dados do Cliente

No que diz respeito ao carrinho, é possível alterar quantidade de cada produto, seleccionar modo de pagamento, finalizar a compra, ou remover itens do carrinho, observar Ilustração 3.13o.

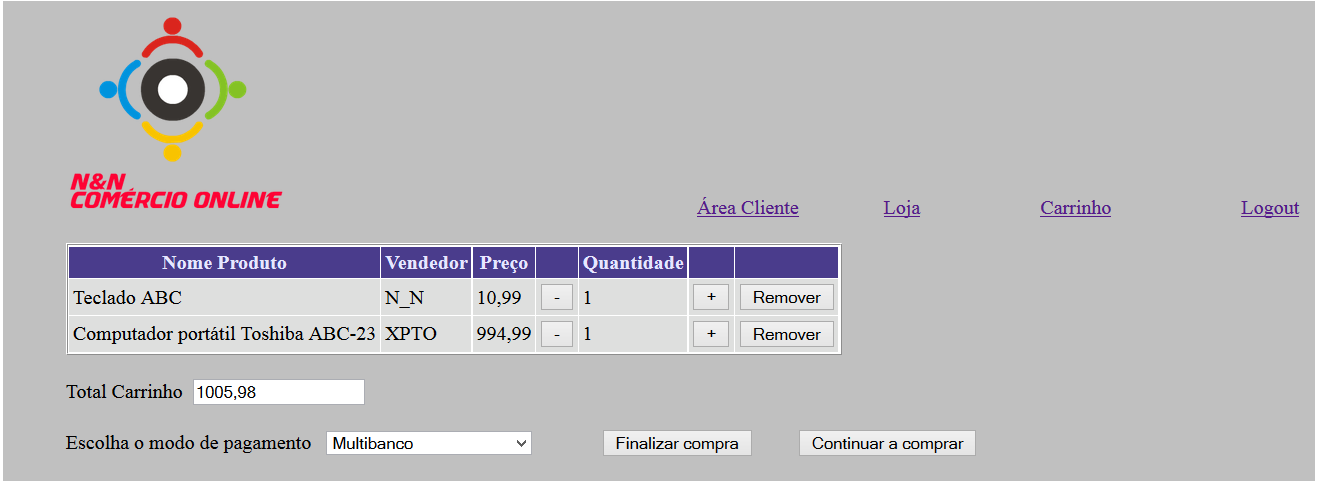


Ilustração 3.13 - Carrinho

# Conclusão

A realização deste projeto no âmbito da unidade curricular *Administração de Base Dados,* foi deveras importante para o nosso desenvolvimento de conhecimentos e experiência a nível de técnicas de programação para base de dados e sua administração bem como para a Web, que de certeza irá ser muito útil futuramente.

Durante a realização deste projeto, encontrámos diferentes dificuldades principalmente na parte da interface ASP.net, mas que se tentou ultrapassar ao longo da realização do trabalho.  
  
Apesar destas dificuldades, achamos que obtivemos um bom projeto, onde conseguimos um bom resultado final que esperamos que se venha a refletir na avaliação.