



UNIVERSIDADE DO MINHO

LICENCIATURA EM ENGENHARIA INFORMÁTICA

BASE DE DADOS

EyeBuy
Sistema de comércio electrónico

Grupo 14:

ANDRÉ SANTOS N°60994

DANIEL ARAÚJO N°61058

DANIEL CARVALHO N°61008

HELENA ALVES N°61000



6 de Fevereiro de 2013

Conteúdo

1	Introdução	5
2	Sistema EyeBuy	6
3	Estudo de Concorrência	7
3.1	Ebay	7
3.2	Olx	7
3.3	Miau	7
3.4	Conclusão do estudo de concorrência	8
4	Base de Dados	9
4.1	Normalização	9
4.2	Esquema conceptual da base de dados	10
4.3	Implementação	11
4.3.1	Criação das tabelas	11
4.3.2	Sequências	16
4.3.3	Views	16
4.3.4	Funções	18
4.3.5	Procedimentos	21
4.3.6	Triggers	24
4.3.7	Tipos	29
4.4	Packages	29
4.5	CronJobs	31
5	Aplicação administrador	33
5.1	Linguagens e Ferramentas usadas na implementação	33
5.2	Segurança	33
5.3	Interface aplicação administrador	34
5.3.1	Login	34
5.3.2	Dashboard	34
5.3.3	Produtos	36
5.3.4	Consultar produto	39
5.3.5	Editar produto	39
5.3.6	Notificações	41
5.3.7	Caixa de entrada e saída	41
5.3.8	Nova Mensagem	42
5.3.9	Consultar Mensagens	44
5.3.10	Consultar utilizadores	45

5.3.11	Consultar perfil utilizador	45
5.3.12	Editar perfil utilizador	49
5.3.13	Categorias	49
5.3.14	Subcategorias	51
5.3.15	Filtro de conteúdos	51
5.3.16	Configurações	53
6	Aplicação cliente	53
6.1	Camada ORM	54
6.2	Decisões relevantes nas listagens de produtos	54
6.3	Implementação do Modelo das Três camadas (Interface, Camada de Negócio, Camada de Persistência)	55
6.4	Interface: Experiência de utilização	55
6.5	Segurança	56
6.6	Interface aplicação cliente	57
6.6.1	Login	57
6.6.2	Página inicial	58
6.6.3	Pesquisar um produto	58
6.6.4	Pesquisa avançada	59
6.6.5	Ver produto	60
6.6.6	Ver produto em leilão	61
6.6.7	Ver perfil de um utilizador	62
6.6.8	Perfil de um utilizador	62
6.6.9	Editar perfil de utilizador	63
6.6.10	Criar nova mensagem	64
6.6.11	Consultar uma mensagem	64
6.6.12	Página de vendas	65
6.6.13	Consultar carrinho de compras	66
6.6.14	Histórico	67
6.6.15	Sugestão de troca	68
6.6.16	Proposta de troca	69
6.6.17	Ajuda	70
6.6.18	Sobre o EyeBuy	70
6.6.19	Mensagem de erro/aviso	71
7	Conclusão	72

Lista de Figuras

1	Esquema conceptual da base de dados	10
2	CronJob que verifica e actualiza o término dos leilões	32
3	Últimas actividades do CronJob	32
4	Página de login para um administrador	34
5	Página inicial para um administrador	35
6	Página de consulta e pesquisa de produtos	37
7	Exemplo de uma procura	38
8	Página de um produto	39
9	Página de edição um produto	40
10	Página de notificações	41
11	Página de consulta das mensagem recebidas	42
12	Página de consulta das mensagem enviadas	42
13	Página de nova mensagem de um administrador	43
14	Página de consulta de uma mensagem enviada	44
15	Página de consulta de uma mensagem recebida	44
16	Página de consulta dos utilizadores do sistema	45
17	Página de consulta de um perfil de um utilizador	46
18	Estatísticas como vendedor disponíveis na consulta de um perfil de um utilizador	47
19	Produtos na montra disponíveis na consulta de um perfil de um utilizador	47
20	Produtos comprados disponíveis na consulta de um perfil de um utilizador	48
21	Produtos vendidos disponíveis na consulta de um perfil de um utilizador	48
22	Página de edição de um perfil de um utilizador	49
23	Página de consulta e gestão de categorias	50
24	Página de consulta e gestão de subcategorias	51
25	Página de filtro de conteúdo	52
26	Página de configurações	53
27	Página de login para um utilizador	57
28	Página inicial do sistema	58
29	Página pesquisar produto	59
30	Página pesquisa avançada	60
31	Página de um dado produto	61
32	Página de uma dado produto em leilão	61
33	Página de um perfil de uma dado urilizador	62
34	Página de perfil do utilizador	63
35	Página editar perfil do utilizador	63
36	Página criar nova mensagem	64
37	Página consultar mensagem	65

38	Página da montra de produtos do utilizador	66
39	Página do carrinho de compras do utilizador	67
40	Página do histórico do utilizador	68
41	Página de sugestão de troca	69
42	Página proposta de troca	69
43	Página de ajuda do sistema	70
44	Página de informação sobre o sistema	71
45	Mensagem de aviso	71

1 Introdução

A disciplina de base de dados, para além de servir como uma introdução aos fundamentos teóricos que sustentam a teoria por trás das bases de dados relacionais teórica também serve como uma introdução à prática e à utilização real de sistemas de gestão de bases de dados.

De acordo com o segundo ponto referido no parágrafo anterior, surge este projecto que tem como objectivo a utilização de um contexto real do SGBD da Oracle, englobando todos os conceitos abordados na Unidade Curricular de Bases de Dados EI, com realce para a utilização da SQL e da PL/SQL. Para além disso, foram dominadas tecnologias que permitissem criar interfaces para os utilizadores (WEB e Desktop).

Assim, neste relatório é descrito o processo de análise e concepção de um sistema e-Commerce (tema proposto) que permite aos utilizadores comprarem e venderem produtos. Este sistema permite aos utilizadores comprar e vender produtos, criar leilões e fazer licitações, funcionamento de suporte a sugestões de compra e troca de produtos, criação de montras de produtos, produtos desejados e produtos seguidos, suporte à interacção entre utilizadores com um sistema de mensagens e avaliação entre outras funcionalidades inerentes a este tipo de sistema. São também disponibilizadas funcionalidades de administração e de gestão das entidades envolvidas na aplicação. Para além disso, é também apresentada uma análise da concorrência onde são retratados os sistemas do mesmo tipo, assim como os seus pontos fortes e pontos fracos de forma a ser feita uma melhor implementação possível do sistema.

Por fim, são expostos os desenhos e planificações das interfaces e da sua co-relação com as funcionalidades a serem implementadas no sistema.

No desenvolvimento do projecto foram desenvolvidas duas aplicações, uma primeira orientada ao cliente desenvolvida em Java e uma aplicação web orientada à administração de todo o sistema em PHP.

2 Sistema EyeBuy

O eyeBuy é um sistema de e-commerce que proporciona aos seus utilizadores a possibilidade de venda e compra de produtos.

Um utilizador registado no sistema pode assumir diferentes papéis no sistema. Para o papel de utilizador normal são garantidas as opções de compra, licitação ou troca de produtos e gestão do carrinho de compras, manutenção do seu perfil, criação de listas de produtos desejados e produtos seguidos, interagir com outros utilizadores através de mensagens e fazer uma avaliação relativa a uma compra efetuada assim como reportar situações de compra à administração. São também garantidas as opções de venda de produtos e aceitação de propostas e manutenção de uma montra de produtos à venda. Por fim o papel de administrador do sistema permite a gestão dos utilizadores do sistema, gestão dos produtos, criar alertas para todos os utilizadores, gerir as categorias gerais em que os produtos podem ser inseridos, gestão das newsletters relativas a produtos, consulta de estatísticas gerais do sistema e gestão de restrições dentro do sistema para os utilizadores.

Em termos genéricos o funcionamento deste sistema permite a um utilizador avançado criar um anúncio de venda de um produto onde constam o nome do produto, preço, descrição, fotografias, métodos de venda (venda direta, licitação ou troca), métodos de pagamentos, método de envio e categoria e ser permitido a utilizadores normais comprarem/licitarem esse produto enviando-o para o seu carrinho de compras e pagando posteriormente. No fim da transação é possível também atribuir um feedback entre vendedor e comprador de forma a avaliar o nível de confiança no vendedor e no seu produto e a facilidade de transação com o comprador de forma a ser possível através dessa avaliação indicar se um vendedor ou comprador é confiável ou não. Como suporte aos utilizadores é também possível reportar uma situação numa transação ao administrador do sistema caso não tenha sido possível na interação vendedor-comprador.

O carrinho de compras torna possível a adição de vários produtos para compra, assim como a remoção caso se desista da compra, de forma a ser possível fazer um pagamento único para todos os artigos facilitando assim as transações na compra de produtos.

De forma a assegurar um alto grau de interação e acessibilidade do sistema é possível também criar listas de produtos desejados tendo em vista garantir ao utilizador guardar produtos nos quais esteja interessado comprar futuramente (caso existam ou não para venda no sistema) e criar listas de produtos seguidos que torna possível um utilizador seguir o percurso de licitação de um produto até o leilão atingir o seu término.

A montra de produtos é outra ferramenta acessível para utilizadores avançados que permite a organização e uma melhor visibilidade de todos os produtos que esse utilizador colocou à venda de forma a assegurar a vendedores mais activos um espaço no formato loja online dentro do sistema de forma a facilitar as transações de produtos similares no mesmo vendedor.

3 Estudo de Concorrência

3.1 Ebay

O Ebay é um site de comércio online, onde é possível negociar inúmeros produtos, desde pequenas coisas até grandes investimentos. É o site líder em vendas a nível mundial, tendo sido o pioneiro neste tipo de negócios. Em termos de oferta, esta é muita variada desde artigos novos a usados, sendo os utilizadores pertencentes a pequenas e médias empresas e de nível individual.

É centro comercial (de compra e venda) mais popular, abrangente e bem-sucedido do mundo online. Está presente em 37 países e é grande para toda a gente. Actualmente, existe um vendedor para cada 78 compradores.

O modelo de negócio do eBay permite que empresas de pequena e média dimensão e pessoas a título individual (que gerem pequenos negócios a partir de casa) exerçam com êxito a sua actividade, assim como, auxilia muitos lojistas do comércio tradicional (lojas de rua), a escoarem os seus produtos através da exportação, cobrindo deste modo um mercado bem mais abrangente.

O Ebay permite várias formas de vendas e leilões tratando também dos detalhes de transacção de produtos entre os utilizadores, garantindo assim o sistema mais seguro e eficaz de todos do género. Para além disso o ebay é altamente organizado e extremamente rico em funcionalidades disponíveis para o utilizador. O ebay oferece funcionalidades como carrinho de compras, lista de produtos seguidos, seguimento de leilões, possibilidade de criação de uma montra para vendedores em forma de loja online, possibilidade de reportar produtos e contactar outros utilizadores, sistema de reolução de problemas que permite aos compradores remeterem um problema para ser resolvido entre o próprio ebay e os vendedores entre muitas outras funcionalidades que asseguram a fiabilidade e segurança do sistema.

3.2 Olx

O olx é um site de anúncios que permite aos seus utilizadores criar anúncios de venda e compra de produtos assim como anúncios mais gerais que vai desde o aluguer de imóveis a serviços prestados e desde automóveis a procura e oferta de emprego. O olx oferece a funcionalidade de criar anúncios, responder a anunciantes, comentar anúncios, mas todo o processo de negociação e transacção é levado a cabo por parte dos utilizadores do site, tornando-o uma plataforma de anúncios simples, eficaz e com um propósito específico.

3.3 Miau

O miau é um site de classificados que permite aos seus utilizadores comprarem e venderem produtos através de venda directa e leilões para além de ser possível colocar perguntas aos anunciantes na própria página do produto à vista de todos. Um dos pontos fortes do miau é

a sua política de segurança e a forma com que verifica os utilizadores através de verificação da sua morada de forma a prevenir fraudes e outros esquemas ilegais.

3.4 Conclusão do estudo de concorrência

Após uma intensa análise às funcionalidades e funcionamento dos sistemas concorrentes foi possível determinar os seus pontos fortes de forma a implementá-los neste sistema assim como os seus pontos fracos de forma a evitá-los na nossa implementação. Algumas das funcionalidades fortes nesses sistemas passam por permitir a criação de produtos desejados assim como a possibilidade de seguir produtos em leilão, a criação de montras de produtos à venda pelo mesmo utilizador de forma a possibilitar uma interacção com vendedores mais avançados de forma a assemelhar-se com uma loja online, um sistema de feedbacks que permite a avaliação de compradores e vendedores mediante transações passadas garantido assim uma forma qualitativa do grau de confiança dos utilizadores, reportar produtos e problemas em anúncios, entre outras funcionalidades.

4 Base de Dados

4.1 Normalização

A abordagem seguida para a normalização da base de dados foi baseada na intuição adquirida nas aulas da Unidade Curricular. Assim após ter sido criada, a base de dados ocorreu o processo de normalização até à verificação da 3^a Forma Normal (a requerida no enunciado). Após ter sido inicialmente desenhado o ”schema” da base de dados recorrendo ao Visual Paradigm, tornou-se necessário recorrer a técnicas de standardização e melhoria de performance através da eliminação de dependências funcionais. Assim sendo, tornou-se necessária a iteração a partir de um primeiro ”schema” em que as tabelas se encontravam na 1^a Forma Normal (não contêm grupos ou valores repetidos). Começamos por verificar se todos os atributos não pertencentes à chave são apenas e só dependentes da chave. E resolver esse tipo de dependência colocando a base de dados na 2^a forma normal. Finalmente, após isso verificamos a ausência de dependências transitivas. , após eliminar as dependências entre elementos não pertencentes às chaves obtivemos tabelas na 3^a forma normal.

4.2 Esquema conceptual da base de dados

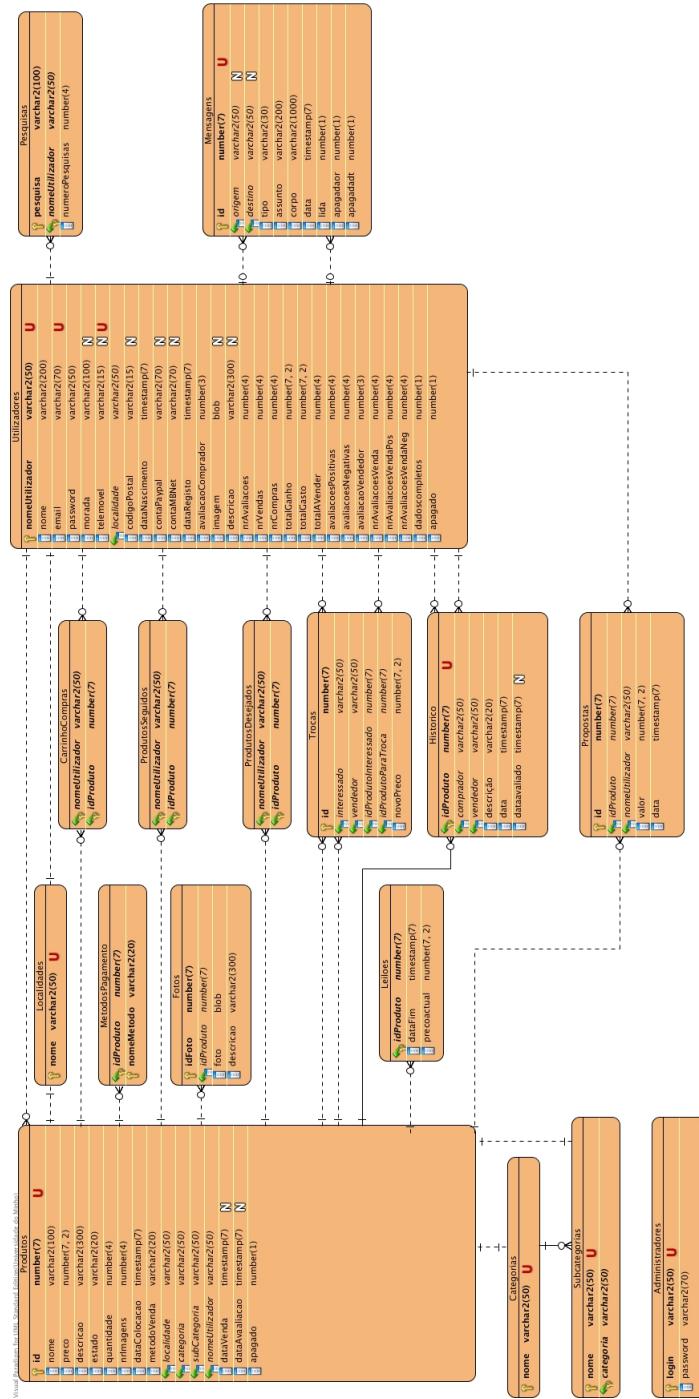


Figura 1: Esquema conceptual da base de dados

4.3 Implementação

4.3.1 Criação das tabelas

```
CREATE TABLE Produtos (
    id          number(7) NOT NULL,
    nome        varchar2(100) NOT NULL,
    preco       number(7,2) NOT NULL,
    descricao   varchar2(300) NOT NULL,
    estado      varchar2(20) NOT NULL,
    quantidade  number(4) DEFAULT 1 NOT NULL,
    nrImagens  number(2) DEFAULT 0 NOT NULL,
    dataColocacao timestamp(7) DEFAULT CURRENT_TIMESTAMP NOT NULL,
    metodoVenda varchar2(20) NOT NULL,
    localidade  varchar2(50) NOT NULL,
    categoria   varchar2(50) NOT NULL,
    subCategoria varchar2(50) NOT NULL,
    nomeUtilizador varchar2(50) NOT NULL,
    dataVenda    timestamp(7),
    dataAvaliacao timestamp(7),
    apagado     number(1) DEFAULT 0 NOT NULL,
    PRIMARY KEY (id));

CREATE TABLE ProdutosDesejados (
    nomeUtilizador varchar2(50) NOT NULL,
    idProduto      number(7) NOT NULL,
    PRIMARY KEY (nomeUtilizador,
    idProduto));

CREATE TABLE Mensagens (
    id          number(7) NOT NULL,
    origem      varchar2(50),
    destino     varchar2(50),
    tipo        varchar2(30) NOT NULL,
    assunto     varchar2(200) NOT NULL,
    corpo        varchar2(1000) NOT NULL,
    data        timestamp(7) NOT NULL,
    lida        number(1) DEFAULT 0 NOT NULL,
    apagadaor  number(1) DEFAULT 0 NOT NULL,
    apagadadt  number(1) DEFAULT 0 NOT NULL,
```

```

PRIMARY KEY (id));

CREATE TABLE Historico (
    idProduto number(7) NOT NULL,
    comprador varchar2(50) NOT NULL,
    descricao varchar2(20),
    data      timestamp(7) DEFAULT CURRENT_TIMESTAMP NOT NULL,
    dataAvaliado timestamp(7),
    PRIMARY KEY (idProduto));

CREATE TABLE CarrinhoCompras (
    nomeUtilizador varchar2(50) NOT NULL,
    idProduto      number(7) NOT NULL,
    PRIMARY KEY (nomeUtilizador,
    idProduto));

CREATE TABLE ProdutosSeguidos (
    nomeUtilizador varchar2(50) NOT NULL,
    idProduto      number(7) NOT NULL,
    PRIMARY KEY (nomeUtilizador,
    idProduto));

CREATE TABLE Fotos (
    idFoto number(7) NOT NULL,
    idProduto number(7) NOT NULL,
    foto      BLOB,
    descricao varchar2(300) NOT NULL,
    PRIMARY KEY (idFoto));

CREATE TABLE MetodosPagamento (
    idProduto number(7) NOT NULL,
    nomeMetodo varchar2(20) NOT NULL,
    PRIMARY KEY (idProduto,
    nomeMetodo));

CREATE TABLE Utilizadores (
    nomeUtilizador      varchar2(50) NOT NULL,
    nome                  varchar2(200) NOT NULL,

```

```

email          varchar2(70) NOT NULL UNIQUE,
password       varchar2(50) NOT NULL,
morada         varchar2(100),
telemovel      varchar2(15) UNIQUE,
localidade     varchar2(50) NOT NULL,
codigoPostal   varchar2(15),
dataNascimento timestamp(7) NOT NULL,
contaPaypal    varchar2(70),
contaMBNet     varchar2(70),
dataRegisto    timestamp(7) DEFAULT CURRENT_TIMESTAMP NOT NULL,
avaliacaoComprador number(3) DEFAULT 100 NOT NULL,
imagem         BLOB,
descricao       varchar2(300),
nrAvaliacoes   number(4) DEFAULT 0 NOT NULL,
nrVendas        number(4) DEFAULT 0 NOT NULL,
nrCompras       number(4) DEFAULT 0 NOT NULL,
totalGanho      number(7,2) DEFAULT 0 NOT NULL,
totalGasto      number(7,2) DEFAULT 0 NOT NULL,
totalAVender    number(4) DEFAULT 0 NOT NULL,
avaliacoesPositivas number(4) DEFAULT 0 NOT NULL,
avaliacoesNegativas number(4) DEFAULT 0 NOT NULL,
avaliacaoVendedor number(3) DEFAULT 100 NOT NULL,
nrAvaliacoesVendaPos number(4) DEFAULT 0 NOT NULL,
nrAvaliacoesVendaNeg number(4) DEFAULT 0 NOT NULL,
dadosCompletos  number(1) DEFAULT 0 NOT NULL,
apagado         number(1) DEFAULT 0 NOT NULL,
PRIMARY KEY (nomeUtilizador));

```

```

CREATE TABLE Categorias (
  nome varchar2(50) NOT NULL,
  PRIMARY KEY (nome));

```

```

CREATE TABLE Subcategorias (
  nome      varchar2(50) NOT NULL,
  categoria varchar2(50) NOT NULL,
  PRIMARY KEY (nome, categoria));

```

```

CREATE TABLE Localidades (

```

```

    nome varchar2(50) NOT NULL,
    PRIMARY KEY (nome));

CREATE TABLE Pesquisas (
    pesquisa      varchar2(100) NOT NULL,
    nomeUtilizador  varchar2(50) NOT NULL,
    numeroPesquisas number(4) DEFAULT 0 NOT NULL,
    PRIMARY KEY (pesquisa,
    nomeUtilizador));

CREATE TABLE Trocas (
    id            number(7) NOT NULL,
    interessado   varchar2(50) NOT NULL,
    vendedor      varchar2(50) NOT NULL,
    idProdutoInteressado number(7) NOT NULL,
    idProdutoParaTroca  number(7) NOT NULL,
    novoPreco     number(7,2) NOT NULL,
    PRIMARY KEY (id));

CREATE TABLE Administradores (
    login      varchar2(50) NOT NULL,
    password  varchar2(70) NOT NULL,
    PRIMARY KEY (login));

CREATE TABLE Leiloes (
    idProduto  number(7) NOT NULL,
    dataFim    timestamp(7) NOT NULL,
    precoactual number(7,2) NOT NULL,
    PRIMARY KEY (idProduto));

CREATE TABLE Propostas(
    id          number(7) NOT NULL,
    idProduto   number(7) NOT NULL,
    idUtilizador  varchar2(50) NOT NULL,
    valor       number(7,2) NOT NULL,
    data        timestamp(7) NOT NULL,
    PRIMARY KEY (id));

```

```

CREATE TABLE Restricoes(
    palavra VARCHAR2(20 BYTE)
    PRIMARY KEY (palavra));

ALTER TABLE CarrinhoCompras ADD CONSTRAINT FK_CarrinhoCo721169
    FOREIGN KEY (idProduto) REFERENCES Produtos (id);
ALTER TABLE CarrinhoCompras ADD CONSTRAINT FK_CarrinhoCo329760
    FOREIGN KEY (nomeUtilizador) REFERENCES Utilizadores (nomeUtilizador);
ALTER TABLE Fotos ADD CONSTRAINT FK_Fotos634962
    FOREIGN KEY (idProduto) REFERENCES Produtos (id);
ALTER TABLE Leiloes ADD CONSTRAINT FK_Leiloes57290
    FOREIGN KEY (idProduto) REFERENCES Produtos (id);
ALTER TABLE Mensagens ADD CONSTRAINT FK_Mensagens46787
    FOREIGN KEY (origem) REFERENCES Utilizadores (nomeUtilizador);
ALTER TABLE Mensagens ADD CONSTRAINT FK_Mensagens939873
    FOREIGN KEY (destino) REFERENCES Utilizadores (nomeUtilizador);
ALTER TABLE MetodosPagamento ADD CONSTRAINT FK_MetodosPag607420
    FOREIGN KEY (idProduto) REFERENCES Produtos (id);
ALTER TABLE Pesquisas ADD CONSTRAINT FK_Pesquisas938924
    FOREIGN KEY (nomeUtilizador) REFERENCES Utilizadores (nomeUtilizador);
ALTER TABLE Produtos ADD CONSTRAINT FK_Produtos665268
    FOREIGN KEY (localidade) REFERENCES Localidades (nome);
ALTER TABLE Produtos ADD CONSTRAINT FK_Produtos519750
    FOREIGN KEY (subCategoria, categoria) REFERENCES Subcategorias (nome, categoria);
ALTER TABLE Produtos ADD CONSTRAINT FK_Produtos884609
    FOREIGN KEY (nomeUtilizador) REFERENCES Utilizadores (nomeUtilizador);
ALTER TABLE ProdutosDesejados ADD CONSTRAINT FK_ProdutosDe782820
    FOREIGN KEY (nomeUtilizador) REFERENCES Utilizadores (nomeUtilizador);
ALTER TABLE ProdutosDesejados ADD CONSTRAINT FK_ProdutosDe731890
    FOREIGN KEY (idProduto) REFERENCES Produtos (id);
ALTER TABLE ProdutosSeguidos ADD CONSTRAINT FK_ProdutosSe646956
    FOREIGN KEY (idProduto) REFERENCES Produtos (id);
ALTER TABLE ProdutosSeguidos ADD CONSTRAINT FK_ProdutosSe697886
    FOREIGN KEY (nomeUtilizador) REFERENCES Utilizadores (nomeUtilizador);
ALTER TABLE Propostas ADD CONSTRAINT FK_PropostasProduto
    FOREIGN KEY (idProduto) REFERENCES Produtos (id);
ALTER TABLE Propostas ADD CONSTRAINT FK_PropostasUtilizador
    FOREIGN KEY (idUtilizador) REFERENCES Utilizadores (nomeUtilizador);

```

```

ALTER TABLE Subcategorias ADD CONSTRAINT FKSubcategor351797
    FOREIGN KEY (categoria) REFERENCES Categorias (nome);
ALTER TABLE Utilizadores ADD CONSTRAINT FKUtilizador529092
    FOREIGN KEY (localidade) REFERENCES Localidades (nome);
ALTER TABLE Trocas ADD CONSTRAINT FKTrocas903040
    FOREIGN KEY (interessado) REFERENCES Utilizadores (nomeUtilizador);
ALTER TABLE Trocas ADD CONSTRAINT FKTrocas847532
    FOREIGN KEY (vendedor) REFERENCES Utilizadores (nomeUtilizador);
ALTER TABLE Trocas ADD CONSTRAINT FKTrocas153908
    FOREIGN KEY (idProdutoInteressado) REFERENCES Produtos (id);
ALTER TABLE Trocas ADD CONSTRAINT FKTrocas295653
    FOREIGN KEY (idProdutoParaTroca) REFERENCES Produtos (id);
ALTER TABLE Historico ADD CONSTRAINT FKHistorico8683
    FOREIGN KEY (comprador) REFERENCES Utilizadores (nomeUtilizador);
ALTER TABLE Historico ADD CONSTRAINT FKHistorico662393
    FOREIGN KEY (idProduto) REFERENCES Produtos (id);

```

4.3.2 Sequências

```

CREATE SEQUENCE mensagens_id MINVALUE 0 INCREMENT BY 1
    START WITH 0 CACHE 20 NOORDER NOCYCLE ;
CREATE SEQUENCE produtos_id MINVALUE 0 INCREMENT BY 1
    START WITH 0 CACHE 20 NOORDER NOCYCLE ;
CREATE SEQUENCE trocas_id MINVALUE 0 INCREMENT BY 1
    START WITH 0 CACHE 20 NOORDER NOCYCLE ;
CREATE SEQUENCE fotos_id MINVALUE 0 INCREMENT BY 1
    START WITH 0 CACHE 20 NOORDER NOCYCLE ;
CREATE SEQUENCE propostas_id MINVALUE 0 INCREMENT BY 1
    START WITH 0 CACHE 20 NOORDER NOCYCLE ;

```

4.3.3 Views

- **Produtos**

Lista de produtos a serem vendidos:

```

CREATE OR REPLACE
VIEW v_produtosaux AS
SELECT * FROM produtos;

```

Lista de produtos a serem vendidos:

```
CREATE OR REPLACE
VIEW v_produtosMontra AS
SELECT * FROM produtos
WHERE datavenda IS NULL
AND apagado = 0;
```

Produtos ordenados pelos melhores vendedores:

```
CREATE OR REPLACE
VIEW v_dosMelhores AS
SELECT m.* FROM v_produtosmontra m, utilizadores u
WHERE m.nomeutilizador = u.nomeutilizador
ORDER BY u.avaliacaovendedor DESC;
```

Produtos ordenados pelos mais recentes:

```
CREATE OR REPLACE
VIEW v_novosProdutos AS
SELECT * FROM v_produtosmontra
ORDER BY datacolocacao DESC;
```

- Histórico

Histórico para consulta:

```
create or replace
VIEW historicoParaConsulta as
select comprador , preco, nomeUtilizador as vendedor,
       h.descricao, data, dataAvaliado
from historico h, produtos p
where h.idproduto = p.id;
```

4.3.4 Funções

- **Passwords**

Função que codifica a password com o algoritmo MD5:

```
CREATE OR REPLACE FUNCTION md5 (valor VARCHAR) RETURN VARCHAR2 IS
    v_input VARCHAR2(2000) := valor;
    hexkey VARCHAR2(32) := NULL;
BEGIN
    hexkey := RAWTOHEX(DBMS_OBFUSCATION_TOOLKIT.md5
        (input => UTL_RAW.cast_to_raw(v_input)));
    RETURN NVL(hexkey, '');
END;
```

- **Produtos**

Função que dado um ID de um produto, retorna o nome do produto correspondente:

```
create or replace
FUNCTION f_getNomeProduto (argIDProduto number) RETURN VARCHAR2 IS

    v_resNomeProduto varchar(100);

BEGIN

    SELECT nome INTO v_resNomeProduto FROM produtos
    WHERE id = argIDProduto;

    RETURN v_resNomeProduto;

END f_getNomeProduto;
```

Função que verifica se há algum produto com aquele id na montra:

```
create or replace
FUNCTION f_naMontra (argIDProduto number) RETURN BOOLEAN IS

    v_res boolean;
    v_nr number(1);
```

```

BEGIN

    SELECT count(*) INTO v_nr FROM v_produtosMontra
    WHERE ID = argIDProduto;

    IF(v_nr > 0) THEN
        RETURN TRUE;
    ELSE
        RETURN FALSE;
    END IF;

END f_naMontra;

```

- **Pesquisas**

Função que verifica as coincidências entre o nome de um produto e as pesquisas de um utilizador:

```

create or replace
FUNCTION f_nrcoincidencias
    (arg_nome VARCHAR, arg_nomeutilizador VARCHAR) RETURN number IS

v_res number(5) := 0;
v_pesq varchar2(100);

CURSOR c_palavraspesquisadas IS
SELECT pesquisa FROM
(SELECT * FROM pesquisas
WHERE nomeutilizador = arg_nomeutilizador
ORDER BY numeropesquisas DESC)
WHERE numeropesquisas/(SELECT sum(numeropesquisas) FROM pesquisas)
> 0.1;

BEGIN

OPEN c_palavraspesquisadas;

LOOP

```

```

EXIT WHEN c_palavraspesquisadas%NOTFOUND;

FETCH c_palavraspesquisadas INTO v_pesq;

IF (upper(arg_nome) LIKE ('%' || upper(v_pesq) || '%')) THEN
    v_res := v_res +1;
END IF;

END LOOP;

CLOSE c_palavraspesquisadas;

RETURN v_res;

END f_nrcoincidencias;

```

Função que verifica o número de coincidências entre o nome de um produto e as palavras na pesquisa de um utilizador:

```

CREATE OR REPLACE FUNCTION F_NRCOINCIDENCIASPARCIAIS
    (arg_nome VARCHAR, arg_nomeutilizador VARCHAR) RETURN number IS

v_res number(5) := 0;
v_pesq varchar2(100);
oStr str_array;

BEGIN

oStr := Str.Split(arg_nome, ' ');
IF oStr.COUNT > 0 THEN
    FOR i IN oStr.FIRST .. oStr.LAST
    LOOP
        IF (length(oStr(i)) > 3) THEN
            v_res := v_res + f_nrcoincidencias(oStr(i),arg_nomeutilizador);
        END IF;
    END LOOP;
END IF;

```

```

RETURN v_res;

END f_nrcoidenciasparciais;

```

4.3.5 Procedimentos

- Mensagens

Notifica um utilizador quando um produto deixa de estar nas suas listas:

```

create or replace
PROCEDURE p_notificaTrocaNaoDisponivel
(argNomeUtilizador varchar2, argIDTroca number) IS

BEGIN

INSERT INTO MENSAGENS
(id,origem,destino,tipo,assunto,corpo,data,lida,apagadaor,apagadadt) VALUES
(mensagens_id.nextval, null,argNomeUtilizador, 'Mensagem Sistema',
'Troca indisponível',

'Olá EyeBuy, venho por este meio notificar que a troca com o id
"' || argIDTroca ||

'" deixou de estar. Se não foi o utilizador a remover a rejeitar
a troca, então algum produto envolvido pode ter sido vendido ou
removido pelo utilizador que o tinha disponibilizado. Obrigado,
Cumprimentos da equipa EyeBuy',

CURRENT_TIMESTAMP, 0, 0, 0);

END p_notificaTrocaNaoDisponivel;

```

Notifica um utilizador quando uma troca deixa de estar disponível:

```

create or replace
PROCEDURE p_notificaNaoDisponivel
(argNomeUtilizador varchar2, argIDProduto number, argLocalizacao varchar2) IS

```

```

BEGIN

INSERT INTO MENSAGENS
(id,origem,destino,tipo,assunto,corpo,data,lida,apagadaor,apagadadt) VALUES
(mensagens_id.nextval, null,argNomeUtilizador, 'Mensagem Sistema',
'Produto ' || argLocalizacao || ' indisponível',

'O EyeBuy vem por este meio notificar que o produto
' || argLocalizacao || ' com o id "' || argIDProduto ||
'" foi removido de uma das suas listas. Se não foi o utilizador a remover
o produto este pode ter sido vendido ou removido pelo utilizador que o
tinha disponibilizado. Obrigado, Cumprimentos da equipa EyeBuy',

CURRENT_TIMESTAMP, 0, 0, 0);

END p_notificaNaoDisponivel;

```

Notifica um utilizador quando é banido:

```

create or replace
PROCEDURE p_notificaBanido
(argNomeUtilizador varchar2) IS

BEGIN

INSERT INTO MENSAGENS
(id,origem,destino,tipo,assunto,corpo,data,lida,apagadaor,apagadadt) VALUES
(mensagens_id.nextval, null,argNomeUtilizador, 'Mensagem Sistema',
'Banido do sistema',

'O EyeBuy vem por este meio notificar que foi banido do sistema.
No entanto tem a oportunidade de acabar os negócios já aceites.
Contacte os administradores para mais informações.
Obrigado, Cumprimentos da equipa EyeBuy',

CURRENT_TIMESTAMP, 0, 0, 0);

```

```
END p_notificaBanido;
```

- **Leilões**

Procedimento que termina um leilão e insere no histórico o comprador com a melhor proposta e marca como vendido:

```
create or replace
procedure p_acabaleilao
IS
cursor c_leilos is SELECT p.* from propostas p, produtos prod, leilos l
where p.idproduto=prod.id and prod.datavenda=null
and rownum<=1 and l.datafim<=CURRENT_TIMESTAMP;
rowprop c_leilos%ROWTYPE;
BEGIN
    open c_leilos;
    LOOP
        fetch c_leilos into rowprop;
        EXIT WHEN c_leilos%NOTFOUND;
        insert into historico (idproduto,comprador, descricao, data,dataavaliado)
        values (rowprop.idproduto, rowprop.idutilizador,null, rowprop.data, null);
        UPDATE produtos set datavenda=rowprop.data where id=rowprop.idproduto;
    END LOOP;
    close c_leilos;
END p_acabaleilao;
```

- **Carrinho**

Procedimento que marca como vendidos os produtos do carrinho e realiza operações na base de dados para coerência de dados:

```
create or replace procedure p_checkoutcarrinho (nomeuser varchar2, mpay varchar2)
IS
var_idprod number;
cursor c_carrinho is SELECT car.idproduto from carrinhocompras car
inner join metodospagamento m on car.idproduto=m.idproduto
where car.nomeutilizador=nomeuser and m.nomemetodo=mpay;
BEGIN
```

```

open c_carrinho;
LOOP
    fetch c_carrinho into var_idprod;
    EXIT WHEN c_carrinho%NOTFOUND;
UPDATE produtos set datavenda=CURRENT_TIMESTAMP where id=var_idprod;
INSERT into historico (idproduto, comprador, descricao, data, dataavaliado)
values (var_idprod, nomeuser, null, CURRENT_TIMESTAMP, null);
END LOOP;
close c_carrinho;
END p_checkoutcarrinho;

```

4.3.6 Triggers

- Mensagens

Trigger que apaga uma mensagem definitivamente quando o emissor e receptor a apagam:

```

CREATE OR REPLACE
TRIGGER trg_apagamensagens
FOR UPDATE ON mensagens
COMPOUND TRIGGER

AFTER STATEMENT IS
BEGIN

    DELETE FROM mensagens WHERE apagadaor = 1 AND apagadadt = 1;

END AFTER STATEMENT;
END trg_apagamensagens;

```

- Produtos

Trigger que apaga um produto de todas as listas (excepto lista de produtos) quando este é vendido:

```

CREATE OR REPLACE
TRIGGER trg_updateproduto
BEFORE UPDATE ON produtos

```

```

FOR EACH ROW

BEGIN

IF ((:NEW.DATAVENDA IS NOT NULL) OR (:NEW.APAGADO = 1)) THEN
    DELETE FROM produtosdesejados WHERE produtosdesejados.idProduto = :NEW.ID;
    DELETE FROM produtosseguidos WHERE produtosseguidos.idProduto = :NEW.ID;
    DELETE FROM carrinhocompras WHERE carrinhocompras.idProduto = :NEW.ID;
    DELETE FROM metodospagamento WHERE metodospagamento.idProduto = :NEW.ID;
    DELETE FROM trocas WHERE trocas.idProdutoInteressado = :NEW.ID
    OR trocas.idProdutoParaTroca = :NEW.ID;
END IF;

END trg_updateproduto;

```

• Produtos desejados

Trigger que notifica um utilizador quando foi removido pelo sistema um produto da sua lista de produtos desejados:

```

CREATE OR REPLACE
TRIGGER trg_removeprodutosdesejados
AFTER DELETE ON produtosdesejados

FOR EACH ROW

BEGIN

    p_notificaNaoDisponivel(:OLD.NOMEUTILIZADOR,:OLD.IDPRODUTO,'desejado');

END trg_removeprodutosdesejados;

```

• Produtos seguidos

Trigger que notifica um utilizador quando foi removido pelo sistema um produto da sua lista de produtos seguidos:

```
CREATE OR REPLACE
```

```

TRIGGER trg_removeprodutosseguidos
AFTER DELETE ON produtosseguidos

FOR EACH ROW

BEGIN

    p_notificaNaoDisponivel(:OLD.NOMEUTILIZADOR,:OLD.IDPRODUTO,'seguido');

END trg_removeprodutosseguidos;

```

- **Carrinho de compras**

Trigger que notifica um utilizador quando foi removido pelo sistema um produto do carrinho de compras:

```

CREATE OR REPLACE
TRIGGER trg_removecarrinhocompras
AFTER DELETE ON carrinhocompras

FOR EACH ROW

BEGIN

    p_notificaNaoDisponivel
        (:OLD.NOMEUTILIZADOR,:OLD.IDPRODUTO,'do carrinho de compras');

END trg_removecarrinhodecompras;

```

- **Trocas**

Trigger que notifica um utilizador quando foi removida uma troca pelo sistema:

```

CREATE OR REPLACE
TRIGGER trg_removetroca
AFTER DELETE ON trocas

FOR EACH ROW

```

```

BEGIN

    p_notificaTrocaNaoDisponivel(:OLD.INTERESSADO,:OLD.ID);
    p_notificaTrocaNaoDisponivel(:OLD.VENDEDOR,:OLD.ID);

END trg_removecarrinhodecompras;

```

- **Categorias**

Inserir subcategoria “Outros” associada à nova categoria inserida:

```

CREATE OR REPLACE
TRIGGER trg_insertCategoria
AFTER INSERT ON categorias
FOR EACH ROW
BEGIN
    INSERT INTO subcategorias (nome, categoria)
        VALUES ('Outros', :new.nome);
END;
/
/* Quando uma categoria é apagada, todos os produtos são colocados na categoria Outro */

CREATE OR REPLACE
TRIGGER trg_removecategorias
BEFORE DELETE ON categorias

FOR EACH ROW

BEGIN

    INSERT INTO subcategorias (nome,categoria) VALUES (:OLD.NOME,'Outros');

    UPDATE produtos      SET categoria = 'Outros',
    subcategoria = :OLD.NOME
    WHERE categoria = :OLD.NOME;

    DELETE FROM subcategorias WHERE categoria = :OLD.NOME;

```

```
END trg_removecategorias;
```

- **Subcategorias**

Quando uma subcategoria é apagada, todos os produtos são colocados na subcategoria “Outros”:

```
CREATE OR REPLACE
TRIGGER trg_removesubcategorias
BEFORE DELETE ON subcategorias

FOR EACH ROW

BEGIN

    UPDATE produtos SET subcategoria = 'Outros'
    WHERE subcategoria = :OLD.NOME AND categoria = :OLD.CATEGORIA;

END trg_removesubcategorias;
```

- **Utilizadores**

Quando um utilizador é eliminado, todos os seus produtos são eliminados:

```
CREATE OR REPLACE
TRIGGER trg_updateutilizador
BEFORE UPDATE ON utilizadores

FOR EACH ROW

BEGIN

    IF (:NEW.APAGADO = 1) THEN
        DELETE FROM produtosdesejados
        WHERE produtosdesejados.nomeUtilizador = :NEW.NOMEUTILIZADOR;
        DELETE FROM produtosseguidos
        WHERE produtosseguidos.nomeUtilizador = :NEW.NOMEUTILIZADOR;
        DELETE FROM carrinhocompras
```

```

        WHERE carrinhocompras.nomeUtilizador = :NEW.NOMEUTILIZADOR;
UPDATE produtos SET apagado = 1
        WHERE produtos.nomeUtilizador = :NEW.NOMEUTILIZADOR;
p_notificaBanido(:NEW.NOMEUTILIZADOR);

END IF;

END trg_updateutilizador;

```

- **Leilões**

Actualiza o valor actual do leilão com base na maior licitação:

```

CREATE OR REPLACE TRIGGER TRG_UPDATELEILAO
    after insert on propostas
    for each row
BEGIN
    update leilos set precoactual = :new.valor where idproduto=:new.idproduto;
END;

```

4.3.7 Tipos

- **Array**

Tipo de dados que guarda um array de varchar, utilizado para *return* de funções:

```

create or replace
TYPE str_array IS TABLE OF VARCHAR2(4000);

```

4.4 Packages

- Package que contém funções que permitem o *parsing* de um *varchar* para um array de *varchar*:

```

CREATE OR REPLACE PACKAGE STR IS
    FUNCTION Split(sText IN VARCHAR2, sDel IN VARCHAR2) RETURN str_array;
    FUNCTION Split(sText IN clob, sDel IN VARCHAR2) RETURN str_array;
END Str;

```

```

CREATE OR REPLACE PACKAGE BODY STR IS
FUNCTION Split(sText IN VARCHAR2, sDel IN VARCHAR2) RETURN str_array IS
    nStartIdx PLS_INTEGER := 1;
    nEndIdx PLS_INTEGER := 1;
    oRet str_array := str_array();
BEGIN

    IF sText IS NULL THEN RETURN oRet; END IF;

    LOOP

        nEndIdx := INSTR(sText, sDel, nStartIdx);

        IF nEndIdx > 0 THEN

            oRet.Extend;
            oRet(oRet.LAST) := SUBSTR(sText, nStartIdx, nEndIdx - nStartIdx);
            nStartIdx := nEndIdx + 1;

        ELSE

            oRet.Extend;
            oRet(oRet.LAST) := SUBSTR(sText, nStartIdx);
            EXIT;

        END IF;

    END LOOP;

    RETURN oRet;
END Split;

FUNCTION Split(sText IN clob, sDel IN VARCHAR2) RETURN str_array IS
    nStartIdx PLS_INTEGER := 1;
    nEndIdx PLS_INTEGER := 1;
    oRet str_array := str_array();

```

```

BEGIN

    IF sText IS NULL THEN RETURN oRet; END IF;
    IF DBMS_LOB.getlength(sText) = 0 THEN RETURN oRet; END IF;

    LOOP

        nEndIdx := DBMS_LOB.INSTR(sText, sDel, nStartIdx);

        IF nEndIdx > 0 THEN

            oRet.Extend;
            oRet(oRet.LAST) :=
DBMS_LOB.SUBSTR(sText, nEndIdx - nStartIdx, nStartIdx);
            nStartIdx := nEndIdx + LENGTH(sDel);

        ELSE

            oRet.Extend;
            oRet(oRet.LAST) :=
DBMS_LOB.SUBSTR(lob_loc => sText, offset => nStartIdx);
            EXIT;

        END IF;

    END LOOP;

    RETURN oRet;

END Split;

END Str;

```

4.5 CronJobs

- CronJob que, de minuto a minuto(alterável), verifica os leilões que terminarão e procede à respectiva actualização de estado:

```
BEGIN
```

```

SYS.DBMS_SCHEDULER.CREATE_JOB (
    job_name => '"CONNECTOR"."cron_acabaleilao"',
    job_type => 'STORED_PROCEDURE',
    job_action => '"CONNECTOR"."P_ACABALEILAO"',
    number_of_arguments => 0,
    start_date => NULL,
    repeat_interval => 'FREQ=MINUTELY',
    end_date => NULL,
    job_class => '"SYS"."DEFAULT_JOB_CLASS"',
    enabled => TRUE,
    auto_drop => FALSE,
    comments => 'Cron job que invoca procedure para fechar leilões após o seu término',
    credential_name => NULL,
    destination_name => NULL);
END;
/

```

Name	Value
1 OWNER	CONNECTOR
2 JOB_NAME	cron_acabaleilao
3 JOB_CLASS	DEFAULT_JOB_CLASS
4 COMMENTS	(null)
5 ENABLED	TRUE
6 CREDENTIAL_NAME	(null)
7 DESTINATION	(null)
8 PROGRAM_NAME	(null)
9 JOB_TYPE	STORED_PROCEDURE
10 JOB_ACTION	"CONNECTOR"."P_ACABALEILAO"
11 NUMBER_OF_ARGUMENTS	0
12 SCHEDULE_OWNER	(null)
13 SCHEDULE_NAME	(null)
14 START_DATE	13.01.28 12:25:49,400000000 EUROPE/LONDON
15 REPEAT_INTERVAL	FREQ=MINUTELY
16 END_DATE	(null)
17 EVENT_QUEUE_OWNER	(null)
18 EVENT_QUEUE_NAME	(null)
19 EVENT_QUEUE_AGENT	(null)
20 EVENT_CONDITION	(null)
21 FILE_WATCHER_OWNER	(null)
22 FILE_WATCHER_NAME	(null)

Figura 2: CronJob que verifica e actualiza o término dos leilões

#	LOG_ID	LOG_DATE	#	OWNER	JOB_NAME	#	JOB_SUBNAME	#	STATUS	#	ERROR#	#	REQ_START_DATE
1	4057	13.02.02 16:12:49,946000000 +0:00	CONNECTOR	cron_acabaleilao	(null)	SUCCEEDED	013.02.02 16:12:49,900000000	EUROPE/LON					
2	4011	13.02.02 02:24:49,742000000 +0:00	CONNECTOR	cron_acabaleilao	(null)	SUCCEEDED	013.02.02 02:24:49,700000000	EUROPE/LON					
3	4010	13.02.02 02:23:49,743000000 +0:00	CONNECTOR	cron_acabaleilao	(null)	SUCCEEDED	013.02.02 02:23:49,700000000	EUROPE/LON					
4	4009	13.02.02 02:22:49,743000000 +0:00	CONNECTOR	cron_acabaleilao	(null)	SUCCEEDED	013.02.02 02:22:49,700000000	EUROPE/LON					
5	4008	13.02.02 02:21:49,744000000 +0:00	CONNECTOR	cron_acabaleilao	(null)	SUCCEEDED	013.02.02 02:21:49,700000000	EUROPE/LON					
6	4007	13.02.02 02:20:49,722000000 +0:00	CONNECTOR	cron_acabaleilao	(null)	SUCCEEDED	013.02.02 02:20:49,700000000	EUROPE/LON					
7	4006	13.02.02 02:19:49,711000000 +0:00	CONNECTOR	cron_acabaleilao	(null)	SUCCEEDED	013.02.02 02:19:49,700000000	EUROPE/LON					
8	4005	13.02.02 02:18:49,724000000 +0:00	CONNECTOR	cron_acabaleilao	(null)	SUCCEEDED	013.02.02 02:18:49,700000000	EUROPE/LON					
9	4004	13.02.02 02:17:49,723000000 +0:00	CONNECTOR	cron_acabaleilao	(null)	SUCCEEDED	013.02.02 02:17:49,700000000	EUROPE/LON					
10	4003	13.02.02 02:16:49,657000000 +0:00	CONNECTOR	cron_acabaleilao	(null)	SUCCEEDED	013.02.02 02:16:49,600000000	EUROPE/LON					
11	4002	13.02.02 02:15:49,621000000 +0:00	CONNECTOR	cron_acabaleilao	(null)	SUCCEEDED	013.02.02 02:15:49,600000000	EUROPE/LON					
12	4001	13.02.02 02:14:49,618000000 +0:00	CONNECTOR	cron_acabaleilao	(null)	SUCCEEDED	013.02.02 02:14:49,600000000	EUROPE/LON					
13	4000	13.02.02 02:13:49,614000000 +0:00	CONNECTOR	cron_acabaleilao	(null)	SUCCEEDED	013.02.02 02:13:49,600000000	EUROPE/LON					
14	3999	13.02.02 02:12:49,611000000 +0:00	CONNECTOR	cron_acabaleilao	(null)	SUCCEEDED	013.02.02 02:12:49,600000000	EUROPE/LON					
15	3998	13.02.02 02:11:49,607000000 +0:00	CONNECTOR	cron_acabaleilao	(null)	SUCCEEDED	013.02.02 02:11:49,600000000	EUROPE/LON					

Figura 3: Últimas actividades do CronJob

5 Aplicação administrador

5.1 Linguagens e Ferramentas usadas na implementação

De forma a ser possível proceder à implementação do sistema de administração, foram consideradas várias linguagens orientadas para a web, das quais a escolha recaiu para o uso de PHP associado a HTML e CSS. No que diz respeito ao PHP, inicialmente considerou-se o uso de uma framework para automatizar o processo de desenvolvimento, mas essa hipótese acabou por ser posta de parte e o sistema foi criado de raiz usando a mesma linguagem. No que concerne à utilização e aspecto da página recorreu-se ao twitter bootstrap que é uma framework de front-end em HTML e CSS. Também se recorreu a Javascript para permitir um uso mais interactivo com o utilizador. Nomeadamente, ao processo de desenvolvimento, de forma a englobar o sistema de administração numa classe e usando o conceito de orientação aos objectos. Todo o funcionamento do sistema está englobado numa classe agregadora e é chamado um método construtor em cada um dos ficheiros da aplicação. Desta forma, é possível separar a modelação das funcionalidades do sistema e da sua interface com o utilizador.

5.2 Segurança

De forma a garantir a segurança da aplicação e do sistema foram feitas algumas opções relativas ao modelo de implementação do sistema. Relativamente ao sistema de login, existe encriptação ao nível da base dados através do algoritmo MD5 (o standard actual), que permite a encriptação de uma password mas não possibilita o processo oposto. Para além disso, a verificação da autenticação é feita ao nível da base de dados prevenindo, deste modo, que haja troca de informações relativas ao login e passwords no sentido da base de dados para a aplicação. Para além disso, e tendo em conta que esta aplicação não se destina ao utilizador normal, mas sim a um utilizador avançado com tarefas de administração, tornou-se óbvio que opções como “Lembrar password” ou “Lembrar login” tornam o sistema inseguro e dão aso a quebras de segurança. Desta forma, foi optado por não as incluir na modelação do sistema. Ainda dentro do sistema de login, após a autenticação são usadas variáveis de ‘session’ do PHP para gerir os utilizadores autenticados no sistema.

Sendo esta uma aplicação de administração, devendo primar pela segurança, optou-se então por não guardar informações de sessão na base de dados, sendo estas guardadas apenas temporariamente em formato cookies, obrigando assim a reautenticação após o fecho da aplicação por parte dos utilizadores. Outro cuidado relativo à segurança é a verificação em todos os pontos da aplicação de informações relativas a sessões, ou seja, só é possível qualquer tipo de interacção com a aplicação após feita a autenticação no sistema.

Relativamente à base de dados, foram tomadas precauções relativas a injecções de SQL nos campos de inserção, por parte de utilizadores, principalmente na zona de login, impedindo

assim autenticações não autorizadas no sistema. Outras medidas tomadas relativamente a funcionalidades como *Directory Listing* e Mensagens de Erro pelo servidor *Apache*, foi criado um ficheiro *.htaccess* onde são tomadas precauções em relação a estes tópicos.

5.3 Interface aplicação administrador

5.3.1 Login

Esta página permite ao administrador entrar no sistema, após inserir os dados necessários para tal. Como foi referido anteriormente foram tidos cuidados de segurança impedindo assim autenticações não autorizadas no sistema.

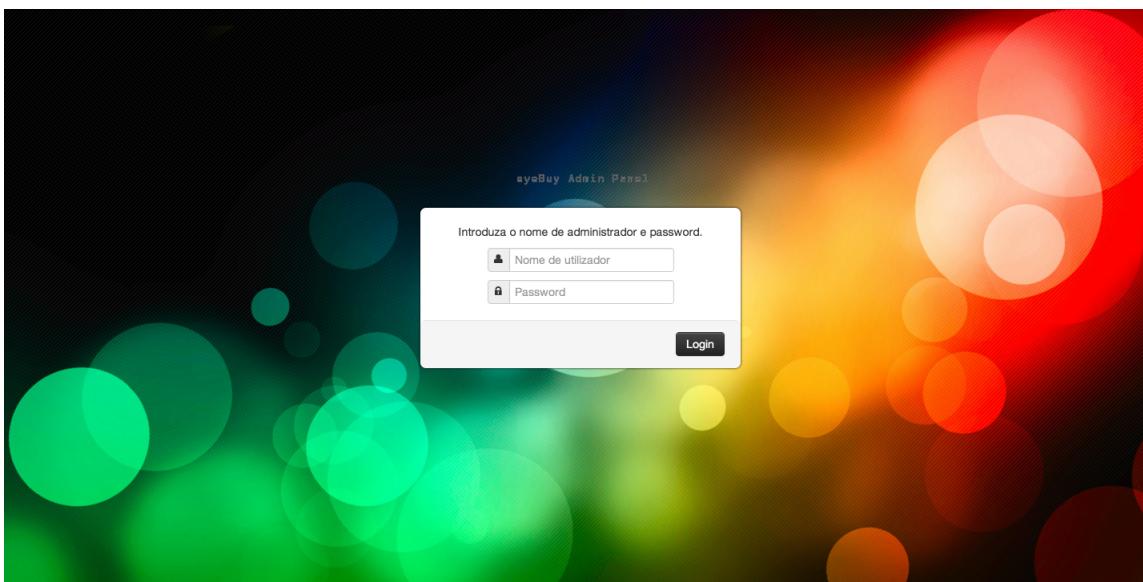


Figura 4: Página de login para um administrador

5.3.2 Dashboard

Depois de efectuado o login, o administrador tem como página inicial uma página onde pode consultar várias informações sobre o estado actual do sistema. Poderá ver o crescimento do sistema com estatísticas, como percentagem de novos utilizadores no último mês, novos anúncios, novas mensagens e novas transacções. Terá também acesso a informações relativas ao número de entidades totais do sistema, tais como utilizadores registados, anúncios colocados, mensagens enviadas, dinheiro transaccionado e número de administradores. É também disponibilizada informação relativa à disposição dos produtos pelas diversas categorias e pelas diferentes zonas do país como também é apresentado um gráfico que demonstra o número de registos nos últimos 5 meses. Por último, pode ver o *ranking* de melhores compradores, vendedores e os produtos mais desejados.

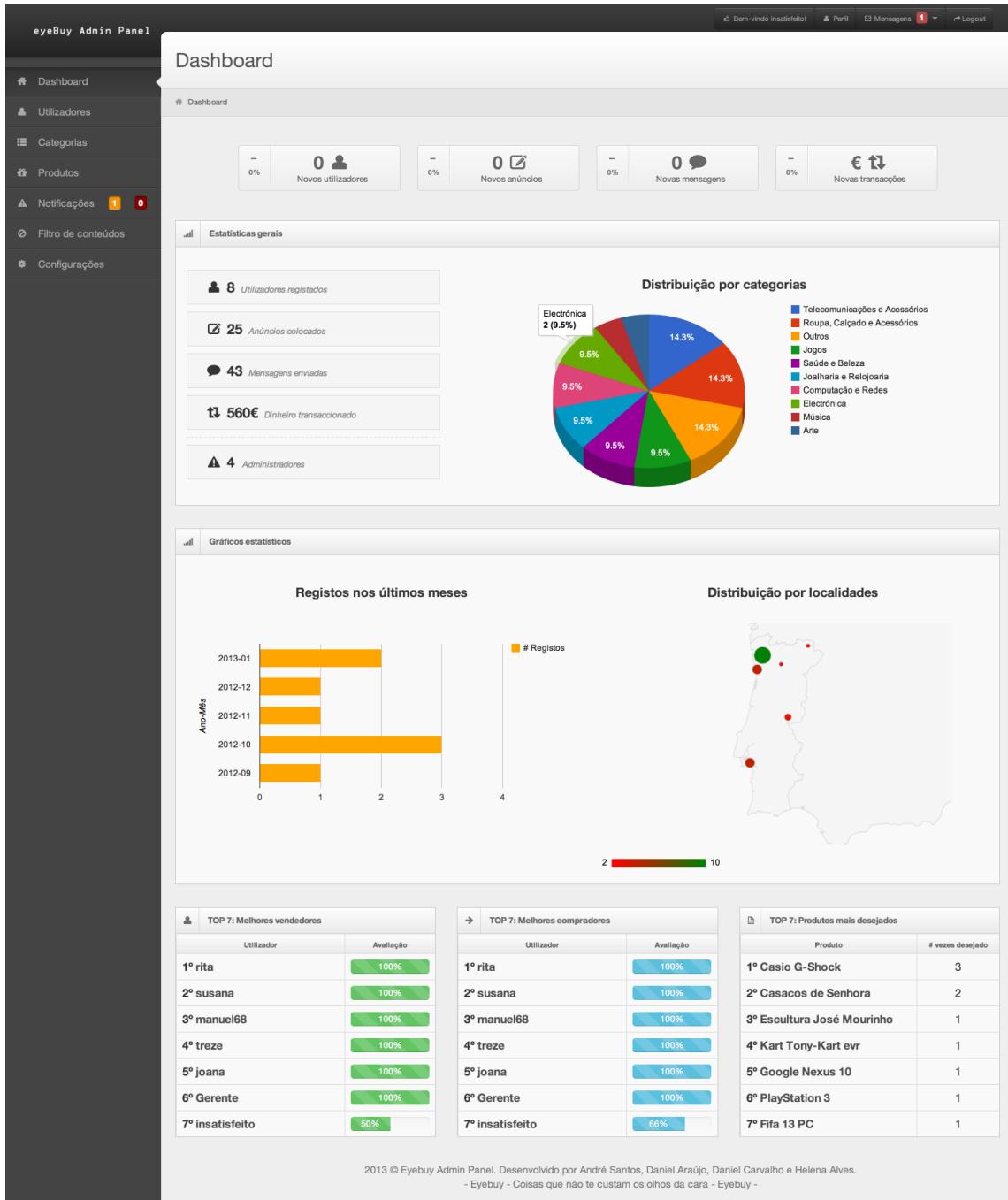


Figura 5: Página inicial para um administrador

5.3.3 Produtos

Quando se acede à página dos produtos, é possível encontrar uma listagem de todos os produtos que estão disponíveis no sistema, apresentados em páginas de tamanho editável. No início da página, encontram-se opções de pesquisa e filtragem de produtos que permitem uma melhor procura e selecção dos produtos desejados.

Na listagem de produtos podem ser encontradas todas as informações relevantes do mesmo. Também são disponíveis diferentes campos de pesquisa, como pode ser observado na imagem seguinte.

sysBuy Admin Panel

[Dashboard](#) [Utilizadores](#) [Categorias](#) [Produtos](#) [Notificações](#) [Filtro de conteúdos](#) [Configurações](#)

Lista de produtos

A procura devolveu 25 produtos! Show [10] entradas

Thumbnail	Nome do Produto	Categoria	Preço	Venda
	Asus N81jq	> Computação e Redes > Computadores	Preço: 500 €	Lelito
	Batons Avon	> Saúde e Beleza > Outros	Preço: 19 €	Venda directa
	Bolas	> Desporto > Bolas	Preço: 50 €	Venda directa
	Botas jeffrey campbell	> Roupas, Calçado e Acessórios > Sapatos	Preço: 330 €	Lelito
	Botas Timberland	> Roupas, Calçado e Acessórios > Sapatos	Preço: 70 €	Venda directa
	Canon S3	> Câmaras e Fotografia > Digitais	Preço: 230 €	Venda directa
	Casa de Férias Lagos	> Outros > Outros	Preço: 90000 €	Lelito
	Casacos de Senhora	> Roupas, Calçado e Acessórios > Casacos	Preço: 79 €	Venda directa
	Casio G-Shock	> Joalharia e Relojaria > Relógios	Preço: 43 €	Lelito
	Escultura José Mourinho	> Arte > Esculturas	Preço: 5600 €	Lelito

Search: First | Previous | **1** | 2 | 3 | Next | Last

Figura 6: Página de consulta e pesquisa de produtos

Se forem preenchidos diferentes campos na pesquisa, deixam de ser apresentados todos os

produtos, passando assim, a serem listados apenas os produtos que estão de acordo com a pesquisa.

The screenshot shows the eyeBuy Admin Panel interface. On the left, there is a dark sidebar with various menu items: Dashboard, Utilizadores, Categorias, Produtos (selected), Notificações (with 1 new notification), Filtro de conteúdos, and Configurações. The main content area has a title "Lista de produtos" and a breadcrumb navigation "Dashboard > Lista de produtos". Below this is a search bar labeled "Pesquisa de produtos" with a magnifying glass icon. There are several filter options: "Procurar:" (text input), "Categoria" (dropdown set to "Computação e Redes - Comput..."), "Pagamento" (dropdown), "Método de venda" (dropdown set to "Leilão", with "Venda directa" and "Vendido" as other options), "Vendido por" (dropdown set to "Gerente"), "Localidade" (dropdown set to "Braga"), "Vendido a:" (text input), and "Preço entre: 400 e 1000" (checkboxes). A blue "Procurar" button is located below the filters. The results section starts with a header "Resultado da pesquisa" and a message "> A procura devolveu 1 produtos!". It lists one item: "Asus N61jq" (image, category "Computação e Redes > Computadores", price "500 €", status "Leilão"). Below the item are details: "Estado: Usado", "Colocado em: 27-09-2012", "Quantidade: 1", and "Localidade: Braga". At the bottom of the results page are "Search:" and pagination controls ("First", "Previous", "1", "Next", "Last").

Figura 7: Exemplo de uma procura

5.3.4 Consultar produto

Quando é seleccionado determinado produto, é apresentada uma página com todas as informações desse mesmo produto, assim como as funcionalidades “editar” e “apagar”. Para além disto, foram adicionadas etiquetas com o intuito de fornecer informação sobre o estado do produto (Apagado ou Activo) e ainda sobre seu método de venda.

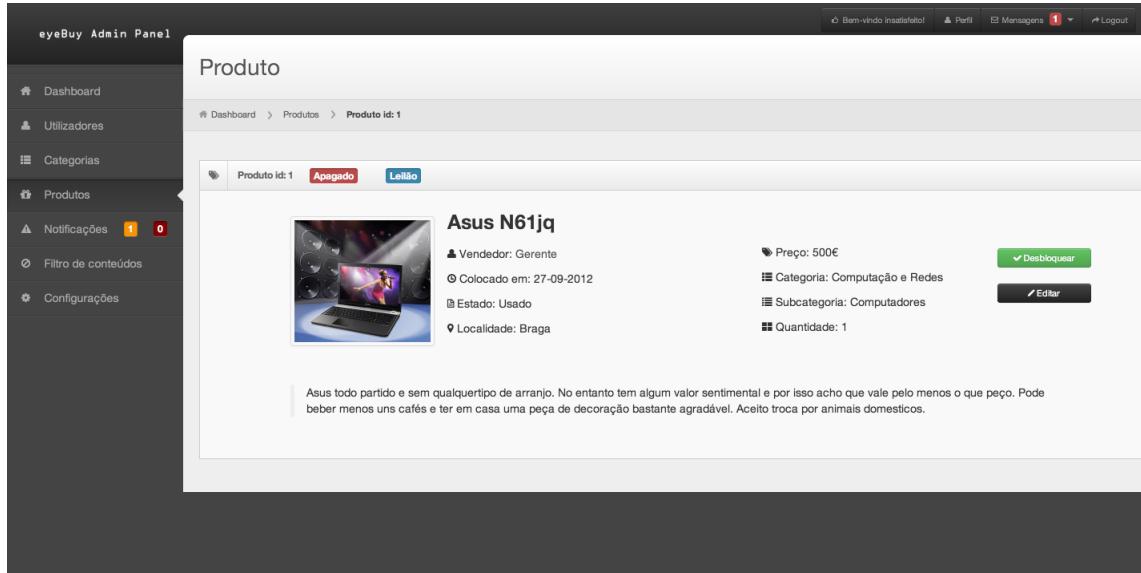


Figura 8: Página de um produto

5.3.5 Editar produto

De seguida, é apresentada a página de edição de um produto onde é possível a edição de todas as informações de um produto por parte do administrador.

eyeBuy Admin Panel

Bem-vindo insatisfeito! Perfil Mensagens 1 Logoff

Dashboard Utilizadores Categorias Produtos Notificações 1 0 Filtro de conteúdos Configurações

Editar Produto

Dashboard > Produtos > Produto id: 1

Produto id: 1 Apagado Leilão

Asus N61jq

Vendedor: Gerente

Colocado em: 27-09-2012

Estado: Usado

Categoria: Computação e Redes - Computadores

Método venda: Leilão

Preço: 500

Quantidade: 1

Asus todo partido e sem qualquertipo de arranjo.
No entanto tem algum valor sentimental e por isso acho que vale pelo menos o que peço.

Guardar Voltar ao produto

Figura 9: Página de edição um produto

5.3.6 Notificações

Uma das funções mais importantes de um administrador passa por manter os clientes agradados. Como tal, foi implementado um sistema de mensagens para utilizadores interagirem com os administradores, dando a possibilidade de os utilizadores poderem reportar comportamentos anormais por parte do sistema, reportarem conteúdo indevido, reportarem compra/venda falhada ou qualquer outro assunto.

Já que o sistema irá ser administrado por vários utilizadores, foi necessário pensar na forma como serão tratadas as respostas às diferentes mensagens. Inicialmente, foram separados os reports das mensagens pessoais por forma a distinguir o comportamento do administrador perante a mensagem. Quando o utilizador abre uma notificação essa é marcada como em “Tratamento” (para informar os outros administradores e assim evitar que mais que um administrador resolva o mesmo problema), até que este a marque como “Resolvida”. As mensagens, que ainda não tenham sido lidas por nenhum administrador, são marcadas como novas ou a aguardar tratamento.

The screenshot shows the 'Notificações' (Notifications) page within the 'eyeBuy Admin Panel'. The left sidebar includes links for Dashboard, Utilizadores, Categorias, Produtos, Notificações (with two notifications marked), Filtro de conteúdos, and Configurações. The main content area is titled 'Notificações' and shows a list of messages under 'Caixa de entrada'. The table has columns for 'Assunto', 'Enviado por', 'Enviada em', and 'Oções'. The first message is 'Novo Bolas vazias' from 'insatisfeito' on '28 Jan 2013 às 00:31' with 'Responder' and 'Resolvido' buttons. The second message is 'Resolvida Livro PHP' from 'insatisfeito' on '28 Jan 2013 às 00:30' with 'Responder' and 'Reabrir' buttons. The third message is 'Em tratamento Mota desaparecida' from 'Gerente' on '27 Nov 2012 às 20:24' with 'Responder' and 'Resolvido' buttons. A search bar and pagination controls ('First', 'Previous', '1', 'Next', 'Last') are at the bottom of the table. Top navigation includes 'Bem-vindo insatisfeito!', 'Perfil', 'Mensagens 1', and 'Logout'.

Figura 10: Página de notificações

5.3.7 Caixa de entrada e saída

Estas páginas permitem ao administrador consultar a lista de mensagens pessoais recebidas e enviadas. Nestas páginas, o administrador pode abrir uma mensagem, abrir a página de nova mensagem, responder a uma mensagem recebida, apagar uma mensagem e marcar todas como lidas. De notar ainda, que as mensagens são listadas por páginas e que é possível alterar o número de entradas por páginas.

Para facilitar a leitura, foram adicionadas etiquetas que distinguem mensagens lidas de mensagens por ler.

Figura 11: Página de consulta das mensagem recebidas

Figura 12: Página de consulta das mensagem enviadas

5.3.8 Nova Mensagem

Esta página permite ao administrador enviar uma nova mensagem a determinado utilizador. De notar que é possível enviar uma mensagem a mais do que um utilizador como

pode ser verificado na imagem que se segue.

De notar ainda que na introdução do destinatário é sugerido o nome de utilizador conforme vão sendo escritos os nomes parcialmente.

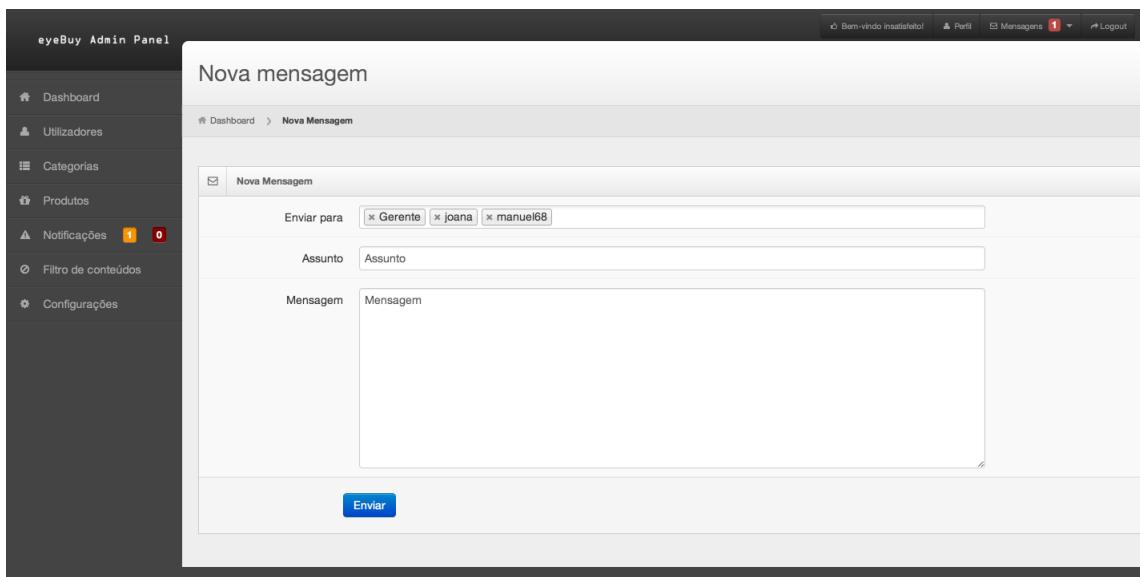


Figura 13: Página de nova mensagem de um administrador

5.3.9 Consultar Mensagens

Existem duas interfaces distintas para consultar mensagem, a interface de consulta de uma mensagem recebida e de uma mensagem enviada. Na interface de consulta de uma mensagem enviada é possível encontrar várias funcionalidades, tais como encaminhar, mandar outra mensagem, e apagar a mensagem enviada. Relativamente às mensagens recebidas é possível encontrar funcionalidades como responder, marcar como não lida e apagar a mensagem.

localhost/eyeBuy/admin/mymessage.php?id=252#collapseTwo

Figura 14: Página de consulta de uma mensagem enviada

localhost/eyeBuy/admin/mymessage.php?id=252#collapseTwo

Figura 15: Página de consulta de uma mensagem recebida

5.3.10 Consultar utilizadores

Esta página permite ao administrador consultar a lista de utilizadores na base de dados do sistema. Nesta página o administrador pode consultar o perfil do utilizador, editar o perfil do mesmo e até banir o utilizador do sistema.

Nesta página os utilizadores são apresentados por páginas com um número de entradas editável. Podemos verificar na imagem seguinte, que na listagem de utilizadores podem ser consultadas informações como nome, email, classificação como comprador e classificação como vendedor.

De notar ainda, que foram adicionadas etiquetas que permitem distinguir a situação dos utilizadores, como “Banido” ou “Administrador”.

Utilizador	Nome	E-mail	Comprador	Vendedor	Opções
Gerente Admin	André Santos	andrecdr@gmail.com	<div style="width: 100%;">100</div>	<div style="width: 100%;">100</div>	
insatisfeito Admin	Daniel Carvalho	dapcarvalho@sreng.urn	<div style="width: 66%;">66</div>	<div style="width: 50%;">50</div>	
joana	Joana Dias	joana@gmail.com	<div style="width: 100%;">100</div>	<div style="width: 100%;">100</div>	
leninha Banido	Helena Alves	helenalves4@hotmail.com	<div style="width: 50%;">50</div>	<div style="width: 100%;">100</div>	
manuel68	Manuel Santos	manuel@gmail.com	<div style="width: 100%;">100</div>	<div style="width: 100%;">100</div>	
rita	Rita Andrade	rita@gmail.com	<div style="width: 100%;">100</div>	<div style="width: 100%;">100</div>	
susana	Susana Freitas	susana@gmail.com	<div style="width: 100%;">100</div>	<div style="width: 100%;">100</div>	
treze Admin	Daniel Araújo	daniel.ara.6@gmail.com	<div style="width: 100%;">100</div>	<div style="width: 100%;">100</div>	

Figura 16: Página de consulta dos utilizadores do sistema

5.3.11 Consultar perfil utilizador

Esta página permite ao administrador ver o perfil completo de um utilizador do sistema. Nesta página o administrador pode consultar os dados do perfil, as estatísticas como vendedor e comprador e o conteúdo das suas listas de produtos. Para além da consulta, o administrador pode utilizar as diferentes funcionalidades como o envio de mensagens, edição do perfil, tornar o utilizador administrador ou até mesmo banir o utilizador do sistema.

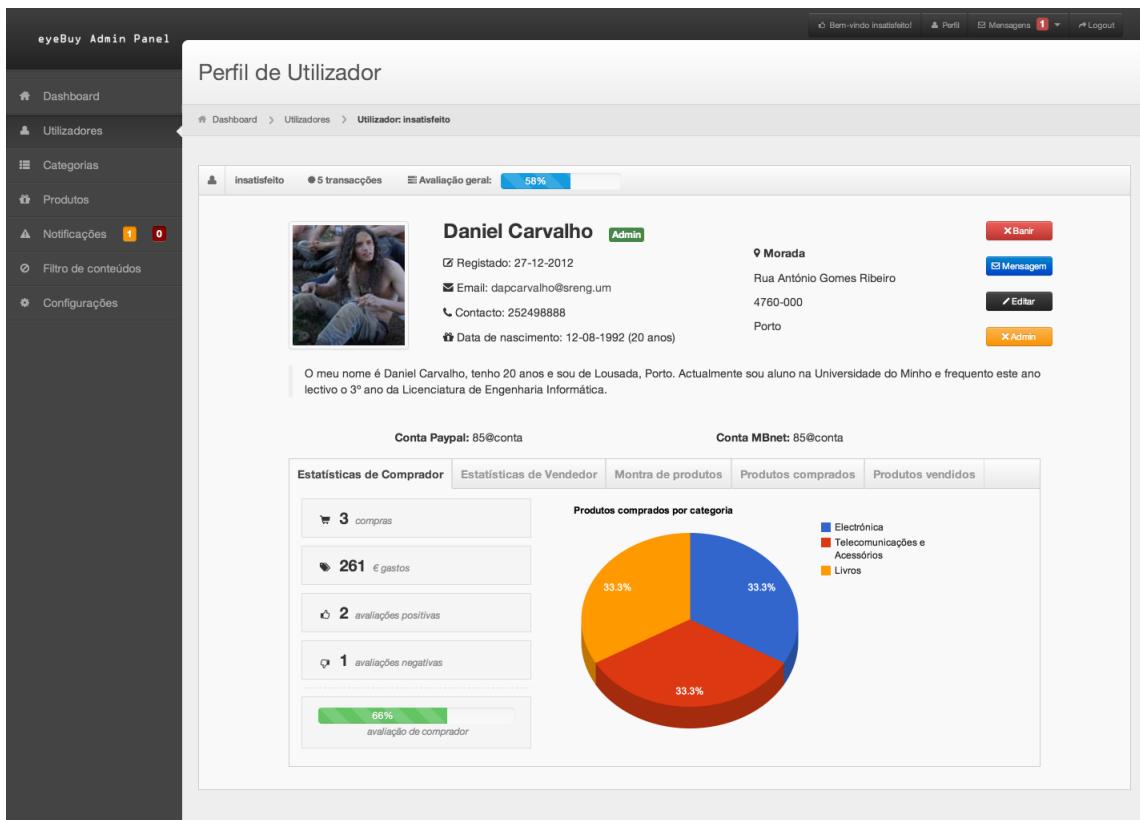


Figura 17: Página de consulta de um perfil de um utilizador

É possível verificar que foram colocadas no perfil algumas estatísticas relevantes do utilizador como comprador e como vendedor:



Figura 18: Estatísticas como vendedor disponíveis na consulta de um perfil de um utilizador

Podemos encontrar também no perfil de um utilizador, todos os produtos que ele está actualmente a vender, como pode ser verificado na imagem seguinte:

Estatísticas de Comprador	Estatísticas de Vendedor	Montra de produtos	Produtos comprados	Produtos vendidos
Botas Timberland	Vendo botas Timberland.	> Roupa, Calçado e Acessórios > Sapatos	Preço: 70 €	
	Estado: Novo	Colocado em: 27-11-2012		Venda directa
	Quantidade: 3	Localidade: Porto		
Nokia 3310	Vendo dois Nokias 3310	> Telecomunicações e Acessórios > Outros	Preço: 12 €	
	Estado: Usado	Colocado em: 27-11-2012		Venda directa
	Quantidade: 2	Localidade: Porto		

Figura 19: Produtos na montra disponíveis na consulta de um perfil de um utilizador

É também possível visualizar todos os produtos que o utilizador comprou e vendeu como pode ser verificado nas imagens seguintes.

Estatísticas de Comprador	Estatísticas de Vendedor	Montra de produtos	Produtos comprados	Produtos vendidos	
			<p>◆ Power TV Toshiba TV nova para venda.</p> <p>● Estado: Usado 📍 Localidade: Setúbal</p>	<p>> Electrónica > TVs</p> <p>Preço: 190 € Venda directa</p>	
			<p>◆ Optimus Boston Vendo Optimus Boston em excelente estado. Ainda com um mês de garantia.</p> <p>● Estado: Usado 📍 Localidade: Castelo Branco</p>	<p>> Telecomunicações e Acessórios > Smartphones</p> <p>Preço: 50 € Venda directa</p>	
			<p>◆ PHP a Bíblia Tradução da 2º edição Tim Converse Joyce Pack</p> <p>● Estado: Usado 📍 Localidade: Braga</p>	<p>> Livros > Manuais Escolares</p> <p>Preço: 21 € Venda directa</p>	

Figura 20: Produtos comprados disponíveis na consulta de um perfil de um utilizador

Estatísticas de Comprador	Estatísticas de Vendedor	Montra de produtos	Produtos comprados	Produtos vendidos	
			<p>◆ Bolas São 300 bolas que vendo em conjunto. O preço é o indicado em cima ou mais.</p> <p>● Estado: Novo 📍 Localidade: Porto</p>	<p>> Desporto > Bolas</p> <p>Preço: 50 € Venda directa</p>	
			<p>◆ Batons Avon Vendo batons avon por preços muito competitivos.</p> <p>● Estado: Novo 📍 Localidade: Porto</p>	<p>> Saúde e Beleza > Outros</p> <p>Preço: 19 € Venda directa</p>	

Figura 21: Produtos vendidos disponíveis na consulta de um perfil de um utilizador

5.3.12 Editar perfil utilizador

Esta página permite ao administrador editar todos os dados do perfil de um determinado utilizador.

eyeBuy Admin Panel

Bem-vindo (Administrador) Perfil Mensagens 1 Logout

Dashboard Utilizadores Categorias Produtos Notificações 2 0 Filtro de conteúdos Configurações

Editar Utilizador

Dashboard > Utilizadores > Utilizador: Gerente

Gerente 0 transacções Avaliação geral: 100%

André Santos

Registado: 27-10-2012

Email: andreccdr@gmail.com

Contacto: 919200815

Data de nascimento: 01-11-1992

Escolher ficheiro Nenhum ficheiro selecionado

Morada

Enderço: Rua S. João 103

Código postal: 4760-777

Distrito: Braga

Descrição

Conta Paypal: andreccdr@gmail.com Conta MBnet: andreccdr@gmail.com

Password: Confirmar Password:

Guardar Voltar ao perfil

Figura 22: Página de edição de um perfil de um utilizador

5.3.13 Categorias

Esta página permite ao utilizador consultar todas as categorias de produtos do sistema, editar, apagar, ver estatísticas das categorias e até mesmo adicionar uma nova categoria ao sistema.

eyeBuy Admin Panel

Bem-vindo Inasfisfot! Perfil Mensagens 1 ▾ Logout

Categorias

Criar nova categoria

Nome: Guardar

Nome	Subcategorias	Número de produtos	Percentagem de utilização	Opções
Antiguidades	4	1	4%	
Arte	5	1	4%	
Colecionáveis	4	0	0%	
Computação e Redes	8	2	8%	
Câmaras e Fotografia	5	1	4%	
Desporto	5	1	4%	
Electrónica	6	2	8%	
Indústria	1	0	0%	
Joalharia e Relojoaria	3	2	8%	
Jogos	4	2	8%	

Show 10 entries

Search:

First Previous 1 2 Next Last

Figura 23: Página de consulta e gestão de categorias

5.3.14 Subcategorias

Esta página permite ao utilizador consultar todas as subcategorias de produtos do sistema, editar, apagar, ver estatísticas das subcategorias e até mesmo adicionar uma nova subcategoria ao sistema.

The screenshot shows the eyeBuy Admin Panel interface. On the left is a dark sidebar with navigation links: Dashboard, Utilizadores, Categorias (selected), Produtos, Notificações (with 1 new), Filtro de conteúdos, and Configurações. The main content area has a header "Subcategorias" and a breadcrumb "Dashboard > Categorias > Subcategorias de Antiguidades". Below this is a form to "Criar nova subcategoria" with a "Nome" input field and a "Guardar" button. The main table lists categories:

Nome	Número de produtos	Percentagem de utilização	Opções
Manuscritos	0	0%	[Edit, Checkmark, Delete]
Mobiliário	1	100%	[Edit, Checkmark, Delete]
Outros	0	0%	[Edit, Checkmark, Delete]
Pinturas	0	0%	[Edit, Checkmark, Delete]

At the bottom are "Search:" and pagination controls: First, Previous, 1, Next, Last.

Figura 24: Página de consulta e gestão de subcategorias

5.3.15 Filtro de conteúdos

Uma das funcionalidades que o administrador tem acesso é o filtro de conteúdos, onde o utilizador pode adicionar e remover palavras que quer procurar no conteúdo de algum anúncio. Esta funcionalidade é importantíssima pois permite que o administrador controle mais facilmente o conteúdo indevido no sistema.

Na página respectiva, o utilizador para além de poder consultar, remover e adicionar palavras, pode também ver os produtos que estão a utilizar alguma das palavras procuradas.

eyeBuy Admin Panel

Bem-vindo Insatisfat! Perfil Mensagens Logout

Filtro de Conteúdos

Dashboard > Filtro de Conteúdos

Sucesso! A(s) palavra(s) Marijuana servem agora de filtro para os produtos à venda.

Criar nova restrição

Palavra(s) Adicionar

Lista de Filtro de Conteúdos

Asus X Drogas X Marijuana X Pudim X

Lista de Produtos que violam as restrições

	Asus N61jq	> Computação e Redes > Computadores	Preço: 500 €
	Asus todo partido e sem qualquertipo de arranjo. No entanto tem algum valor sentimental e por isso acho que vale pelo menos o que peço. Pode beber menos uns cafés e ter em casa uma peça de decoração bastante agradável. Aceito troca por animais domesticos.	Estado: Usado	Colocado em: 27-09-2012
		Quantidade: 1	Localidade: Braga
	Pudim de ovos	> Outros > Outros	Preço: 15 €
	Pudim de ovos overecido pela minha avó. Ainda não provei mas como costuma ser duro posso vender á melhor proposta. Por favor comprem, não quero deitar fora comida.	Estado: Novo	Colocado em: 27-09-2012
		Quantidade: 1	Localidade: Braga

Figura 25: Página de filtro de conteúdo

5.3.16 Configurações

Uma das tarefas importantes de um administrador passa pela gestão da base de dados. Uma das páginas que o administrador tem acesso é a página de configuração da base de dados, onde podem ser alterados os seguintes campos:

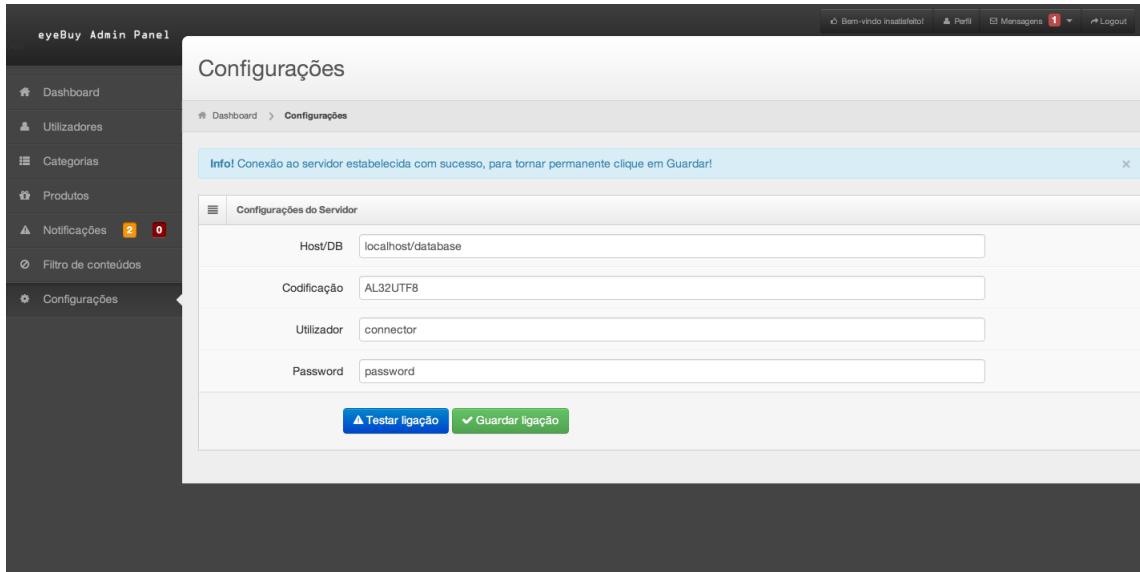


Figura 26: Página de configurações

6 Aplicação cliente

Para o desenvolvimento da aplicação cliente, para conseguir uma melhor consiliação com o projecto de Desenvolvimento de Sistemas de Software foi utilizada a linguagem de programação orientada aos objectos Java. Toda a aplicação foi desenvolvida aplicando metodologias lecionadas na cadeira de DSS, tais como Model View Controller e Modelo das três camadas.

Para fazer a comunicação entre a aplicação cliente (desenvolvida em Java) usamos a API de acesso a bases de dados JDBC (Java DataBase Connectivity) aqui são disponibilizados métodos que permitem estabelecer ligação à BD, a execução de queries (Select da linguagem SQL) e de instruções que permitem a remoção, actualização e inserção de dados (delete, update e insert respectivamente na SQL), entre outros. Neste modelo de funcionamento são criadas strings que contêm como texto as instruções a executar na base de dados, assim é possível inserir as variáveis Java nas instruções SQL, por exemplo imagens (recorrendo a PreparedStatements), lidando com um nível de abstracção maior.

6.1 Camada ORM

Uma vez que o desenvolvimento ocorreu em paralelo com o projecto pedido na Unidade Curricular de DSS foram feitas algumas escolhas no sentido de permitir que fossem apresentados produtos funcionais nomeadamente um programa cliente - em Java, para a UC de DSS - e uma aplicação para correr no browser para administrar o sistema - um site que basicamente administra a base de dados de utilizadores. A nota anterior é pertinente porque permite explicar alguma decisões, relacionadas com o desenvolvimento da aplicação, especialmente da camada ORM. Para desenvolver a camada ORM, e uma vez que o sistema tinha sido modelado basicamente (modelo de domínio, diagramas de usecase, diagramas de sequência) aquando da entrega da primeira etapa, a escolha reincidiu em derivar as tabelas da base de dados a partir dos objectos presentes no diagrama de classes original. Após isso foram criados objectos DAO (Data Access Objects), um para cada objecto usado na aplicação. Estes objectos servem para abstrair a camada de persistência, devendo, por isso, conter métodos que permitam a interacção com o Base de Dados. Na realidade, devido à circunstância referida no início desta secção, os métodos contidos nas classes permitem fazer muito mais que as operações de CRUD (create, remove, update, delete) de um objecto. São disponibilizadas na Base de Dados procedimentos que permitem executar conjuntos de operações que dizem respeito a uma acção bem definida no sistema (por exemplo comprar tudo o que se encontra no carrinho); isto permite garantir atomicidade embora ponha parte da camada de negócios na base de dados.

6.2 Decisões relevantes nas listagens de produtos

Na página inicial optamos por mostrar ao utilizador uma lista de produtos que lhe podem interessar, essa lista é constituída por todos os produtos que o utilizador pode comprar ordenados por aqueles que achamos que podem ter mais interesse para ele. O critério de ordenação que utilizamos foi as pesquisas com mais frequência que esse mesmo utilizador faz, ou melhor, irão ser mostrados em primeiro lugar os produtos que têm um maior numero de coincidências com as palavras mais pesquisadas do utilizador. Isto acrescenta valor ao sistema porque aumenta a probabilidade de haver mais negócios realizados.

O mesmo acontece com a lista de produtos relacionados mostrada quando abrimos cada produto que procura, é mostrado ao utilizador uma lista de todos os produtos ordenados pelas coincidências com o nome do produto que ele está a ver. Quando o utilizador abre um produto também lhe é mostrado uma lista de possíveis trocas que pode propor ao vendedor. Essa lista de trocas contém todas as possíveis relações de troca entre aquele produto que está a visitar e a sua montra de produtos.

6.3 Implementação do Modelo das Três camadas (Interface, Camada de Negócio, Camada de Persistência)

Devido ao facto de aquando do inicio do trabalho ainda não termos perfeita noção da forma de implementar o modelo das três camadas, começamos a criação da aplicação usando uma abordagem que não terá sido a melhor. De facto, em grande parte da aplicação são chamados métodos das classes DAO que contém parte do modelo de negócio.

Entretanto, obtivemos conhecimento e consciência da importância e de quanto seria simplificada a divisão de trabalhos entre os elementos do grupo caso fosse implementado. Como a pressão do tempo não permitiu uma alteração para este modelo, decidimos mostrar que conseguímos implementar (visto que na modelação já estava previsto) alguma parte do projecto com o modelo das 3 camadas. Assim em anexo vão as classes que contêm a parte do sistema que lida com as mensagens. Aqui os métodos das classes do Modelo (Utilizador e Mensagem), são chamados na GUI. Por sua vez nas classes do modelo são chamados métodos que fazem a ligação à base de dados contidos nas classes DAO (UtilizadorDAO e MensagensDAO).

6.4 Interface: Experiência de utilização

Tentamos desde o inicio construir uma interface que oferecesse ao utilizador uma boa experiência de utilização. Para isso tomamos as seguintes decisões:

Imagens:

Decidimos listar os produtos no nosso sistema com imagens de tamanho médio, para promover o interesse do utilizador e incentivar a que este abra o produto. As imagens são mais informativas em relação ao produto em questão e tornam também a procura e navegação menos cansativa e mais agradável.

Botões:

- **Icones:** No nosso sistema a maior parte dos botões limitam-se a ícones, esta decisão tem como objectivo obter uma interface mais limpa, apresentando assim em alternativa a botões de texto que são por si mais espaçosos, pequenos botões intuitivos a cada acção.
- **Sugestão:** Tentamos em todos os botões colocar ícones sugestivos da acção, no entanto para facilitar ainda mais a acção do utilizador, nos botões que pudessem suscitar mais duvidas colocamos uma pequena sugestão que aparece quando colocado o rato por cima do botão, esta sugestão mostra explicitamente a acção do botão.
- **Botões de navegação** Em todas as navegações entre listas foram colocados 3 botões de navegação que facilitam essa mesma navegação, o botão próximo, o anterior e o actualizar. De notar que foram tidos cuidados para que esses mesmos botões fossem coerentes e por tanto em listagens de texto como de imagens utilizamos o mesmo sistema.

Agregação de elementos relacionados:

Tivemos o cuidado na construção das nossas interfaces de agrupar por elementos relacionados, para permitir que quando o utilizador executa determinada acção possa encontrar acções relacionadas na mesma interface e por isso muitas vezes evitar mudar de janela. Podemos assim encontrar as mensagens no perfil do utilizador, um histórico de ultimas vendas quando o utilizador visita a sua página de vendas, as listas de produtos seguidos e relacionados na página inicial e por isso a página onde normalmente o utilizador vê produtos.

Take a Tour:

No sistema disponibilizamos também ao utilizador um video que demonstra todo o funcionamento do sistema. Com este video pretendemos mostrar ao utilizador todas as funcionalidades do programa, incentivar a uma utilização completa e incentivar ao registo.

Link video: <https://dl.dropbox.com/u/1015358/takeatour.avi>

6.5 Segurança

No sentido de criar um produto que mesmo sendo implementado num âmbito académico estivesse minimamente preparado para o mundo real, foram tidos alguns cuidados no sentido de garantir a segurança e estabilidade tanto da aplicação como dos seus utilizadores. O uso de um sistema de login torna-se fundamental neste tipo de aplicação, pois é necessária a obtenção da identidade de cada um dos utilizadores para o conceito da aplicação poder funcionar. Tendo em conta que esta aplicação funciona num regime fechado, ou seja, é necessário o registo e login para ser possível o acesso às funcionalidades da aplicação o que torna necessário alguns cuidados na sua implementação. Um desses cuidados foi na encriptação da password na base de dados através do algoritmo MD5 que torna possível a encriptação no sentido texto -*à* password mas não o processo contrário, garantido assim um acréscimo à segurança da aplicação e dos seus utilizadores para além de que a verificação do login é feito na base de dados, ou seja não há tráfego no sentido da base de dados para a aplicação, nem tarefas de autenticação feitas directamente na aplicação. Outra funcionalidade é descrita pela interacção dos utilizadores com a administração do sistema no sentido de resolver problemas entre utilizadores onde a interacção utilizador-utilizador falhou. Assim sendo é possível notificar a administração de forma a obter resolução de problemas. Apesar de ter sido tomada em consideração a possibilidade de falhas de segurança através de injecções de SQL nos campos de inserção de dados na aplicação por parte dos utilizadores tornou-se também evidente que seria um incómodo reduzir o leque de caracteres a utilizar pelos utilizadores como plicas e barras (método de protecção contra sql injections através da remoção de alguns caracteres da sintaxe da linguagem) pelo que assim se ignorou este problema visto não ser um ponto muito relevante para o projecto.

6.6 Interface aplicação cliente

6.6.1 Login

Na página login do sistema, é onde acontece a primeira familiarização do utilizador com o sistema. Esta página permite ao utilizador entrar no sistema, após inserir os dados necessários para tal. Aqui, também é possível aos utilizadores não registados no sistema fazer o seu próprio registo. Caso algum utilizador precise de ajuda para alguma situação, é possível obtê-la apenas clicando no botão *Ajuda*, ou caso necessite de alguma informação sobre o sistema basta clicar no botão *Sobre o EyeBuy*. Existe também, nesta página, um vídeo promocional ao sistema (<https://dl.dropbox.com/u/13351850/0001.mpg>).

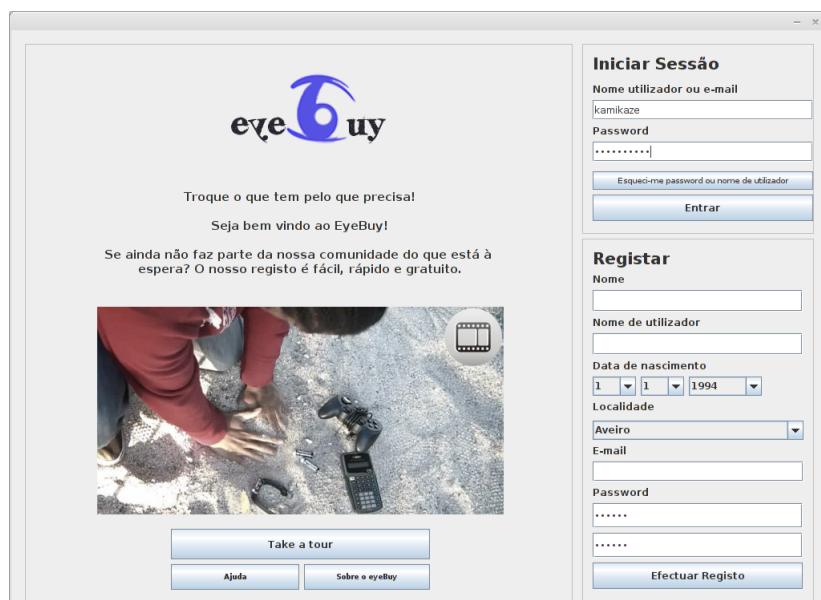


Figura 27: Página de login para um utilizador

6.6.2 Página inicial

Na página home do sistema são apresentadas várias listas de produtos, nomeadamente, lista de novos produtos, de produtos que podem interessar ao utilizador, de produtos dos melhores vendedores, de produtos desejados e seguidos. A existência de várias listas permite ao utilizador uma pesquisa mais eficiente e facilitada. Relativamente aos produtos, cada produto é representado pelo seu nome, preço e por uma imagem. Foi optado por colocar uma imagem junto ao produto ao invés de apenas listar os produtos com nome e preço, pois assim torna o produto mais cativante, aumentando o interesse do utilizador pelo mesmo. Enquanto que o uso de listas com apenas nome e preço ia tornar o programa chato e os produtos menos atractivos.

Como não poderia deixar de ser, e estando na página principal do sistema, é possível pesquisar produtos através do nome, categoria e localidade ou optar pela pesquisa avançada. De notar, que é possível pesquisar diferentes produtos em todas as páginas do sistema, facilitando assim a procura e a navegação no sistema.

A partir desta página é possível ir para outras páginas, como o perfil de utilizador, página de vendas, histórico, carrinho de compras, etc.

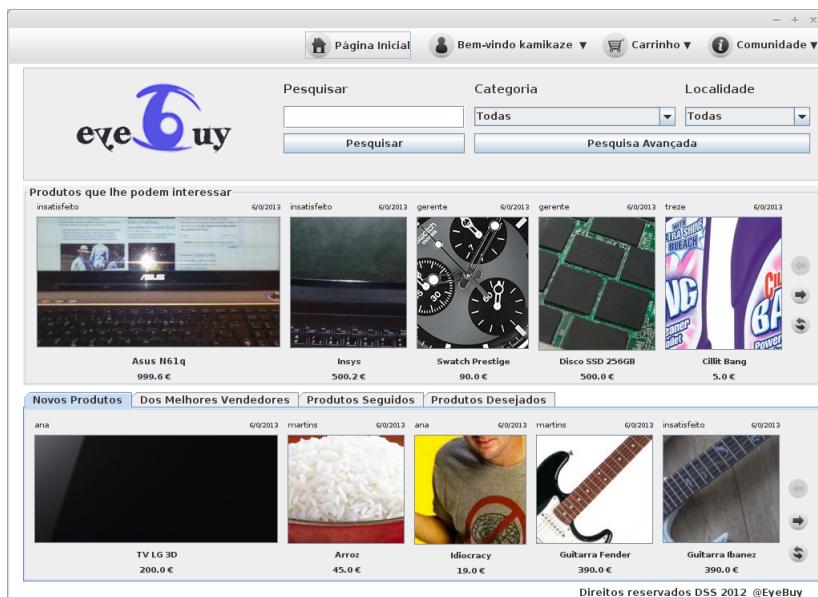


Figura 28: Página inicial do sistema

6.6.3 Pesquisar um produto

Após o utilizador escolher o nome do produto que quer pesquisar e escolher a sua categoria e localidade, é apresentada uma página com uma lista de produtos relacionados com a preferência anteriormente imposta pelo utilizador. Nesta página também é apresentado um

sub-menu de pesquisa, onde o utilizador pode procurar produtos por categoria, sub-categoria, estado, localidade e preço. Este novo sub-menu proporciona ao utilizador uma pesquisa muito mais facilitada e eficiente. A partir desta página é possível ir para outras páginas, como o perfil de utilizador, página de vendas, histórico, carrinho de compras, etc.

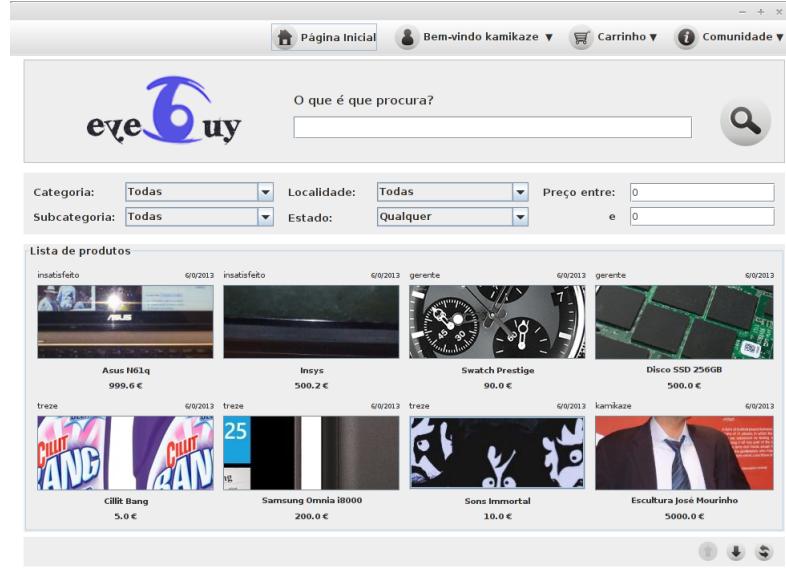


Figura 29: Página pesquisar produto

6.6.4 Pesquisa avançada

Após seleccionar pesquisa avançada, o utilizador pode procurar um produto através do seu nome, categoria, subcategoria, localidade, estado, método de pagamento, método de venda, quantidade, número de imagens, nome do vendedor, classificação do vendedor, e o preço. Esta pesquisa fornece ao utilizador uma lista de produtos com as características descritas pelo utilizador. A partir desta página é possível ir para outras páginas, como o perfil de utilizador, página de vendas, histórico, carrinho de compras, etc.



Figura 30: Página pesquisa avançada

6.6.5 Ver produto

Relativamente aos produtos, cada página de um dado produto apresenta todas as informações deste, como o seu preço, o seu estado, uma pequena descrição, a categoria e subcategoria a que pertencem, a data de colocação à venda e o seu respectivo vendedor. É possível ver o perfil de utilizador do vendedor do produto, clicando apenas no nome deste, tornando assim mais fácil a comunicação entre os utilizadores, no caso de haver interesse por parte de algum.

Para o total interesse do utilizador, também é apresentada uma lista de produtos relacionados e outra lista de sugestões para troca. A partir desta página é possível pesquisar produtos e ir para outras páginas, como o perfil de utilizador, página de vendas, histórico, carrinho de compras, etc.

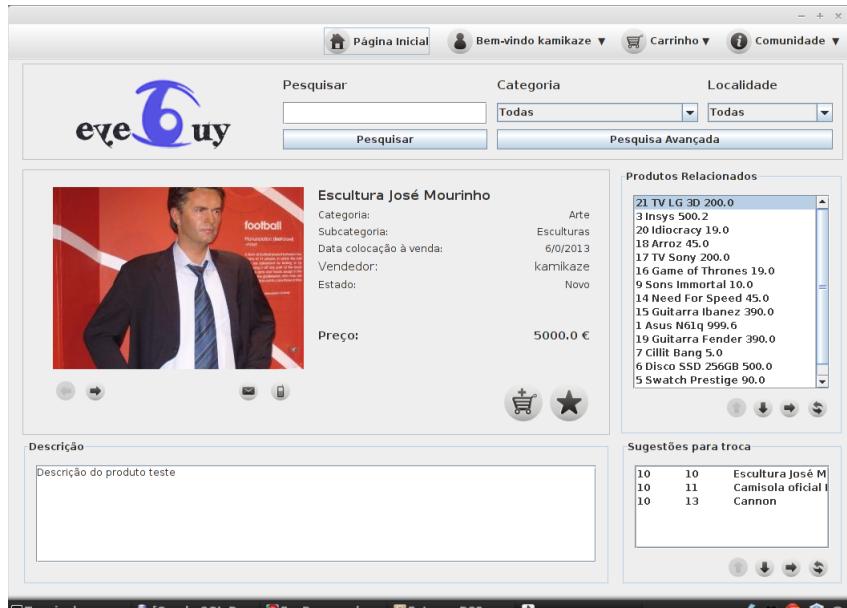


Figura 31: Página de um dado produto

6.6.6 Ver produto em leilão

Relativamente à página relacionada com os produtos que estão em leilão, esta apresenta o mesmo aspecto que a página *Ver produto*, excepto em algumas situações. Como o produto está a ser leiloado, é possível os utilizadores efectuarem uma *bid* superior à actual, que pode ser visualizada nesta página. Está disponível também a lista das últimas propostas e uma lista de produtos relacionados.

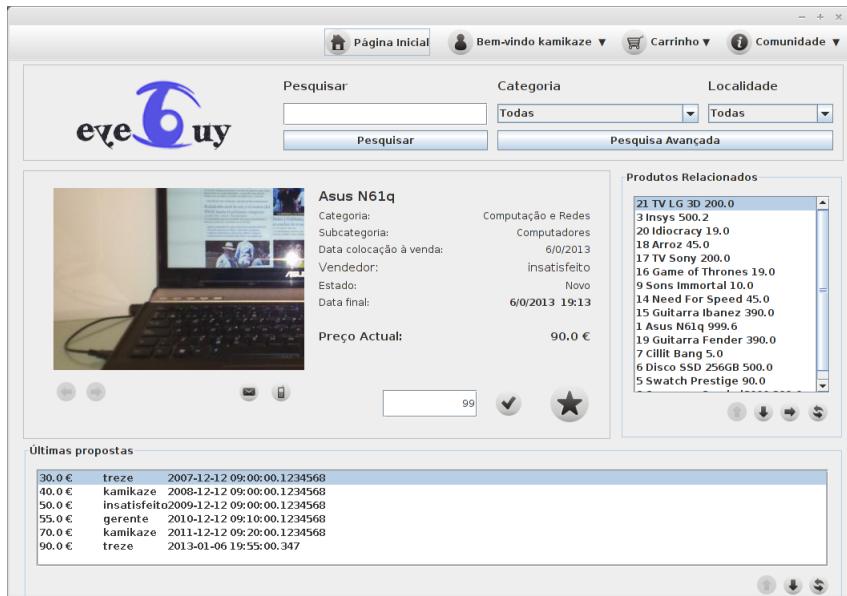


Figura 32: Página de uma dado produto em leilão

6.6.7 Ver perfil de um utilizador

Nesta página é possível visualizar todas as informações de um dado utilizador, nomeadamente, o seu nome, morada, localidade, e-mail, número de telemóvel e outras informações. É também apresentado aos utilizadores a montra de produtos ao utilizador em causa e o seu respectivo histórico. Caso o utilizador tenha vendido ou comprado algum produto é possível aos utilizadores intervenientes nesta venda ou compra, avaliar o utilizador. Só é possível uma avaliação por cada produto vendido/comprado. Deste modo, o utilizador em causa possui uma avaliação como vendedor e comprador, o número total de avaliações de ambos e o número total de compras e vendas.

A partir desta página é possível pesquisar produtos e ir para outras páginas, como o perfil de utilizador, página de vendas, histórico, carrinho de compras, etc.

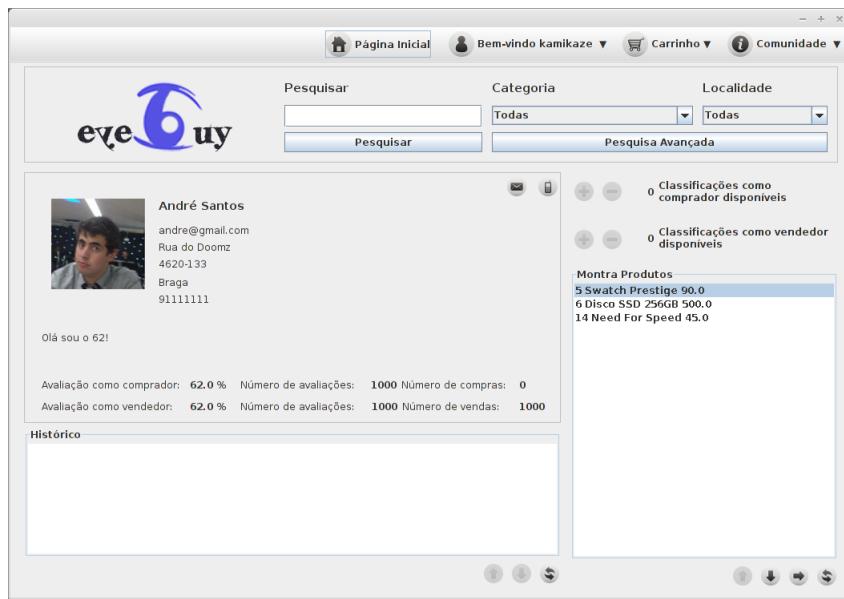


Figura 33: Página de um perfil de uma dado urilizador

6.6.8 Perfil de um utilizador

Nesta página é possível visualizar o perfil do utilizador que iniciou sessão no sistema. É possível visualizar o seu nome, morada, localidade, e-mail, número de telemóvel e outras informações. É também apresentado ao utilizador o seu histórico e a sua lista de trocas. É nesta página que o utilizador pode consultar as suas mensagens, tanto recebidas como enviadas, remover e adicionar mensagens e também editar as suas defginições pessoais. A partir desta página é possível perquisar produtos e ir para outras páginas, como o perfil de utilizador, página de vendas, histórico, carrinho de compras, etc.

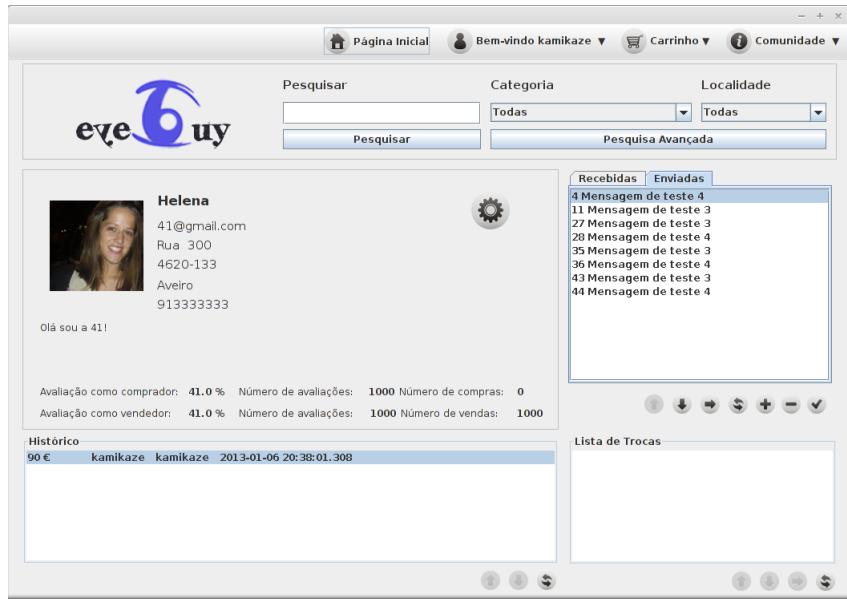


Figura 34: Página de perfil do utilizador

6.6.9 Editar perfil de utilizador

Na página editar perfil, o utilizador pode alterar todas as suas definições pessoais. Se estas alterações foram válidas, o perfil é actualizado com sucesso, caso contrário é apresentada uma mensagem de erro.

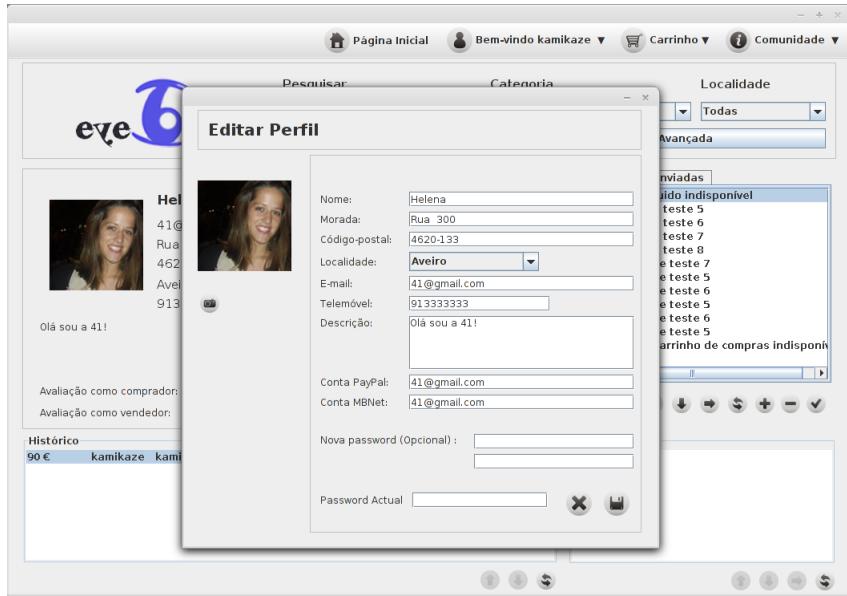


Figura 35: Página editar perfil do utilizador

6.6.10 Criar nova mensagem

É nesta página onde o utilizador pode criar uma mensagem, após o clique no botão adicionar mensagem na página do perfil do utilizador. Após o utilizador inserir os dados da mensagem, caso os dados sejam válidos, a mensagem é enviada, caso não sejam é apresentada uma mensagem de erro. No campo tipo de mensagem, no caso de o utilizador escolher *Mensagem Administrador* é automaticamente escrito no campo destinatário *EyeBuy*.

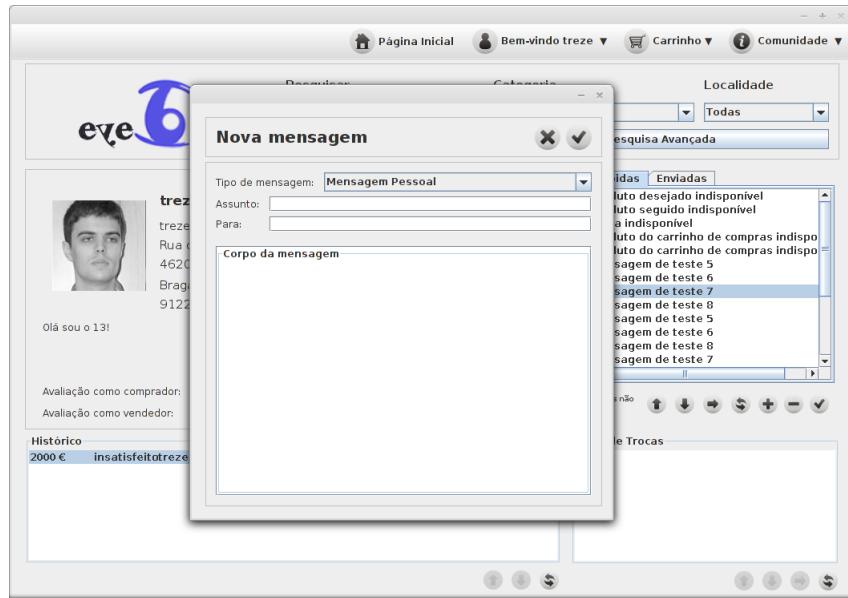


Figura 36: Página criar nova mensagem

6.6.11 Consultar uma mensagem

Nesta página o utilizador pode consultar uma mensagem, tanto recebida como enviada. Após duplo clique sobre a mensagem (ou clique no botão *ir para*) é possível visualizar a mensagem, nomeadamente, o corpo da mensagem, o destinatário e ainda uma lista de últimas mensagens recebidas.

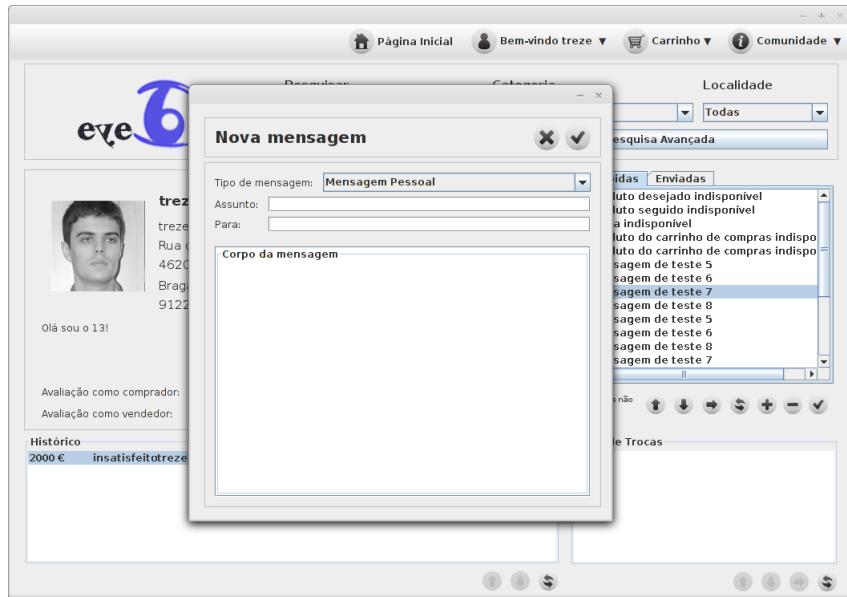


Figura 37: Página consultar mensagem

6.6.12 Página de vendas

Na página de vendas é apresentada a montra de produtos do utilizador, onde são listados todos os produtos que o utilizador tem à venda. O utilizador pode adicionar um novo produto à montra como também pode remover e editar os seus produtos. Nesta página também é apresentada uma lista de produtos relacionados e uma lista de últimas vendas, permitindo ao utilizador estar sempre actualizado das suas vendas. A partir desta página é possível pesquisar produtos e ir para outras páginas, como o perfil de utilizador, página de vendas, histórico, carrinho de compras, etc.

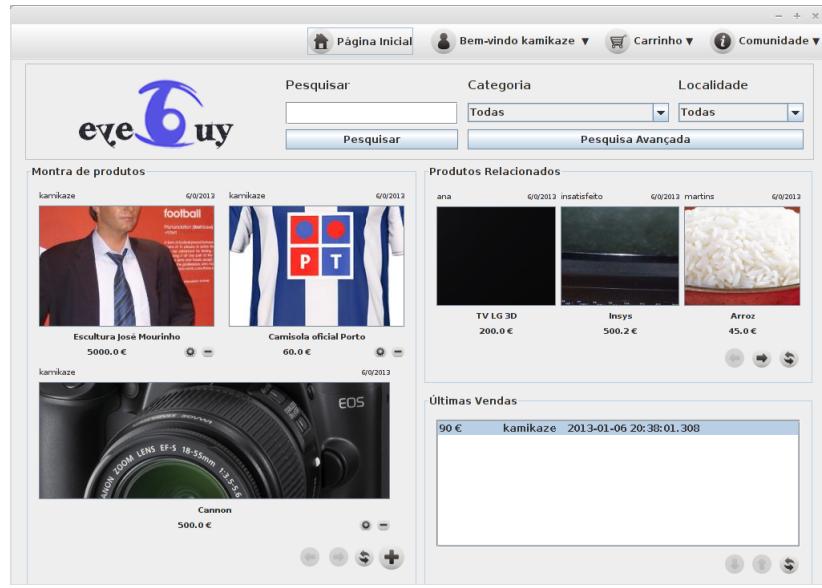


Figura 38: Página da montra de produtos do utilizador

6.6.13 Consultar carrinho de compras

Nesta página é apresentada uma lista de todos os produtos que o utilizador adicionou ao carrinho de compras. Os produtos do carrinho são separados pelos dois métodos de pagamento, nomeadamente, *MBNet* e *PayPal*. Deste modo, nesta página, são apresentadas três listas: a lista que contém todos os produtos do carrinho e as outras duas listas comtêm os produtos por método de pagamento. Esta implementação facilita o utilizador, de maneira que é mais fácil efectuar o pagamento dos produtos.

Em todas as listas é apresentado o montante total, ou seja, a soma dos preços de todos os produtos pertencentes à respectiva lista. É possível também remover um produto do carrinho ou, de uma forma mais geral, esvaziar o carrinho, ou seja, eliminar de uma só vez todos os produtos do carrinho. A partir desta página é possível pesquisar produtos e ir para outras páginas, como o perfil de utilizador, página de vendas, histórico, carrinho de compras, etc.

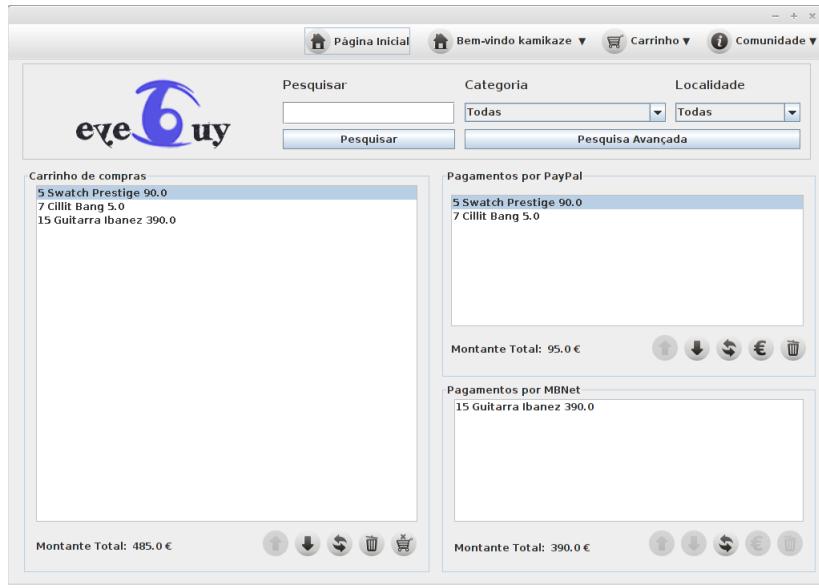


Figura 39: Página do carrinho de compras do utilizador

6.6.14 Histórico

Relativamente ao histórico de um dado utilizador, este apresenta todas as compras e vendas efectuadas pelo utilizador. O histórico também está dividido em duas listas, a lista de vendas e a lista de compras. O histórico de vendas apresenta todos os produtos que o utilizador vendeu e o total gasto. O histórico de compras funciona da mesma forma. A partir desta página é possível pesquisar produtos e ir para outras páginas, como o perfil de utilizador, página de vendas, histórico, carrinho de compras, etc.

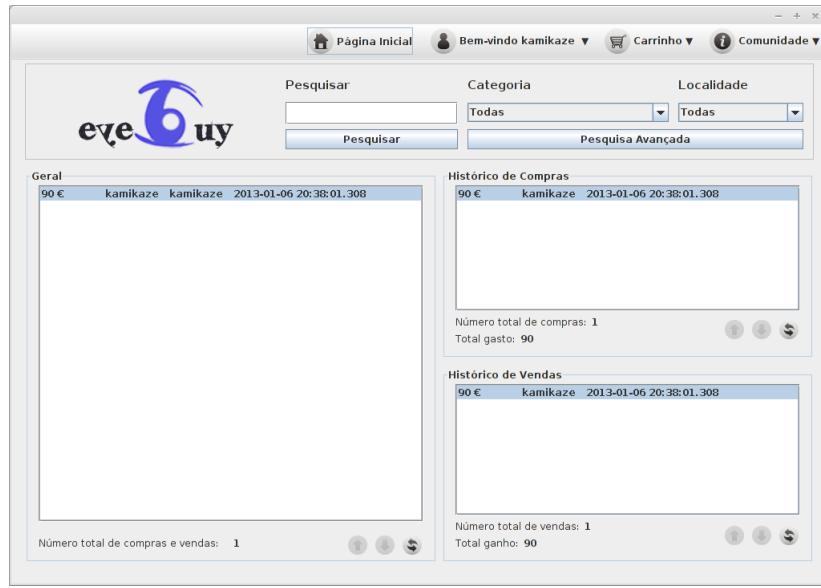


Figura 40: Página do histórico do utilizador

6.6.15 Sugestão de troca

As sugestões de troca são apresentadas quando o utilizador visualiza um produto. Nesta lista de sugestões de troca são apresentados os produtos que o utilizador tem à venda. Após clicar num produto dessa lista, é apresentado a página de sugestão de troca. Nesta página são apresentados os dois produtos, um produto que o utilizador tem à venda e o produto que o utilizador está a visualizar. Sobre os produtos são apresentadas algumas características, como o estado, o preço, o vendedor e uma imagem. Deste modo, é proporcionado ao utilizador fazer uma proposta em troca do produto seleccionado. Após isto, o outro utilizador recebe a proposta e decide aceitar ou não.

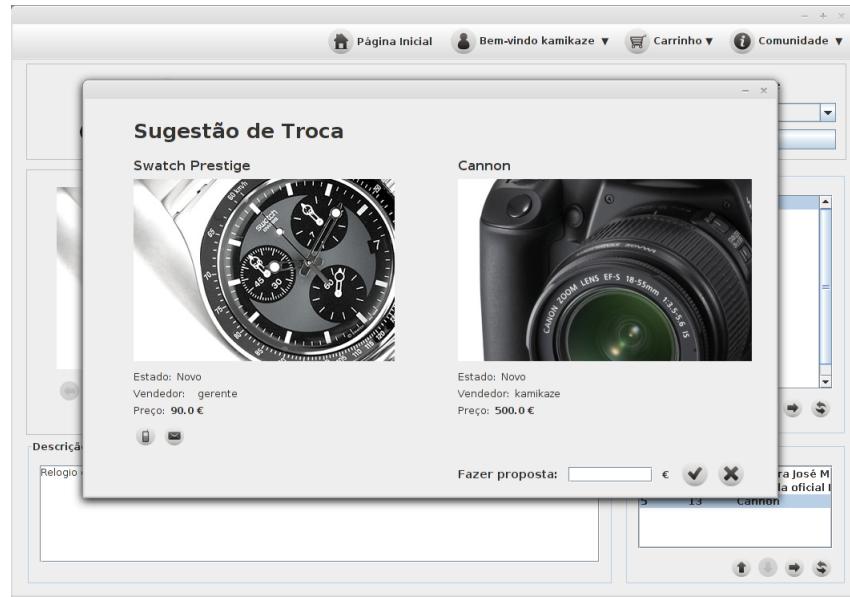


Figura 41: Página de sugestão de troca

6.6.16 Proposta de troca

A lista de propostas de troca são apresentadas quando o utilizador visita o seu perfil do utilizador. Nesta lista são apresentadas todas as propostas relativas aos produtos do utilizador. No entanto, o utilizador pode aceitar ou rejeitar a proposta.

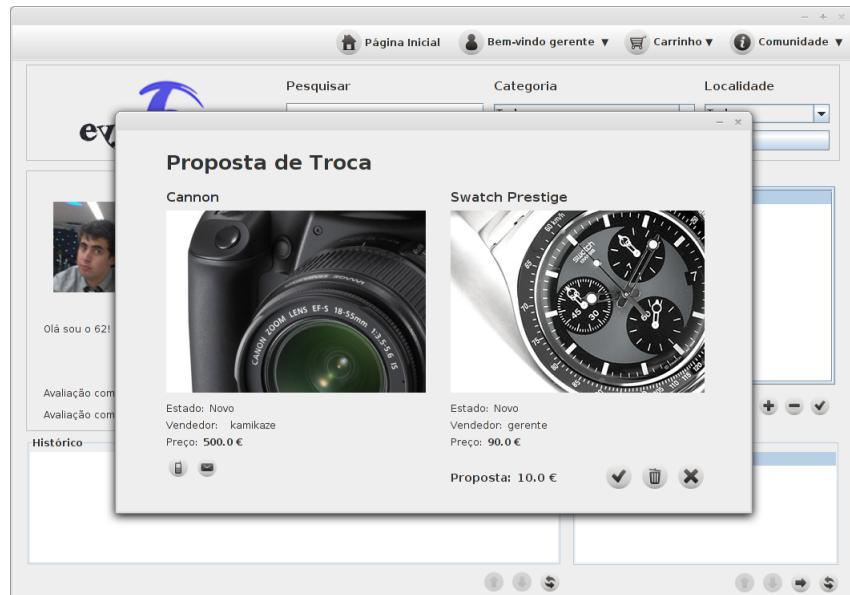


Figura 42: Página proposta de troca

6.6.17 Ajuda

Em qualquer momento o utilizador pode obter ajuda. Aqui são dados alguns conselhos ao utilizador, informações e os contactos do sistema.



Figura 43: Página de ajuda do sistema

6.6.18 Sobre o EyeBuy

Em qualquer momento o utilizador pode aceder às informações do sistema clicando no botão *Sobre o EyeBuy*.



Figura 44: Página de informação sobre o sistema

6.6.19 Mensagem de erro/aviso

Quando o utilizador pretende remover algum produto de uma dada lista de produtos é apresentada uma mensagem de aviso, assim como quando o utilizador pretende sair do sistema, quando insere dados inválidos, entre outras.

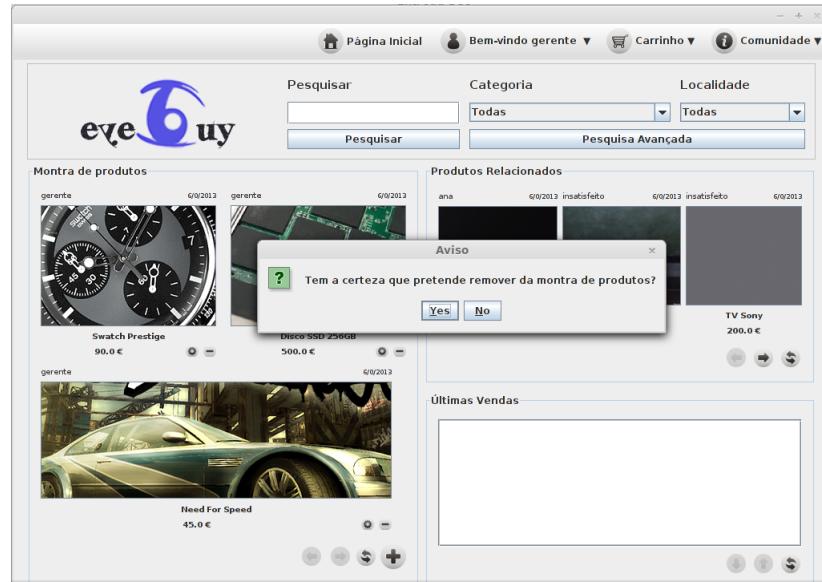


Figura 45: Mensagem de aviso

7 Conclusão

Após a análise e modelação de um sistema que cobrisse as especificações definidas no enunciado tornou-se possível fazer uma descrição suficientemente pormenorizada do sistema a ser implementado. Não obstante, o sistema modelado possui uma complexidade de implementação digna de nota sendo as funções críticas em relação à acção vender produtos por leilão, relacionadas essencialmente com o controlo baseado no tempo decorrido. Todas as funcionalidades requeridas no enunciado foram implementadas sendo que de acordo com a nossa modelação conseguimos acrescentar diversas funcionalidades adicionais que permitiram fazer com que este projecto crescesse e ganhasse assim contornos que se podem equiparar a sistemas que são utilizados no mercado real. Podemos dizer, portanto, que os objectivos traçados pelo grupo foram atingidos, tendo possivelmente superado alguns com sucesso. Analizando a aplicação cliente em termos da utilização é possível dizer que esta foi desenvolvida tendo em consideração a experiência do utilizador garantido assim uma usabilidade intuitiva e agradável onde foi utilizada a ferramenta Swing para a construção das interfaces de interacção com o utilizador. À medida que fomos desenhando as interfaces foram surgindo bastantes ideias anteriormente descritas que possibilitaram assim como resultado final uma melhor iteração com o utilizador.

Por outro lado a aplicação de administração foi pensada de modo a ser utilizada por utilizadores mais avançados pelo que a necessidade de confirmação para todas as operações não se pôs. Apesar disso todas as operações devolvem feedback acerca do sucesso ou insucesso. Outra característica essencial desta aplicação é a existência de todas as estatísticas consideradas por nós relevantes para o administrador deste sistema. As referidas estatísticas são apresentadas soba forma de gráficos do Google Charts o que permite detalhe e facilidade ao mesmo tempo na apresentação de factos ao administrador. Como já foi referido, houve algumas situações que requereram um maior esforço para serem ultrapassadas, nomeadamente a implementação do sistema de leilões e de trocas que envolvem uma complexidade um pouco acrescida e onde foi despendido mais tempo para se chegar a uma conclusão sobre o modo de funcionamento dos mesmos.

O balanço de tecnologias aprendidas/utilizadas tambem é bastante bom. Durante o desenvolvimento foi aprefeiçoadada a utilização de sistemas de controlo de versões (nomeadamente o git, estando os repositórios no github.com), as linguagens de programação orientadas aos objectos (com o uso de Java e PHP), ferramentas de protipagem e desenho de interfaces Web (HTML, JavaScript, BalsamiqMockups) e Desktop (Swing), um sistema de base de dados comercialmente usado (Oracle), bem como as lingagens SQL e PL/SQL.