|  |
| --- |
|  |
| Universidade do Minho |
| Licenciatura em Engenharia Informática |

Relatório Final

Projecto da U.C. de Modelação e Desenvolvimento

**Elaborado por:**

Grupo 33

|  |  |
| --- | --- |
|  | C:\Users\marcelo\Desktop\paxa.jpg |
| Pedro Miguel Costa Maia – A6107 | Luís Carlos Caseiro – A61009 |
| C:\Users\marcelo\Desktop\ces.jpg | C:\Users\marcelo\Pictures\EU\Pedro Faria - Cópia.jpg |
| César Morais Perdigão – A61007 | Pedro Miguel Oliveira Faria- A60998 |

Índice

[Índice Tabelas: 3](#_Toc344908890)

[Índice Figuras: 4](#_Toc344908891)

[4 – Introdução 5](#_Toc344908892)

[5 – Desenvolvimento 6](#_Toc344908893)

[5.1 – Modelo de Domínio 6](#_Toc344908894)

[5.2 – Modelo de Use-Case 7](#_Toc344908895)

[5.3 – Diagramas de Sequência 10](#_Toc344908896)

[5.4 – Diagramas de actividade 11](#_Toc344908897)

[5.5 – Diagrama de Classes 12](#_Toc344908898)

[5.6 – Diagrama da Base de Dados 13](#_Toc344908899)

[5.7 – Interface gráfica 14](#_Toc344908900)

[6 – Conclusão 21](#_Toc344908901)

[7 – Anexos 22](#_Toc344908902)

# Índice Tabelas:

[Tabela 1 - Reportar produto suspeito 22](#_Toc344909631)

[*Tabela 2 – login* 22](#_Toc344909632)

[*Tabela 3 - Fazer licitação* 23](#_Toc344909633)

[Tabela 4 - Comprar 24](#_Toc344909634)

[Tabela 5 - Analisar casos suspeitos 24](#_Toc344909635)

[Tabela 6 - Pedir sugestões 25](#_Toc344909636)

[Tabela 7 - Confirmar envio produto 25](#_Toc344909637)

[Tabela 8 - Vender 26](#_Toc344909638)

[Tabela 9 - Bloquear utilizador 26](#_Toc344909639)

[Tabela 10 - Adicionar à wishlist 27](#_Toc344909640)

[Tabela 11 - Registar 27](#_Toc344909641)

[Tabela 12 - Confirmar pagamento 28](#_Toc344909642)

[*Tabela 13 - Pesquisa produto* 28](#_Toc344909643)

[Tabela 14 - Leiloar 29](#_Toc344909644)

[Tabela 15 - Classificar utilizador 30](#_Toc344909645)

[Tabela 15 - Trocas 30](#_Toc344909646)

# Índice Figuras:

[Fig. 1– Modelo de domínio. 6](#_Toc344909647)

[*Fig. 2 - Diagrama de Use-Case* 9](#_Toc344909648)

[Fig. 3 - Diagrama de Classes 12](#_Toc344909649)

[Fig. 4 - Diagrama Base de Dados 13](#_Toc344909650)

[Fig. 5-Login 14](file:///C:\Users\marcelo\Desktop\Relatorio%20Fase%202.docx#_Toc344909651)

[Fig. 6 - Favoritos 14](file:///C:\Users\marcelo\Desktop\Relatorio%20Fase%202.docx#_Toc344909652)

[Fig. 7 - Principal 15](file:///C:\Users\marcelo\Desktop\Relatorio%20Fase%202.docx#_Toc344909653)

[Fig. 8 - Vender 15](file:///C:\Users\marcelo\Desktop\Relatorio%20Fase%202.docx#_Toc344909654)

[Fig. 9 - Histórico 16](#_Toc344909655)

[Fig. 10 - Leiloar 16](file:///C:\Users\marcelo\Desktop\Relatorio%20Fase%202.docx#_Toc344909656)

[Fig. 11 - Pesquisa avançada 17](file:///C:\Users\marcelo\Desktop\Relatorio%20Fase%202.docx#_Toc344909657)

[Fig. 12 - Produto em leilão 17](file:///C:\Users\marcelo\Desktop\Relatorio%20Fase%202.docx#_Toc344909658)

[Fig. 13 - Histórico 18](file:///C:\Users\marcelo\Desktop\Relatorio%20Fase%202.docx#_Toc344909659)

[Fig. 14 - Produto à venda 18](file:///C:\Users\marcelo\Desktop\Relatorio%20Fase%202.docx#_Toc344909660)

[Fig. 15 - Pesquisa 19](file:///C:\Users\marcelo\Desktop\Relatorio%20Fase%202.docx#_Toc344909661)

[Fig. 16 - A vender 19](file:///C:\Users\marcelo\Desktop\Relatorio%20Fase%202.docx#_Toc344909662)

[Fig. 17 - Trocas 20](file:///C:\Users\marcelo\Desktop\Relatorio%20Fase%202.docx#_Toc344909663)

[Fig. 18 - Reportar Produto Suspeito 31](#_Toc344909664)

[Fig. 19 – Login 31](#_Toc344909665)

[Fig. 20 - Fazer Licitação 32](#_Toc344909666)

[Fig. 21 - Comprar 33](#_Toc344909667)

[Fig. 22 - Analisar casos suspeitos 34](#_Toc344909668)

[*Fig. 23 - Pedir Sugestões* 35](#_Toc344909669)

[*Fig. 24 - Confirmar envio de produto* 35](#_Toc344909670)

[Fig. 25- Vender 36](#_Toc344909671)

[*Fig. 26 – Bloquear utilizador* 36](#_Toc344909672)

[Fig. 27 - Adicionar à wishlist 37](#_Toc344909673)

[Fig. 28 - Registar 37](#_Toc344909674)

[Fig. 29 - Confirmar pagamento 38](#_Toc344909675)

[Fig. 30 - TARZAN!!!! 39](#_Toc344909676)

[Fig. 32 - Pesquisar Produto 39](#_Toc344909677)

[Fig. 33 – Leiloar 40](#_Toc344909678)

# 4 – Introdução

Este relatório surge no âmbito da realização do trabalho da disciplina

de Desenvolvimento de Sistemas de Software, do curso de Engenharia

Informática da Universidade do Minho. Neste trabalho pretende-se desenvolver uma aplicação informática que suporta o comércio electrónico, permitindo aos utilizadores comprar e vender produtos.

Desta forma procedemos á análise de requisitos do sistema com base num modelo de domínio, modelo Use-Case com a respetiva especificação textual, digramas de sequência diagrama de classes e diagrama de base de dados

Este relatório descreve os passos tomados no desenvolvimento do projecto, servindo assim de apoio.

# 5 – Desenvolvimento

## 5.1 – Modelo de Domínio

O modelo de domínio e um esquema detalhado sobre todas as entidades de um grande sistema, e que representa todas as suas interacções.

Neste trabalho prático, o desejo foi modelar uma aplicação informática que suporta o comércio electrónico, que está representado no modelo de domínio na página que se segue (figura1).

Para a sua realização começamos então por analisar o enunciado do projecto onde debatemos ideias sobre as entidades a serem modeladas, tendo em conta as relações entre elas.

Sendo assim no nosso modelo de domínio começamos então por definir a entidade dos utilizadores estendendo-o de forma a capturar toda a essência do contexto do projecto.

|  |
| --- |
| C:\Users\Pedro\Dropbox\DSS\prints(atualizado)\Vendas.jpg |
| Fig. 1– Modelo de domínio. |

## 5.2 – Modelo de Use-Case

Um diagrama de use-case’s é um diagrama que representa todos os actores envolventes num determinado sistema e, mais importante que isso, todas as acções que poderão executar.

Neste modelo começamos por definir os intervenientes do sistema, designados por actores (User, Visitante, Admin.).

De acordo com o enunciado do projecto o sistema consistirá em duas aplicações, ***front end*** e ***back end,*** sendo a primeira aplicação para acesso ao sistema e a segunda para tarefas de administração. Desta forma tivemos que fazer uma distinção entre as funcionalidades relevantes do ***user*** e do ***visitante*** sendo assim as funcionalidades mais relevantes do Visitante são:

* Pesquisar Produto;
* Registar.

As principais funcionalidades do *user* são:

* Comprar;
* Vender;
* Reportar produto suspeito;
* Pesquisar produto;
* Leiloar;
* Fazer licitação;
* Login;
* Classificar utilizador;
* Adicionar à *whislist*;
* Pedir sugestões;
* Trocas.

Relativamente às funcionalidades do *Admin*, este deverá realizar a gestão e manutenção do sistema “como um todo”, sendo algumas das suas principais actividades:

* Login
* Bloquear utilizador;
* Confirmar pagamento;
* Analisar casos suspeitos.
* Confirmar envio produto;

Como ambos os intervenientes do sistema (***User, Visitante, Admin***) são entidades virtuais, temos então o ***Admin*** que pode comportar-se como um ***User***.

|  |
| --- |
| C:\Users\Pedro\Dropbox\DSS\prints(atualizado)\VendasDSS.jpg |
| Fig. 2 - Diagrama de Use-Case  FALTA A TROCA!!!!!! |

## 5.3 – Diagramas de Sequência

Os diagramas de sequência são dos mais importantes diagramas que se apresentam. São eles que mais se aproximam ao código implementado.

Após da elaboração dos ***use cases*** e da sua respectiva especificação, procedemos á elaboração dos diagramas de sequência sendo estes centrados na arquitectura do sistema que estamos a desenvolver, permitindo começar a análise do sistema.

Sendo assim cada um dos diagramas de sequência irá representar os actores que interagem com o sistema, os eventos gerados pelos actores e as respostas geradas pelo sistema.

Inicialmente os diagramas de sequência foram gerados automaticamente porém verificamos que grande parte destes não se encontravam elaborados do melhor modo, desta forma procedemos há sua rectificação. Sendo o resultado final as seguintes figuras (figura 3-16) que seguem em anexo.

## 5.4 – Diagramas de actividade???????????????

Os diagramas de actividade são diagramas que nos ajudam a compreender quais as entidades que actuam no decorrer de uma determinada actividade, e que acções são por elas executadas.

Iremos então apresentar os diagramas de actividades das actividades mais importantes que decorrem nesta aplicação informática que suporta o comércio electrónico.

|  |
| --- |
|  |

## 5.5 – Diagrama de Classes

Com base em todos os diagramas aqui apresentados, chegamos ao Diagramade Classes, que consiste na representação de todos os packages do*BuyKing* e das respectivas classes, bem como os seus métodos e atributos.

|  |
| --- |
| C:\Users\marcelo\Documents\GitHub\DSS\Classes.jpg  Fig. - Diagrama de Classes |

## 5.6 – Diagrama da Base de Dados

De seguida, encontra-se um modelo conceptual da Base de Dados criada para este projecto:

|  |
| --- |
| Fig. - Diagrama Base de Dados |

## 5.7 – Interface gráfica

Tal como pedido procedemos á criação de uma interface gráfica tendo em conta todas as funcionalidades da análise de requisitos.

|  |
| --- |
| C:\Users\Pedro\Dropbox\DSS\Interface\LogIn.png  Fig. -Login |

|  |
| --- |
| C:\Users\Pedro\Dropbox\DSS\Interface\Favoritos.png  Fig. - Favoritos |

|  |
| --- |
| C:\Users\Pedro\Dropbox\DSS\Interface\principal.png  Fig. - Principal |

|  |
| --- |
| C:\Users\Pedro\Dropbox\DSS\Interface\formulario venda.png  Fig. - Vender |

|  |
| --- |
| C:\Users\Pedro\Dropbox\DSS\Interface\Historico.png  Fig. - Histórico |

|  |
| --- |
| Fig. - Leiloar  C:\Users\Pedro\Dropbox\DSS\Interface\leiloar.png |

|  |
| --- |
| C:\Users\Pedro\Dropbox\DSS\Interface\pesquisa avançada.png  Fig. - Pesquisa avançada |

|  |
| --- |
| C:\Users\Pedro\Dropbox\DSS\Interface\produto leilão.png  Fig. - Produto em leilão |

|  |
| --- |
| C:\Users\Pedro\Dropbox\DSS\Interface\Historico.png  Fig. - Histórico |

|  |
| --- |
| C:\Users\Pedro\Dropbox\DSS\Interface\produto vender.png  Fig. - Produto à venda |

|  |
| --- |
| C:\Users\Pedro\Dropbox\DSS\Interface\resultado pesquisa.png  Fig. - Pesquisa |

|  |
| --- |
| C:\Users\Pedro\Dropbox\DSS\Interface\venderleiloar.png  Fig. - A vender |

|  |
| --- |
| Fig. - Trocas |

## 5.8 – Código Fonte

### 5.8.1 – Class BuyKing??????????????????????????????

|  |
| --- |
| package Business;  import Data.LeiloesDAO;  import Data.ProdutosDAO;  import Data.SuspeitasDAO;  import Data.TrocasDAO;  import Data.UserDAO;  import Data.VendasDAO;  import java.math.BigInteger;  import java.security.MessageDigest;  import java.security.NoSuchAlgorithmException;  import java.sql.SQLException;  import java.util.ArrayList;  import java.util.GregorianCalendar;  import java.util.List;  import java.util.logging.Level;  import java.util.logging.Logger;  public class BuyKing {  private UserDAO \_utilizadores;  private VendasDAO \_vendas;  private LeiloesDAO \_leiloes;  private SuspeitasDAO \_suspeitas\_;  private TrocasDAO \_trocas\_;  private ProdutosDAO \_produtos;  public List<Venda> pesquisaVendasSimples(String aPchave, String aCat) throws SQLException  {  return pesquisaVendasAvançada(aPchave, aCat, 0, Float.MAX\_VALUE);  }    public List<Venda> pesquisaVendasAvançada(String aPchave, String aCat, float minP, float maxP) throws SQLException {  //throw new UnsupportedOperationException();  List<Venda> l = \_vendas.getVendasAbertas();  List<Venda> res = new ArrayList<Venda>();  for(Venda v:l)  {  if((v.getPreco()>minP)&&(v.getPreco()<maxP)&&  (aPchave.matches(v.getProduto().getNome())||aPchave.matches(v.getProduto().getDescricao()))  &&(aCat.equals(v.getProduto().getCategoria())))  res.add(v);  }  return res;  }  public List<Leilao> pesquisaLeilao(String aPchave, String aCat) {  throw new UnsupportedOperationException();  }  public boolean login(String aUsername, String aPassword) throws SQLException {  //throw new UnsupportedOperationException();  Utilizador u = \_utilizadores.get(aUsername);  String md5 = BuyKing.md5crypt(aPassword);  return ((u != null) && (u.getPassmd5().equals(md5)));  }  public boolean registar(Utilizador u) throws SQLException {  //throw new UnsupportedOperationException();  return \_utilizadores.add(u);  }  public static String md5crypt(String s) {  MessageDigest md = null;  try {  md = MessageDigest.getInstance("MD5");  } catch (NoSuchAlgorithmException ex) {  Logger.getLogger(BuyKing.class.getName()).log(Level.SEVERE, null, ex);  }  byte[] digest = md.digest(s.getBytes());  BigInteger big = new BigInteger(1, digest);  String md5 = big.toString(16);  while (md5.length() < 32) {  md5 = "0" + md5;  }  return md5;  }  public boolean registaCompra(Utilizador u, Venda v) throws SQLException {  v.setComprador(u);  GregorianCalendar d = new GregorianCalendar();  d.add(GregorianCalendar.DAY\_OF\_MONTH, 7); //7 dias para conclusao da venda  v.setDataLimiteVenda(d);  boolean res = \_vendas.update(v);  return res;  }    public boolean vender(Utilizador u, Produto p, float preco) throws SQLException  {  boolean res=false;  List<Produto> pr = \_produtos.getFromUser(u);  if(pr.contains(p)&&!\_vendas.aVenda(p)&&!\_leiloes.emLeilao(p))  {  Venda v = new Venda(VendasDAO.getNewId(),preco, new GregorianCalendar(), null, null, null,p, u, null);  res=\_vendas.add(v);  }  return false;  }    public boolean licitar(Utilizador u, Leilao l, float v) throws LeilaoFechadoException, BaixaLicitacaoException, SQLException  {  boolean res=false;  if(!l.fechado())  {  Licitacao lit = new Licitacao(u, v, new GregorianCalendar());  res=l.registaLicitacao(lit);  if(res&&(v>=l.getTecto()))  {  GregorianCalendar d = new GregorianCalendar();  d.add(GregorianCalendar.DAY\_OF\_MONTH, 7);  l.setDataLimiteLeilao(d);  \_leiloes.update(l);  }  }  return res;  }    public boolean leiloar(Utilizador u, Produto p, float base, float tecto) throws SQLException  {  boolean res=false;  List<Produto> pr = \_produtos.getFromUser(u);  if(pr.contains(p)&&!\_vendas.aVenda(p)&&!\_leiloes.emLeilao(p))  {  GregorianCalendar hoje = new GregorianCalendar();  GregorianCalendar fecho = (GregorianCalendar) hoje.clone();  fecho.add(GregorianCalendar.DAY\_OF\_MONTH,14);  GregorianCalendar limite = (GregorianCalendar) fecho.clone();  limite.add(GregorianCalendar.DAY\_OF\_MONTH, 7);  Leilao l = new Leilao(LeiloesDAO.getNewId(), u, p, hoje, fecho, limite, null, null, base, tecto);  res=\_leiloes.add(l);  }  return res;  }  } |

### 5.8.2 – Class Leilão

|  |
| --- |
| package Business;  import Data.LicitacoesDAO;  import java.sql.SQLException;  import java.util.GregorianCalendar;  public class Leilao {    private int id;  private Utilizador leiloador;  private Produto p;  private GregorianCalendar dataLeilao;  private GregorianCalendar \_dataFecho;  private GregorianCalendar \_dataLimiteLeilao;  private GregorianCalendar dataPagamento;  private GregorianCalendar dataEnvioProduto;  private float \_base;  private float \_tecto;  private LicitacoesDAO licitacoes;  public Leilao(int id, Utilizador leiloador, Produto p, GregorianCalendar dataLeilao, GregorianCalendar \_dataFecho, GregorianCalendar \_dataLimiteLeilao, GregorianCalendar dataPagamento, GregorianCalendar dataEnvioProduto, float \_base, float \_tecto) {  this.id = id;  this.leiloador = leiloador;  this.p = p;  this.dataLeilao = dataLeilao;  this.\_dataFecho = \_dataFecho;  this.\_dataLimiteLeilao = \_dataLimiteLeilao;  this.dataPagamento = dataPagamento;  this.dataEnvioProduto = dataEnvioProduto;  this.\_base = \_base;  this.\_tecto = \_tecto;  this.licitacoes=new LicitacoesDAO(id);  }  } |

### 5.8.3 – Class Produto

|  |
| --- |
| package Business;  import Data.SuspeitasDAO;  import java.awt.image.BufferedImage;  public class Produto {  private int \_id;  private String \_nome;  private BufferedImage \_imagem;  private String descricao;  private String categoria;  private SuspeitasDAO suspeitas;  public Produto(int \_id, String \_nome, BufferedImage \_imagem, String descricao, String categoria) {  this.\_id = \_id;  this.\_nome = \_nome;  this.\_imagem = \_imagem;  this.descricao = descricao;  this.categoria = categoria;  suspeitas = new SuspeitasDAO(\_id);  }  } |

### 5.8.4 – Class Suspeita

|  |
| --- |
| package Business;  public class Suspeita {  private Utilizador \_queixoso;    private String \_just;  public Suspeita(Utilizador \_queixoso, String \_just) {  this.\_queixoso = \_queixoso;    this.\_just = \_just;  } |

### 5.8.5 – Class Troca

|  |
| --- |
| package Business;  import java.util.GregorianCalendar;  public class Troca {  private Utilizador \_prop;  private Utilizador \_convidado;  private Produto desejado;  private Produto oferta;  private GregorianCalendar dataProposta;  private GregorianCalendar dataConfirmacao;  private GregorianCalendar dataConclusao;  private GregorianCalendar dataLimite;  private int id;  public Troca(Utilizador \_prop, Utilizador \_convidado, Produto desejado, Produto oferta, GregorianCalendar dataProposta, GregorianCalendar dataConfirmacao, GregorianCalendar dataConclusao, GregorianCalendar dataLimite, int id) {  this.\_prop = \_prop;  this.\_convidado = \_convidado;  this.desejado = desejado;  this.oferta = oferta;  this.dataProposta = dataProposta;  this.dataConfirmacao = dataConfirmacao;  this.dataConclusao = dataConclusao;  this.dataLimite = dataLimite;  this.id = id;  }  } |

### 5.8.6 – Class Venda

|  |
| --- |
| package Business;  import java.util.GregorianCalendar;  public class Venda {  private int id;  private float \_preco;  private GregorianCalendar dataVenda;  private GregorianCalendar \_dataLimiteVenda;  private GregorianCalendar dataEnvioProduto;  private GregorianCalendar dataPagamento;  private Produto \_produto;  private Utilizador vendedor;  private Utilizador comprador;      public Venda(int id, float \_preco, GregorianCalendar dataVenda, GregorianCalendar \_dataLimiteVenda, GregorianCalendar dataEnvioProduto, GregorianCalendar dataPagamento, Produto \_produto, Utilizador vendedor, Utilizador comprador) {  this.id = id;  this.\_preco = \_preco;  this.dataVenda = dataVenda;  this.\_dataLimiteVenda = \_dataLimiteVenda;  this.dataEnvioProduto = dataEnvioProduto;  this.dataPagamento = dataPagamento;  this.\_produto = \_produto;  this.vendedor = vendedor;  this.comprador = comprador;  }  } |

### 5.8.7 – Class Utilizador

|  |
| --- |
| package Business;  import Data.ClassificacaoDAO;  import Data.FavoritosDAO;  import Data.ProdutosDAO;  import java.awt.image.BufferedImage;  import java.util.GregorianCalendar;  public class Utilizador {  private String \_username;  private String \_passmd5;  private String email;  private String localidade;  private GregorianCalendar dataNascimento;  private GregorianCalendar dataRegisto;  private BufferedImage imagem;  private ClassificacaoDAO \_classificacao;  private FavoritosDAO \_wishlist;  private ProdutosDAO meusProds;      public Utilizador(String name, String pass, String mail, String l, GregorianCalendar dN, GregorianCalendar dR, BufferedImage i)  {  \_username=name;  \_passmd5=pass;  email=mail;  localidade=l;  dataNascimento= dN;  dataRegisto=dR;  imagem=i;  \_classificacao=new ClassificacaoDAO(name);  \_wishlist=new FavoritosDAO(name);  meusProds=new ProdutosDAO();  }  } |

### 5.8.8 – Class Licitação

|  |
| --- |
| package Business;  import java.util.GregorianCalendar;  public class Licitacao {    private Utilizador u;  private float v;  private GregorianCalendar data;    public Licitacao(Utilizador user, float valor,GregorianCalendar data)  {  u=user;  v=valor;  this.data = data;  }  } |

### 5.8.9 – Class Leilao Fechado Exception

|  |
| --- |
| package Business;  public class LeilaoFechadoException extends Exception{  public LeilaoFechadoException() {  }  public LeilaoFechadoException(String message) {  super(message);  }    } |

### 5.8.9 – Class Classificação

|  |
| --- |
| package Business;  import java.util.GregorianCalendar;  public class Classificacao {  public Classificacao(Utilizador classificador, GregorianCalendar data, int valor) {  this.classificador = classificador;  this.data = data;  this.valor = valor;  } |

### 5.8.9 – Class Baixa Licitação Exception

|  |
| --- |
| package Business;  public class BaixaLicitacaoException extends Exception{  public BaixaLicitacaoException() {  }  public BaixaLicitacaoException(String message) {  super(message);  }    } |

# 6 – Conclusão

Ao longo da realização deste trabalho, deparamo-nos com inúmeras dificuldades, que fomos superando à medida que nos tornamos mais familiarizados com a UC.

Como esperávamos a elaboração desta modelação gerou várias discussões, principalmente a nível da escolha das entidades *vs* atributos do modelo de domínio, na escolha correcta das actividades dos actores,????????????????. Após várias tentativas e optando pela simplicidade, ficamos com diversos modelos simples e de fácil compreensão.

O grupo conseguiu ter sempre em mente os objectivos a que se propôs e cumpri-los com sucesso, contribuindo assim para um bom trabalho e uma boa formação individual.

Após a realização deste trabalho, chegamos \_a conclusão que adquirimos conhecimentos importantes sobre a planificação e modelação de projectos complexos. Os conhecimentos adquiridos durante a realização do projecto e a frequência da Unidade Curricular permitir-nos-~ao abordar problemas de uma forma mais metódica e standardizada.

# 7 – Anexos

**- Tabelas de descrição textual dos use cases:**

|  |
| --- |
| C:\Users\marcelo\Dropbox\DSS\tabelas\reportar produto.png |
| Tabela 1 - Reportar produto suspeito |

|  |
| --- |
| C:\Users\marcelo\Dropbox\DSS\tabelas\login.png |
| *Tabela 2 – login* |

|  |
| --- |
| C:\Users\marcelo\Dropbox\DSS\tabelas\fazer licitação1.png  C:\Users\marcelo\Dropbox\DSS\tabelas\fazer licitação2.png |
| *Tabela 3 - Fazer licitação* |

|  |
| --- |
| C:\Users\marcelo\Dropbox\DSS\tabelas\comprar.png |
| Tabela 4 - Comprar |

|  |
| --- |
| C:\Users\marcelo\Dropbox\DSS\tabelas\analisa caso suspeito.png |
| Tabela 5 - Analisar casos suspeitos |

|  |
| --- |
| C:\Users\marcelo\Dropbox\DSS\tabelas\pedir sugestões.png |
| Tabela 6 - Pedir sugestões |

|  |
| --- |
| C:\Users\marcelo\Dropbox\DSS\tabelas\confirma envio produto.png |
| Tabela 7 - Confirmar envio produto |

|  |
| --- |
| C:\Users\marcelo\Dropbox\DSS\tabelas\vender.png |
| Tabela 8 - Vender |

|  |
| --- |
| C:\Users\marcelo\Dropbox\DSS\tabelas\bloquear utilizador.png |
| Tabela 9 - Bloquear utilizador |

|  |
| --- |
| C:\Users\marcelo\Dropbox\DSS\tabelas\adic wishlist.png |
| Tabela 10 - Adicionar à wishlist |

|  |
| --- |
| C:\Users\Pedro\Dropbox\DSS\prints(atualizado)\registar.jpg |
| Tabela 11 - Registar |

|  |
| --- |
| C:\Users\marcelo\Dropbox\DSS\tabelas\confirma pagamento.png |
| Tabela 12 - Confirmar pagamento |

|  |
| --- |
| C:\Users\Pedro\Dropbox\DSS\prints(atualizado)\pesquisar.jpg |
| *Tabela 13 - Pesquisa produto* |

|  |
| --- |
| C:\Users\marcelo\Dropbox\DSS\tabelas\leiloar1.png  C:\Users\marcelo\Dropbox\DSS\tabelas\leiloar2.png |
| Tabela 14 - Leiloar |

|  |
| --- |
| C:\Users\marcelo\Dropbox\DSS\tabelas\classificar users.png |
| Tabela 15 - Classificar utilizador |

|  |
| --- |
|  |
| Tabela - Trocas |

**- Imagens relativas aos diagramas de sequência:**

|  |
| --- |
| C:\Users\Pedro\Dropbox\DSS\prints(atualizado)\Reportar produto suspeito - Main.jpg |
| Fig. - Reportar Produto Suspeito |

|  |
| --- |
| C:\Users\Pedro\Dropbox\DSS\prints(atualizado)\Login - Main.jpg |
| Fig. 19 – Login |

|  |
| --- |
| C:\Users\Pedro\Dropbox\DSS\prints(atualizado)\Fazer licitação - Main.jpg |
| Fig. 20 - Fazer Licitação |

|  |
| --- |
| C:\Users\Pedro\Dropbox\DSS\prints(atualizado)\Comprar - Main.jpg |
| Fig. 21 - Comprar |

|  |
| --- |
| C:\Users\Pedro\Dropbox\DSS\prints(atualizado)\Analisar casos suspeitos - Main.jpg |
| Fig. - Analisar casos suspeitos |

|  |
| --- |
| C:\Users\Pedro\Dropbox\DSS\prints(atualizado)\Pedir Sugestões - Main.jpg |
| *Fig. 23 - Pedir Sugestões* |

|  |
| --- |
| C:\Users\Pedro\Dropbox\DSS\prints(atualizado)\Confirmar envio de produto - Main.jpg |
| *Fig. 24 - Confirmar envio de produto* |

|  |
| --- |
| C:\Users\Pedro\Dropbox\DSS\prints(atualizado)\Vender - Main.jpg |
| Fig. 25- Vender |

|  |
| --- |
| C:\Users\Pedro\Dropbox\DSS\prints(atualizado)\Bloquear utilizadores - Main.jpg |
| *Fig. 26 – Bloquear utilizador* |

|  |
| --- |
| C:\Users\Pedro\Dropbox\DSS\prints(atualizado)\Adicionar à wishlist - Main.jpg |
| Fig. 27 - Adicionar à wishlist |

|  |
| --- |
| C:\Users\Pedro\Dropbox\DSS\prints(atualizado)\Registar - Main.jpg |
| Fig. 28 - Registar |

|  |
| --- |
| C:\Users\Pedro\Dropbox\DSS\prints(atualizado)\Confirmar pagamento - Main.jpg |
| Fig. 29 - Confirmar pagamento |

|  |
| --- |
|  |
| Fig. - TARZAN!!!! |

|  |
| --- |
| C:\Users\Pedro\Dropbox\DSS\prints(atualizado)\Pesquisar produto - Main.jpg |
| Fig. - Pesquisar Produto |

|  |
| --- |
| C:\Users\Pedro\Dropbox\DSS\prints(atualizado)\Leiloar - Main.jpg |
| Fig. 33 – Leiloar |