



UNIVERSIDADE POLITÉCNICA
A POLITÉCNICA
Instituto Superior de Gestão, Ciências e Tecnologias

LICENCIATURA EM ENGENHARIA INFORMÁTICA E DE TELECOMUNICAÇÕES

**SISTEMA ONLINE DE COBRANÇA DE IMPOSTO AUTÁRQUICO DE VEÍCULOS
DE AUTOMOVÉIS**

CASO DE ESTUDO:
CONSELHO MUNICIPAL DA CIDADE DE MAPUTO

SOARES DE SOUSA CHIROUTE

MAPUTO

2024



UNIVERSIDADE POLITÉCNICA
A POLITÉCNICA
Instituto Superior de Gestão, Ciências e Tecnologias

LICENCIATURA EM ENGENHARIA INFORMÁTICA E DE TELECOMUNICAÇÕES

**SISTEMA ONLINE DE COBRANÇA DE IMPOSTO AUTÁRQUICO DE VEÍCULOS
DE AUTOMOVÉIS**

CASO DE ESTUDO:

CONSELHO MUNICIPAL DA CIDADE DE MAPUTO

SOARES DE SOUSA CHIROUTE

SUPERVISOR: Prof. Doutor Horácio Zimba

Monografia científica apresentada a Instituto Superior de Gestão e Tecnologias – Universidade Politécnica, como parte dos requisitos para obtenção do grau de Licenciatura em Engenharia Informática e de Telecomunicações

MAPUTO

2024

DECLARAÇÃO DE HONRA

Declaro que este trabalho de pesquisa nunca foi apresentado na sua essência para obtenção de qualquer grau académico ou outra finalidade dentro ou fora da instituição. Ela constitui o resultado da minha pesquisa pessoal em coordenação com meu supervisor.

Maputo, Março de 2024

(Soares de Sousa Chiroute)

DEDICATÓRIA

O presente trabalho é dedicado exclusivamente aos meus pais *Mário Adriano Chirrute e Rosa Luís Valoi*. Aos meus irmãos: *Nemésio Mário Chirrute, Ancha Mário Chirrute, Neila Adriano Chirrute* e aos meus tios *Herculano Adriano Chirrute e Estrela Chiule*, pelo apoio, atenção, companheirismo e pela motivação.

AGRADECIMENTOS

Primeiramente quero agradecer a Deus pelo dom da vida e saúde, força e acima de tudo pelas bênçãos sucessivas que me tem concedido durante a minha jornada.

Quero também aproveitar a oportunidade para agradecer profundamente e de forma sincera a minha família, especialmente ao meu tio *Herculano Chirrute*, ao meu pai *Mário Chirrute* e ao meu *Irmão Nemésio Chirrute* que desempenharam um papel vital na minha formação.

Os meus agradecimentos se estendem aos meus amigos *Márcia Novela*, *Jesse Galbo* e *Madalena Rangel*, que sempre apoiaram nas minhas decisões, incentivando a seguir em frente no curso de Engenharia Informática e de Telecomunicações.

Um agradecimento especial ao meu supervisor *Prof. Doutor Horácio Zimba*, que aceitou o meu pedido e sem impor condicionalismos orientou-me sabiamente e com muita paciência, o meu muito obrigado.

Estendo os meus agradecimentos a todos os meus docentes pelos ensinamentos e aos meus colegas do curso de Engenharia Informática e Telecomunicações por terem incentivado e contribuído na minha formação.

RESUMO

O Imposto tem como finalidade munir o Estado de recursos financeiros com vista a realizar suas actividades a favor dos cidadãos. Na capital moçambicana a cobrança de Imposto Autárquico sobre Veículos (IAV) – vulgarmente conhecido por “manifesto” tem sido de extrema importância para suprir esta necessidade.

O Conselho Municipal da Cidade de Maputo (CMCM), possui postos de cobranças móveis escalados em locais de maior aglomeração populacional e instituições com elevado número de trabalhadores, como forma de dinamizar o processo de cobrança de impostos, porém, apesar de existência destes, os automobilistas queixam-se de problemas como: mau atendimento, lentidão do sistema, incompatibilidade de impressoras nos locais de atendimento (postos de cobrança) o que acaba gerando longas filas e morosidade no atendimento dos cidadãos.

Com base nas dificuldades registadas nos postos de cobrança, surgiu a necessidade de conceber um *Sistema de Cobrança Online de Impostos (SCOI)*, como objectivo descongestionar os postos de cobrança, responder as necessidades do dia-a-dia ajustando-se aos utilizadores e seus dispositivos móveis garantindo mobilidade e flexibilidade através da internet, em qualquer lugar.

Este trabalho pretendia desenvolver uma simulação de um Sistema de Cobrança Online de Impostos (*SCOI*) que permitirá flexibilizar o processo de atendimento aos munícipes no pagamento de impostos e aumentar o valor das receitas fiscais para os cofres do CMCM.

O trabalho seguiu uma metodologia baseada no método descritivo-exploratório com uma abordagem significativamente qualitativa com algum suporte quantitativo, tendo se utilizado as seguintes técnicas de recolha de dados: entrevista, observação, questionário.

A amostra colhida de forma aleatória, consistiu em 21 condutores de veículos na cidade de Maputo e para a análise da parte quantitativa dos dados recorreu-se ao SPSS na sua Versão 2.0.

Os resultados da pesquisa revelaram que, os condutores concordam que um sistema de cobrança online de Impostos, pode facilitar e flexibilizar o processo de pagamento, bem como contribuir para o aumento de receitas fiscais para os cofres do município.

Palavras-chave: *Cobrança online, Mobilidade e Flexibilidade de atendimento, Impostos autárquico de viatura.*

ABSTRACT

The purpose of the Tax is to provide the State with financial resources to carry out its activities in favor of citizens. In the Mozambican capital the collection of Municipal Vehicle Tax (IAV) - commonly known as "manifesto" has been extremely important.

The Municipal Council of Maputo City (CMMC) has mobile collection points scaled in places with greater population crowding and institutions with a high number of workers, as a way to streamline the process of collecting taxes, however, although there are collection points furniture, motorists complain of poor service, slow system, incompatibility of printers, which ends up generating long queues and slowness in serving citizens.

Based on the difficulties registered at the collection points, the initiative to design an online tax collection system arises. This solution aims to decongest billing points, respond to the daily necessities by adjusting to users and their mobile devices in a very intuitive and simple way, guaranteeing mobility and flexibility through the intranet, anywhere. Noting the need for the user to have to travel to the collection points scaled by the city council.

For the conception of this work the descriptive methodology was applied, since it was necessary to collect data and based on the results of the data collected, to model the proposed system. With the elaboration of this work, we expect to increase the number of tax revenues, contributing significantly in the fiscal year.

Keywords: *Online Billing, Mobility and Flexibility, Taxes*

LISTA DE SIGLAS E ACRÓNIMOS

API.....	Interface De Programação de Aplicações
CMCM	Conselho Municipal da Cidade de Maputo
HTML.....	Linguagem de Marcação de Hipertexto
HTTP.....	Protocolo de Transferência de Hipertexto
IAV.....	Imposto Autárquico de Veículos
NUIT.....	Número Único de Identificação Tributária
BI.....	Bilhete de Identidade
PHP	Pré-processador de Hipertexto
RF.....	Requisitos Funcionais
RFN.....	Requisitos Não funcionais
SGBD	Sistema de Gestão de Banco de Dados
SQL	Linguagem de Consulta Estruturada
SSL	Camada de Soquetes Segura
TICS	Tecnologias de Informação e Comunicação
UML	Linguagem de Modelagem Unificada
URL.....	Localizador Uniforme de Recursos

LISTA DE FIGURAS

Figura 1: Visão lógica da arquitectura cliente leve da web fonte: marcel (2011)	23
<u>Figura 2: Processo da extreme programming (xp) fonte: pressman (2011, p.88)</u>	<u>27</u>
<u>Figura 3: Estrutura orgânica do Conselho Municipal de Maputo.....</u>	<u>42</u>
figura <u>4: Distribuição das dificuldades dos utentes nos postos de cobrança</u>	<u>45</u>
<u>Figura 5: Distribuição da opinião da população sobre o actual sistema de cobrança</u>	<u>46</u>
<u>FIGURA 6: Distribuição da opinião da população sobre o atendimento</u>	<u>46</u>
<u>FIGURA 7: Processo de cobrança de imposto actual (Fonte: elaborado pelo autor).....</u>	<u>49</u>
<u>Figura 8: Fluxograma do sistema actual (fonte: elaborado pelo autor).....</u>	<u>49</u>
<u>FIGURA 9: Modelo genérico do sistema proposto (Fonte: elaborado pelo autor)</u>	<u>50</u>
<u>FIGURA 10: Diagrama de caso de uso.....</u>	<u>52</u>
<u>FIGURA 11: Diagrama de classe</u>	<u>54</u>
<u>Figura 12: Diagrama de actividade verificar estado de pagamento</u>	<u>55</u>
<u>Figura 13: Diagrama de sequência emitir pagamento.....</u>	<u>56</u>
<u>figura 14: Diagrama de Sequência verificar estado pagamento</u>	<u>56</u>

ÍNDICE

DECLARAÇÃO DE HONRA	I
DEDICATÓRIA	II
AGRADECIMENTOS	III
RESUMO.....	IV
ABSTRACT	V
LISTA DE SIGLAS E ACRÓNIMOS.....	VI
LISTA DE FIGURAS	VII
1 INTRODUÇÃO	1
1.1 DEFINIÇÃO DO PROBLEMA	2
1.2 HIPÓTESES DA PESQUISA	2
1.3 OBJECTIVO GERAL.....	3
1.3.1 <i>Objectivos Específicos</i>	3
1.4 JUSTIFICATIVA	3
2 REVISÃO DA LEITURA.....	5
2.1 IMPOSTO AUTÁRQUICO DE VEÍCULOS (IAV)	5
2.2 INCIDÊNCIA SUBJECTIVA DO IMPOSTO AUTÁRQUICO DE VEÍCULOS DE AUTOMÓVEIS.....	5
2.3 ISENÇÕES DO IMPOSTO AUTÁRQUICO DE VEÍCULOS DE AUTOMÓVEIS	5
2.4 SISTEMAS DE INFORMAÇÃO.....	6
2.4.1 <i>Tipologia de Sistemas de Informação</i>	6
2.4.2 <i>Informação</i>	7
2.4.3 <i>Software</i>	8
2.4.4 <i>Arquitectura de Software</i>	9
2.5 METODOLOGIA DE DESENVOLVIMENTO DE SOFTWARE.....	12
2.5.1 <i>Modelo Cascata</i>	12
2.5.2 <i>Modelo Incremental</i>	13
2.5.3 <i>Modelo espiral de Boehm</i>	13
2.5.4 <i>Metodologias de Desenvolvimento Ágil</i>	14
2.6 ENGENHARIA DE REQUISITOS DE SOFTWARE.....	17
2.6.1 <i>Requisitos funcionais</i>	17
2.6.2 <i>Requisitos não funcionais</i>	18
2.7 TECNOLOGIAS E FERRAMENTAS DE DESENVOLVIMENTO.....	18
2.7.1 <i>Linguagem de Modelagem Unificada – UML</i>	18
2.7.2 <i>Elementos de Modelagem</i>	20
2.7.3 <i>Diagramas UML</i>	21

2.7.4	<i>Bootstrap</i>	24
2.7.5	<i>Sistema de Gestão de Banco de Dados</i>	25
2.7.6	<i>Modelos de banco de dados</i>	26
3	METODOLOGIA	28
3.1	TIPO DE PESQUISA.....	28
3.2	POPULAÇÃO.....	28
3.3	AMOSTRA.....	29
3.3.1	<i>Técnicas de recolha e organização de dados</i>	29
3.3.2	<i>Dados Primários</i>	29
3.3.3	<i>Dados Secundários</i>	30
3.3.4	<i>Colecta de Dados</i>	30
4	APRESENTAÇÃO DE RESULTADOS	31
4.1	CARACTERIZAÇÃO DE MUNICÍPIO DE MAPUTO – LOCAL DE ESTUDO	31
4.1.1	<i>Estrutura orgânica do Conselho Municipal de Maputo</i>	32
4.2	APRESENTAÇÃO DOS RESULTADOS DO INQUÉRITO	33
4.2.1	<i>Dificuldades Enfrentadas</i>	33
4.2.2	<i>Eficiência do Sistema</i>	34
4.2.3	<i>Atendimento</i>	34
4.3	CARACTERÍSTICAS DO SISTEMA EM USO PARA O PROCESSO DE COBRANÇA DE IMPOSTOS	35
4.4	CARACTERÍSTICAS DO SISTEMA PROPOSTO PROCESSO DE COBRANÇA DE IMPOSTOS	36
4.5	FUNCIONALIDADES DO SISTEMA.....	37
4.5.1	<i>Requisitos Funcionais do Sistema</i>	38
4.5.2	<i>Requisitos não funcionais</i>	38
4.5.3	<i>Diagrama de Casos de Uso</i>	38
4.5.4	<i>Diagrama de Classe</i>	41
4.5.5	<i>Diagrama de Actividades</i>	41
4.5.6	<i>Diagrama de Sequência</i>	42
5	CONCLUSÕES	43
6	RECOMENDAÇÕES	45
7	BIBLIOGRAFIA	46
8	ANEXOS	49
8.1	ANEXO 1– AUTORIZAÇÃO PARA A RECOLHA DE DADOS	49
8.2	APÊNDICE 1– INQUÉRITO	50
8.3	APÊNDICE 2 – INTERFACES GRÁFICAS DO SISTEMA	51

1 INTRODUÇÃO

A Tecnologia da Informação (TI) representa um conjunto de actividades, ferramentas e soluções tecnológicas que visam produzir, manipular, transmitir e armazenar informações para atingir objectivos específicos.

A introdução de ferramentas de TI em uma empresa ou instituição provoca impactos consideráveis na sua estrutura e organização, repercutindo nos níveis estratégicos, operacionais e táticos.

Um Sistema de Informação é todo mecanismo projectado com a finalidade de colectar, processar, armazenar e transmitir informações, de maneira a facilitar o acesso de usuários interessados, solucionando problemas e atendendo suas necessidades (AZEVEDO, 2018).

A troca de informação por meio da tecnologia facilita o contacto das empresas (instituições) com seus clientes, funcionários e fornecedores, além de prover, de forma mais flexível, uma melhor organização do trabalho (LAUDON e LAUDON, 2010).

Com a vasta gama de ferramentas disponibilizadas pelas tecnologias de informação, o processo de automatização das tarefas tornou-se uma necessidade nas empresas, permitindo uniformizar o método de trabalho e centralizar informação, evitando erros humanos, o que permite uma melhoria no serviço ao cliente.

Com o desenvolvimento de uma solução baseada em tecnologias *Web* é possível minimizar os constrangimentos enfrentados, e por sua vez, agilizar vários processos de negócio reduzindo o envolvimento de recursos humanos.

As aplicações *Web* respondem as necessidades do dia-a-dia ajustando-se aos utilizadores, aos seus negócios e aos seus dispositivos móveis de forma muito intuitiva e simples, através da internet, em qualquer lugar.

O presente trabalho subordinado ao tema: *sistema online de cobrança de imposto autárquico de veículos de automóveis* propõe uma solução para resolver um dos maiores problemas enfrentados pelo Conselho Municipal da cidade de Maputo que é a falta de um sistema de cobrança online de impostos autárquicos de veículos de automóveis capaz de dinamizar o processo de cobrança de imposto em qualquer lugar e a qualquer momento.

1.1 Definição do problema

O processo de cobrança de impostos autárquicos de veículos de automóveis aos cidadãos, por parte do Conselho Municipal, sempre seguiu os mesmos moldes, sendo que desde que o sistema actual foi concebido a sua estrutura e regras básicas nunca foram alteradas.

Para efectuar o pagamento, o cidadão deve passar por uma serie de procedimentos (05 principais): deslocar-se aos Postos de Cobrança (PC's), onde apresenta o Livrete, Bilhete de identidade (BI) e Número Único de Identificação Tributária (NUIT). Ao receber estes documentos, o funcionário do Conselho Municipal, emite um aviso de pagamento. Em seguida o cidadão desloca-se até ao caixa onde efectua o pagamento, e posteriormente desloca-se até a mesa de impressão onde levanta o Manifesto.

Este processo torna-se moroso e muitas das vezes não permite satisfazer a demanda durante a busca por estes serviços, por parte dos munícipes ou que gera tumultos e aglomerados nos Postos de cobranças.

O Sistema de Cobrança de Imposto de Veículos de Automóveis (SCIVA) existente actualmente, possui as seguintes limitações: (i) não permite a inserção dos dados do livrete, BI e NUIT de forma remota ou *on-line*; (ii) não permite o pagamento *on-line* para o banco; e (iii) não emite os recibos electrónicos remotamente,

Tomando em consideração, todos os aspectos acima citados, que obrigam os munícipes a se deslocarem até os postos de cobrança para obter os recibos, incluindo a morosidade nos processos de cobrança, causada pelas enchentes e filas enormes nos postos de cobranças de impostos, coloca-se a seguinte questão de partida:

Como as TIC móveis podem dinamizar o processo de cobrança e reduzir grandes aglomerados nos postos de cobrança de impostos autárquicos de veículos no Conselho Municipal?

1.2 Hipóteses da pesquisa

H₀: A introdução de um sistema de cobranças *on-line* de impostos autárquicos de veículos de automóveis não irá dinamizar o processo de cobrança e reduzir enchentes nos postos de cobrança.

H₁: A introdução de um sistema de cobrança *on-line* de impostos autárquicos de veículos de automóveis poderá dinamizar o processo de cobrança, contudo reduzir grandes enchentes nos postos de cobrança.

1.3 Objectivo Geral

Desenvolver um protótipo de sistema *on-line* de cobrança de imposto autárquicos de veículos de automóveis para a Cidade de Maputo.

1.3.1 Objectivos Específicos

- a) Descrever o sistema de cobrança actualmente utilizado nos PC's no Município da Cidade de Maputo;
- b) Identificar os requisitos necessários para a concepção de um Sistema de Cobrança On-line de Imposto de Veículos;
- c) Construir um protótipo do Sistema de cobrança proposto;
- d) Analisar a eficiência e eficácia do protótipo (modelo) do Sistema concebido no desembaraço do processo de cobrança.

1.4 Justificativa

A morosidade no processo de cobrança de impostos sobre veículos é um problema que afecta directamente aos proprietários dos veículos, porém de uma certa forma pode afectar a instituição.

No concernente aos proprietários, estes dependem muito tempo em filas de espera, comprometendo as outras actividades diárias. Por outro lado, no que tange a instituição, a falta de flexibilidade no processo de cobrança pode afectar de forma negativa a imagem da instituição (Conselho Municipal), devido a insatisfação do proprietário no processo de atendimento, incorrendo-se ao isco de redução significativa a receita fiscal.

Desta forma, o sistema *Web* responde de pronto às necessidades dos utilizadores, para além de se ajustar a qualquer dispositivo móvel garantindo desta forma a mobilidade.

Tendo como base a mobilidade e disponibilização de uma solução de cobrança a tempo inteiro, a implementação do tema proposto é de extrema relevância:

a) Âmbito social

- Prestação de serviços com qualidade aos munícipes porque com a disponibilidade de informação, reduz o tempo e o custo para o acesso a mesma.
- Aumento do Grau de Satisfação dos Munícipes no uso dos Serviços.
- Serviço disponível a todo o momento (24h).
- Evita aglomerações nos locais de Cobrança, o que pode contribuir para diminuição de propagação de várias doenças.

- Evita a pressão e sobrecarga por parte dos Colaboradores do CMCM.

b) Âmbito Económico

- Flexibilizar o processo de pagamento Imposto Sobre Veículos.
- Arrecadação de mais receitas fiscais para os cofres do Município.
- Descentralizar actividade e melhorar os serviços.
- Elevar taxa de eficiência e eficácia no atendimento.

c) Âmbito Académico

- Servira de base para pesquisas, consulta, quiçá propostas de melhoria de funcionamento do sistema.

2 REVISÃO DA LEITURA

Neste capítulo apresentamos e discutimos alguns conceitos relacionados com o tema que serviram de base de sustentação na análise de dados.

2.1 Imposto Autárquico de veículos (IAV)

De acordo com o artigo 65 do Decreto nº 19/2002, o Imposto Autárquico de Veículos (IAV) substitui nas autarquias o Imposto Sobre Veículos (ISV).

Este imposto incide sobre o uso e fruição dos veículos, matriculados ou registados nos serviços competentes no território moçambicano, ou, independentemente, de registo ou matrícula, logo que, decorridos cento e oitenta dias a contar da sua entrada no território moçambicano, venham a circular ou a ser usados em condições normais da sua utilização.

De acordo com o artigo 65 do Decreto nº 19/2002 os veículos sobre os quais incide o imposto autárquico são listados a seguir:

- a) Automóveis ligeiros e automóveis pesados de antiguidade menor ou igual a vinte e cinco anos;
- b) Motociclos de passageiros com ou sem carro de antiguidade menor ou igual a quinze anos;
- c) Aeronaves com motor de uso particular;
- d) Barcos de recreio com motor de uso particular.

2.2 Incidência subjectiva do Imposto Autárquico de veículos de Automóveis

São sujeitos passivos do imposto os proprietários dos veículos, quer sejam pessoas singulares ou colectivas, de direito público ou privado, residentes na respectiva autarquia, presumindo-se como tais, até que se prove o contrário, as pessoas em nome dos quais se encontrem matriculados ou registados, (Artigo 66 do Decreto nº 19/2002).

2.3 Isenções do Imposto Autárquico de veículos de Automóveis

Estão isentos do imposto:

- a) O estado e qualquer dos seus serviços, estabelecimentos e organismos, ainda que personalizados;
- b) As autarquias locais e suas associações e ou federações municipais;
- c) Os estados estrangeiros, quando haja reciprocidade de tratamento;
- d) O pessoal das missões diplomáticas e consulares nos termos das respectivas conversações;

- e) As organizações estrangeiras ou internacionais, nos termos de acordos celebrados pelo estado moçambicano;

2.4 Sistemas de informação

Um sistema de informação é um conjunto de componentes inter-relacionados que colectam, processam e disseminam dados e informações para atingir um objectivo (STAIR e REYNOLDS, 2005). Ele desempenha um papel fundamental nas operações das organizações, automatizando processos, facilitando a tomada de decisões e melhorando a eficiência geral.

Segundo STAIR e REYNOLDS (2005), um sistema de informação é composto por quatro elementos principais a descrever:

- (i) **Entrada:** Envolve a coleta e captura de dados básicos, que são essenciais para desencadear ações de processamento.
- (ii) **Processamento:** Consiste na conversão ou transformação de dados em saídas úteis. Isso pode incluir cálculos, comparações e armazenamento de dados para uso futuro.
- (iii) **Saída:** Produz informações úteis na forma de documentos e relatórios, essenciais para apoiar a tomada de decisões e facilitar a comunicação.
- (iv) **Realimentação:** Utiliza a saída para ajustar a entrada ou as atividades de processamento. Por exemplo, erros ou problemas identificados na saída podem exigir correção nos dados de entrada ou alteração nos processos. Por exemplos erros ou problemas que poderiam criar tornar-se necessário corrigir os dados de entrada ou alterar um processo.

2.4.1 Tipologia de Sistemas de Informação

Existem diferentes tipologias de sistemas de informação que atendem a necessidades específicas das organizações. Segundo STAIR e REYNOLDS (2005), alguns exemplos incluem:

- (i) **Sistemas de Processamento de Transações (TPS):** São sistemas que automatizam o processamento de transações rotineiras de uma organização, como vendas, compras e registos de transações financeiras.
- (ii) **Sistemas de Informação Gerencial (MIS):** Esses sistemas fornecem informações de nível médio para ajudar na tomada de decisões gerenciais, como relatórios de desempenho, análise de tendências e projeções.

- (iii) **Sistemas de Apoio à Decisão (DSS):** São sistemas que auxiliam os tomadores de decisão em análises complexas e não rotineiras, fornecendo ferramentas e modelos para análise de dados e simulação de cenários.
- (iv) **Sistemas de Informação de Executivos (EIS):** Oferecem informações estratégicas para os executivos da organização.

Essas tipologias representam diferentes abordagens para a utilização de sistemas de informação nas organizações, cada uma com suas próprias características e benefícios específicos.

Para alcançar o objectivo geral, a tipologia escolhida foi: **Sistemas de Processamento de Transações (TPS)**, passo a explicar o porque da escolha.

Os sistemas de pagamento online lidam com transações financeiras eletrônicas, como pagamentos com cartão de crédito, transferências bancárias, e-wallets, entre outros.

Eles são responsáveis por processar essas transações de forma segura e eficiente, garantindo a integridade e a confidencialidade dos dados financeiros dos usuários.

Os TPS geralmente envolvem várias etapas, como autenticação do usuário, autorização da transação, processamento do pagamento e registo da transação.

Esses sistemas devem ser altamente confiáveis, escaláveis e seguros para lidar com grandes volumes de transações e proteger as informações financeiras dos usuários contra fraudes e violações de segurança.

Portanto, um sistema de pagamentos online se enquadra na categoria de TPS devido à sua natureza de lidar com transações financeiras eletrônicas em tempo real.

2.4.2 Informação

É um conjunto de factos organizados de modo a terem valor adicional, além do valor dos factos propriamente ditos. De acordo com (STAIR e REYNOLDS, 2005), “um gestor pode considerar que ter informações sobre todas as vendas é mais apropriado ou valioso para os seus objectivos ao invés de ter conhecimento das quantidades de vendas feitas de forma individual”.

O fornecimento de informações a clientes pode também auxiliar as companhias na melhoria dos seus resultados e aumento dos lucros.

A informação é constituída por um conjunto de dados com características específicas, ou seja, são um conjunto de dados significativos e relevantes para a componente ou sistema a quem se destinam.

2.4.3 Software

É o conjunto de todos os programas que possibilitam o funcionamento do próprio sistema e sua gestão ao mais baixo nível. Também pode ser definido como programas que comandam o funcionamento de um computador. (SOUSA, 2009).

São vários os tipos de *software* existentes e criados para vários fins, estes, normalmente são criados por empresas especializadas na criação de softwares, muitas das vezes, a criação dos mesmos em empresas ou instituições tem sido feita pelo departamento de informática da empresa para a utilização interna. Os *softwares* se subdividem em duas categorias, software do sistema e software de aplicação.

De acordo com PRESSMAN (1995), a descrição de *software* pode assumir a seguinte forma: (i) instruções (programas de computador) que, quando executadas, produzem a função e o desempenho desejados; (ii) estruturas de dados que possibilitam que os programas manipulem adequadamente a informação; e (iii) documentos que descrevem a operação e o uso dos programas.

2.4.3.1 Software do Sistema

Segundo (SOUSA, 2009), são designados por *software* do sistema todos programas responsáveis pelo funcionamento do computador e pela gestão de todo seu hardware. Por outro lado, pesquisadores como (STAIR e REYNOLDS, 2005) defendem que *software* constitui um conjunto de programas projectados para coordenar as actividades e as funções do *hardware* e diversos programas pelo sistema computacional.

Um pacote de *software* de sistemas em especial é projectado para um projecto específico da (Central Processing Unit) CPU e classe de hardware, por sua vez, o software do sistema subdivide-se em dois grandes grupos: os sistemas operativos e as linguagens de programação.

2.4.3.2 Software de aplicação

É representado pelo conjunto de programas que permitem ao computador executar tarefas práticas e úteis para o dia-a-dia do utilizador. (SOUSA, 2009).

Este é constituído por programas que facilitam ou ajudam o usuário a resolver problemas específicos de computação, o tipo de *software* em referência pode dar apoio a indivíduos, ou grupos e mesmo a empresas.

Esta aplicação informática tem maior potencial para afectar processos que agregam valor aos negócios.

A implementação e uso eficaz do *software* de aplicação podem proporcionar melhorias significativas e dar apoio aos objectivos corporativos, antes de um individuo, o grupo ou empresa decidir sobre a melhor abordagem para adquirir software de aplicação, objectivos e necessidades deveriam ser analisados cuidadosamente. Estes programas são normalmente designados por “aplicações ou *packages*”.

2.4.4 Arquitectura de *Software*

Com a evolução na área das Tecnologias de Informação e Comunicação, novos padrões de arquitectura para o desenvolvimento de sistemas web têm surgindo e têm sido colocados em prática no mundo inteiro pelos desenvolvedores de sistemas.

Uma aplicação web é um software que é instalado em um servidor web e é projectado para responder a solicitações, processar informações, armazenar informações e dimensionar as respostas de acordo com a demanda e, em muitos casos, é distribuído em vários sistemas ou servidores (Macedo, 2017).

No que concerne ao desenvolvimento de sistemas web, ou seja, *sistemas que rodam no servidor*, podendo ser acedidas por clientes em diferentes máquinas e locais.

Os três padrões de arquitectura mais comuns são: *cliente leve da web*, *cliente pesado da web e entrega pela web*. Neste estudo foi aplicado o padrão cliente leve da *Web*.

2.4.4.1 *Cliente leve da Web.*

Cliente leve da *Web* é um computador cliente em uma rede de modelo cliente-servidor de duas camadas o qual tem poucos ou nenhum aplicativos instalados, de modo que depende primariamente de um servidor central para o processamento de actividades.

O padrão de arquitectura “cliente leve da *Web*” é muito útil para aplicativos baseados na intranet, para os quais pode-se garantir apenas a configuração mínima no cliente.

Toda a lógica de negócio é executada no servidor durante o processamento das solicitações de página do navegador cliente.

Este padrão é mais adequado a aplicativos da *Web* baseados na *intranet* ou ambientes em que o cliente tenha uma capacidade mínima de computação ou não tenha nenhum controle sobre a configuração.

De acordo com MARCEL (2011), os principais componentes do padrão de arquitectura cliente leve da *Web* incluem o servidor web, conexão http, páginas html, páginas de servidor, servidor de aplicativos, navegador de cliente, que são detalhados a seguir.

- a) **Servidor Web** – é o principal ponto de acesso para todos os navegadores de cliente. Os navegadores de cliente na arquitetura cliente leve da Web, acessam o sistema somente por meio do servidor *Web*, que aceita pedidos de páginas da *Web* páginas em HTML estático ou páginas do servidor *Web*. Dependendo da solicitação, o servidor da *Web* pode iniciar algum processamento no próprio servidor. Se o pedido de página for para um módulo (Common Gateway Interface) CGI, (Internet Server API) ISAPI ou (Netscape Server API) NSAPI da página com scripts do servidor, o servidor *Web* delegará o processamento para o interpretador de *script* ou *módulo executável apropriado*. De qualquer forma o resultado será uma página em formato HTML apropriada para ser processada por um navegador HTML.
- b) **Conexão HTTP** – é o protocolo mais comum em uso entre os navegadores de clientes e os servidores da *Web*. Esse elemento de arquitetura representa um tipo de comunicação sem conexão entre cliente e servidor. Sempre que o cliente ou servidor enviar informações um para o outro, uma nova via camada de *soquetes de segurança* (SSL). Esse tipo de conexão criptografa as informações que estão sendo transmitidas entre cliente e servidor, usando a tecnologia de criptografia de chave pública/privada. Segundo TANENBAUM (2003), o HTTP é um protocolo da camada de aplicação do modelo *Open Systems Interconnection* (OSI), utilizado para transferência de dados na rede mundial de computadores, a World Wide Web. Este protocolo, utiliza a porta 80 e é usado para a comunicação de aplicações Web cuja comunicação é feita na linguagem HTML. Para que haja comunicação com o servidor, utiliza-se comandos próprios, os quais não estão na linguagem HTML.
- c) **Página HTML** – é uma página da *Web* interfaceada com usuário e informações de conteúdo que não passam por nenhum processamento no lado do servidor. Estas páginas geralmente contêm texto explicativo como, por exemplo: *direções e informações de ajuda, ou formulários HTML de entrada*. Quando o servidor da web recebe uma solicitação de uma página HTML, ele simplesmente recupera o arquivo e o envia para o “cliente solicitante”, sem filtrá-lo.
- d) **Página de servidor** – são páginas da *Web* que passam por alguma forma de processamento no lado do servidor. Em geral, essas páginas são implementadas no servidor como páginas com scripts, que são processadas por meio de um filtro no servidor do aplicativo ou de módulos executáveis (ISAPI ou NSAPI). Elas têm possibilidade de acesso a todos os recursos do servidor, incluindo componentes da

lógica de negócios, banco de dados, sistemas legados e sistemas de contabilidade comercial.

- e) **Servidor de aplicativo** – é o principal mecanismo para executar a lógica de modelo de negócio do lado do servidor. O servidor de aplicativo é responsável pela execução do código nas páginas do servidor. Ele pode ser executado no mesmo espaço de processamento. O servidor de aplicativo é, em termos de lógica, um elemento de arquitectura separado, pois está envolvido somente na execução da lógica do negócio e pode usar uma tecnologia completamente diferente da usada no servidor *Web*.

Navegador de cliente – é qualquer navegador (*HyperText Markup Language*) *HTML* padrão com capacidade para formulários. O navegador age como um dispositivo genérico da *interface* de usuário. Quando usado em uma arquitectura cliente leve da *Web*, o único outro serviço que ele fornece é a capacidade de aceitar retornar *cookies*. O usuário do aplicativo utiliza o navegador para solicitar páginas da *Web*: *HTML* ou servidor. A página retorna uma interface de usuário totalmente formatada com controlos de entrada de texto, que é convertida pelo navegador no modo de exibição do cliente. Todas as interações do usuário com o sistema são feitas por meio do navegador.

Na arquitectura cliente leve da *Web* alguns componentes opcionais mais comuns encontrados em aplicativos da *Web* estão ausentes, em especial o banco de dados.

A maioria de aplicativos da *Web* utiliza banco de dados. Às vezes, o banco de dados é também usado para armazenar as próprias páginas (essa utilização de banco de dados, representa um outro padrão de arquitectura). A figura 1, ilustra um diagrama da visão lógica da arquitectura Cliente leve da *Web*.

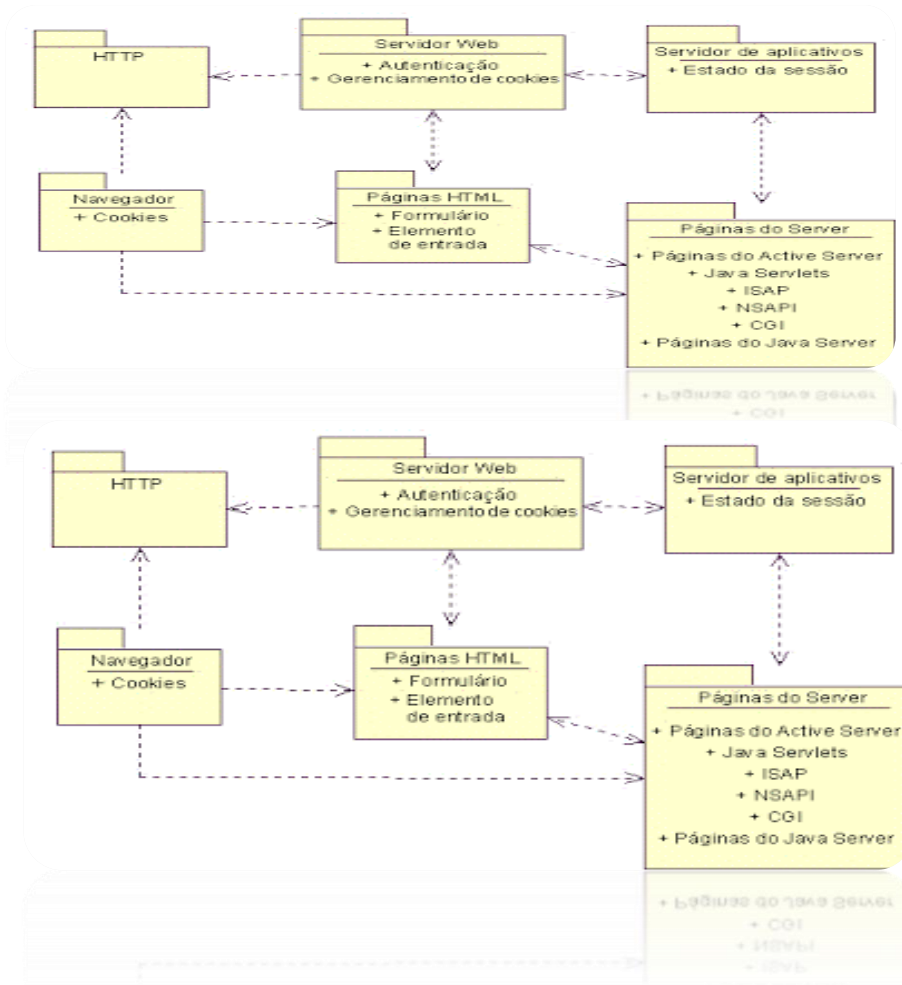


Figura 1: Visão lógica da arquitectura Cliente leve da Web Fonte: MARCEL (2011)

2.5 Metodologia de desenvolvimento de software

PRESSMAN (2002) define o modelo de desenvolvimento como uma representação abstrata do processo de desenvolvimento que define como as etapas relativas ao desenvolvimento de software serão conduzidas e inter-relacionadas para atingir o objetivo do desenvolvimento do software.

A seguir são apresentados alguns modelos de desenvolvimento de softwares:

2.5.1 Modelo Cascata

O modelo cascata segundo Cohn (2011) caracteriza-se por possuir uma tendência na progressão sequencial entre uma fase e a seguinte. Eventualmente, pode haver uma necessidade de revisitar uma fase anterior, mas de forma geral, as fases seguem fundamentalmente de forma sequencial.

O modelo cascata utiliza uma abordagem que inclui processo, métodos e ferramentas utilizadas pelos desenvolvedores de software. Tem como principal ideia de que as diferentes etapas de desenvolvimento seguem uma sequência.

De acordo com Sommerville (2011), o resultado de cada estágio é a aprovação de um ou mais documentos ('assinados'). O estágio seguinte não deve ser iniciado até que a fase anterior seja concluída. Na prática, esses estágios se sobrepõem e alimentam uns aos outros de informações. Durante o projeto, os problemas com os requisitos são identificados; durante a codificação, problemas de projeto são encontrados e assim por diante.

2.5.2 Modelo Incremental

Este modelo divide o desenvolvimento do software em incrementos menores e entregáveis, com cada incremento adicionando funcionalidades ao sistema. Cada incremento passa por todas as fases do ciclo de vida do desenvolvimento.

De acordo com Sommerville (2011), o modelo incremental é baseado na ideia de desenvolver uma implementação inicial, expô-la aos comentários dos usuários e continuar por meio da criação de várias versões até que um sistema adequado seja desenvolvido. Cada incremento ou versão do sistema incorpora alguma funcionalidade necessária para o cliente. Frequentemente, os incrementos iniciais incluem a funcionalidade mais importante ou mais urgente.

Significa que o cliente pode avaliar o sistema em um estágio relativamente inicial do desenvolvimento para ver se ele oferece o que foi requisitado.

Segundo Fagundes (2015), o modelo incremental foi proposto justamente para ser a resposta aos problemas encontrados no Modelo em Cascata. Segundo essa abordagem, um processo de desenvolvimento, divide o desenvolvimento de um produto de software em ciclos. E em cada ciclo de desenvolvimento, podem ser identificadas as fases de análise, projecto, implementação e testes.

2.5.3 Modelo espiral de Boehm

Segundo Cohn (2011), consiste no desenvolvimento em uma sequência de iterações e nisso cada iteração corresponde a uma volta na espiral e cada fase ou atividade é um setor, um "ângulo" da volta.

Este modelo foi proposto por Boehm (1988). Aqui, o processo de software é representado como uma espiral, e não como uma sequência de atividades com alguns retornos de uma para outra. Cada volta na espiral representa uma fase do processo de software. Dessa forma, a volta mais interna pode preocupar-se com a viabilidade do sistema; o ciclo seguinte, com definição de requisitos; o seguinte, com o projeto do sistema, e assim por diante. O modelo em espiral combina prevenção e tolerância a mudanças, assume que mudanças são um resultado de riscos

de projeto e inclui atividades explícitas de gerenciamento de riscos para sua redução, (Sommerville ,2011).

2.5.4 Metodologias de Desenvolvimento Ágil

Na década de 1980 e início da de 1990, havia uma visão generalizada de que a melhor maneira para conseguir o melhor software era por meio de um planejamento cuidadoso do projecto, qualidade da segurança formalizada, do uso de métodos de análise e projecto apoiado por ferramentas CASE (Computer-aided software engineering) e do processo de desenvolvimento de software rigoroso e controlado, (Sommerville ,2011).

Como resultado, Cohn (2011) afirma que muitas empresas estão tentando tornar suas equipes de desenvolvimento mais ágeis. Certamente, equipes ágeis tendem a introduzir seus produtos no mercado com muito mais rapidez, levando em consideração principalmente a satisfação dos clientes.

Existem várias metodologias deste modelo, como:

- (i) **eXtreme Programming (XP)**: é uma metodologia ágil para equipes pequenas e médias e que irão desenvolver software com requisitos vagos e em constante mudança;
- (ii) **Scrum**: processo de desenvolvimento iterativo e incremental para o gerenciamento de projetos e desenvolvimento de software ágil;
- (iii) **Feature Driven Development (FDD)**: é uma metodologia ágil de desenvolvimento que combina as práticas de gerenciamento ágil com abordagens à engenharia de software orientada a objetos;

Porém Optamos por adotar o **Extreme Programming (XP)** para este projeto, devido à sua abordagem rigorosa de práticas de engenharia e entrega contínua de software de alta qualidade. O XP enfatiza a colaboração próxima da equipe, testes automatizados e integração contínua, o que nos permite responder rapidamente às necessidades do cliente e entregar valor de forma eficaz.

2.5.4.1 Extreme Programming (XP)

Segundo Pressman (2006), a metodologia Extreme Programming (XP) é a abordagem ágil de desenvolvimento mais amplamente adotada pelos desenvolvedores de software. Apesar das variações entre as metodologias ágeis, todas elas compartilham a mesma essência, enfatizando um processo de desenvolvimento leve, iterativo e incremental (Ramos, 2013).

A metodologia Extreme Programming (XP) foi criada por Kent Beck durante o final da década de 1990. Beck desenvolveu a XP enquanto trabalhava em projetos de desenvolvimento de software na Chrysler Corporation. Ele procurava uma abordagem mais eficaz para lidar com os desafios frequentes encontrados no desenvolvimento de software, como a necessidade de adaptação rápida às mudanças nos requisitos do cliente e a garantia de alta qualidade do código. De acordo com Sommerville (2007), os desenvolvedores começaram a propor métodos ágeis de desenvolvimento devido à sua insatisfação com o uso de metodologias inadequadas em pequenos projetos. Eles perceberam que o tempo dedicado ao desenvolvimento era significativamente menor do que o tempo gasto na elaboração do projeto e na documentação do sistema.

Em 2001, Kent Beck e outros 16 renomados desenvolvedores, autores e consultores da área de software [Bec01] (batizados de “Agile Alliance” – “Aliança dos Ágeis”) assinaram o “Manifesto para o Desenvolvimento Ágil de Software” (“Manifesto for Agile Software Development”). Ele declarava:

Ao desenvolver e ajudar outros a desenvolver software, desvendamos formas melhores de desenvolvimento. Por meio deste trabalho passamos a valorizar: (i) indivíduos e interações acima de processos e ferramentas; (ii) software operacional acima de documentação completa; (iii) colaboração dos clientes acima de negociação contratual; e (iv) respostas a mudanças acima de seguir um plano, ou seja, embora haja valor nos itens à direita, valorizaremos os da esquerda mais ainda durante o desenvolvimento do software por possuírem maior importância (Pressman, 2011).

A Extreme Programming (Programação Extrema) emprega uma metodologia orientada a objetos como seu paradigma de desenvolvimento e envolve um conjunto de regras e práticas constantes no contexto de quatro atividades metodológicas: planejamento, projeto, codificação e testes (Pressman, 2011). A Figura 2, ilustra o processo XP e destaca alguns conceitos e tarefas-chave associados a cada uma das atividades metodológicas.

(i) Planejamento: A etapa de planejamento (também chamada de o jogo do planejamento) se inicia com a atividade de "ouvir" o cliente, uma atividade de levantamento de requisitos que capacita os membros técnicos da equipe XP a entender o ambiente de negócios do software e permite obter uma percepção ampla sobre os resultados solicitados, fatores principais e funcionalidade (Pressman, 2011). Para garantir a eficácia dessa atividade, é essencial que o cliente esteja profundamente envolvido no desenvolvimento, assumindo um papel integral como membro ativo da

equipe. Isso implica em sua presença constante ao longo do processo de desenvolvimento (Sommerville, 2007).

- (ii) **Projecto:** O projeto XP adota firmemente o princípio KISS (keep it simple, stupid), enfatizando a simplicidade sobre a complexidade. Prioriza-se um projeto simples em detrimento de representações mais complicadas. Além disso, um sistema planejado de forma simples pode ser rapidamente adaptado às mudanças, proporcionando redução de custos e tempo durante seu desenvolvimento, sem deixar de possuir recursos para atender às necessidades do cliente (Pressman, 2011).
- (iii) **Codificação:** Um conceito-chave na atividade de codificação (e um dos mais discutidos aspectos da XP) é a programação em pares. A XP recomenda que duas pessoas trabalhem juntas em uma mesma estação de trabalho. Isso fornece um mecanismo para solução de problemas em tempo real e garantia da qualidade em tempo real (o código é revisto à medida que é criado) (Pressman, 2011). O desenvolvimento não envolve sempre a mesma dupla de programadores, pois o intuito da programação em pares é que os pares sejam criados dinamicamente, possibilitando aos membros trabalharem com outros diferentes em uma programação em pares no decorrer do projeto, aumentando a união e disseminação das experiências da equipe de desenvolvimento (Sommerville, 2007).
- (iv) **Testes:** Uma das principais diferenças da XP em relação às metodologias convencionais é a maneira de como é testada a aplicação. Na XP, a ênfase dada à fase de testes é maior, pois o desenvolvimento iterativo não gera uma documentação detalhada do sistema, impossibilitando que uma equipe externa desenvolva os testes do software (Sommerville, 2007).

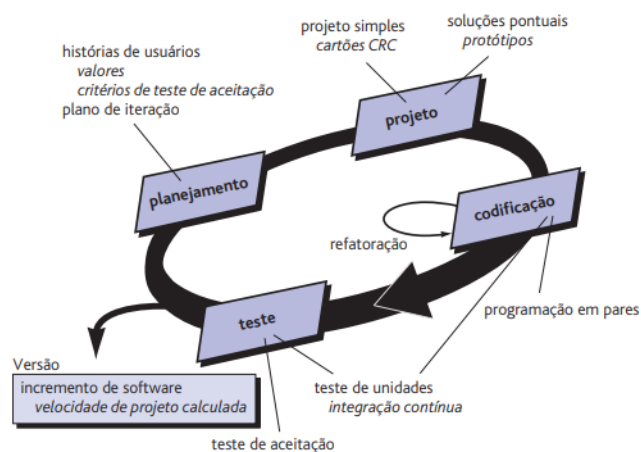


Figura 2- processo da Extreme Programming (XP)

Fonte: Pressman (2011, p.88)

Segundo Sommerville (2007) as principais características de teste na XP são:

- i. Desenvolvimento test-first
- ii. Desenvolvimento incremental de teste a partir de cenários;
- iii. Envolvimento do usuário no desenvolvimento e validação de testes;
- iv. O uso de ferramentas de teste automatizadas.

2.6 Engenharia de Requisitos de Software

Requisito de software é qualquer condição ou capacidade que deve ser implementada por determinado *software* ou componente deste para alcançar determinado fim (ENGHOLM, 2010).

Requisitos de um *software* são descrições sobre seu comportamento, funções e especificações das operações que deve realizar, como também das suas propriedades e atributos.

Os requisitos de *software* englobam a área de estudo relacionada com a criação, análise, documentação, rastreabilidade e validação (KOSCIANSKI & SOARES, 2007).

Os requisitos subdividem-se em: *funcionais* e *não funcionais*. Os requisitos *funcionais* - são os que ilustram o que o sistema deve fazer, e os *não-funcionais* ilustram os aspectos de qualidade ou restrições.

2.6.1 Requisitos funcionais

Ao fazer-se o levantamento de requisitos funcionais deve se ter em conta o tipo de *software* a ser desenvolvido e seus possíveis usuários. Quando estes são expressos como requisitos de utilizador, normalmente apresentam uma forma altamente abstracta, para possibilitar ao máximo o entendimento do mesmo por parte dos interessados pelo projecto. Estes, também podem ser descritos como requisitos funcionais de sistema, acabando por ser mais específicos, descrevendo em detalhes as funções do sistema, suas entradas e saídas, excepções, dentre outras características, (SOMMERVILLE, 2011).

O levantamento de requisitos deve ser completo e consistente, de forma que ao apresentar os mesmos não resulte definições contraditórias. A seguir, apresentamos um exemplo de alguns requisitos funcionais de um sistema:

Exemplo: Destacar que a pesquisa de dados no sistema pode ser realizada utilizando o nome, número de celular ou número de estudante como critérios de busca.

2.6.2 Requisitos não funcionais

Os requisitos não funcionais normalmente especificam ou restringem as características de um sistema como um todo, sendo frequentemente mais críticos que requisitos funcionais individuais. Eles surgem das necessidades dos interessados, devido a restrições de orçamento, políticas organizacionais, necessidade de interoperabilidade com outros sistemas de *software* ou hardware, ou a partir de factores externos, como regulamentos de segurança ou até legislações de privacidade, (SOMMERVILLE, 2011, p. 60).

Para ilustrar os requisitos não funcionais de um sistema, consideremos o seguinte exemplo: durante o processo de carregamento, é fundamental que o sistema exiba uma animação. Essa animação tem o propósito de proporcionar aos usuários um feedback visual, indicando que o carregamento está em andamento. Esta é uma exigência relevante para garantir uma experiência do usuário mais amigável e informativa.

2.7 Tecnologias e Ferramentas de Desenvolvimento

Nesta seção, discutiremos as tecnologias e ferramentas que foram empregadas no desenvolvimento do sistema proposto. A escolha adequada de tecnologias é crucial para o sucesso do projeto, influenciando diretamente aspectos como desempenho, manutenção e escalabilidade. A seguir, detalharemos as principais tecnologias e ferramentas utilizadas:

2.7.1 Linguagem de Modelagem Unificada – UML

É uma linguagem para especificação, construção, visualização e documentação de artefactos de um sistema de *software* (SILVA e VIDEIRA, 2001).

De maneira idêntica, Pender (2014), define a UML (Unified Modeling Language) como uma linguagem de especificação, documentação, visualização e desenvolvimento de sistemas orientados a objectos.

A UML permite que a equipe de desenvolvimento visualize os seus trabalhos em diagramas.

GUDWIN, 2015 sustenta que a Linguagem UML (Unified Modeling Language) é uma linguagem de modelagem que foi inventada visando-se a criação de modelos abstractos de processos.

Em princípio, não existem restrições quanto aos tipos de processos sendo modelados, estes modelos podem ilustrar processos do mundo real, processos de desenvolvimento de *software* assim como detalhes internos do próprio *software*.

Um dos grandes problemas enfrentados pelos desenvolvedores de sistemas nas fases de análise de requisitos e de análise de sistemas, era a não existência de uma notação padronizada e realmente eficaz, que pudesse abranger a qualquer tipo de aplicação que se deseje.

Aquando do surgimento da UML, muitos desenvolvedores ficaram entusiasmados devido a padronização que este propõe, porque era o tipo de modelo que eles esperavam.

O UML não se aplica somente na padronização de uma notação, mas também é útil no desenvolvimento de novos conceitos normalmente usados e também na modelação de objectos no estado da arte.

Particularmente no que se refere a engenharia de *software*, a linguagem UML pode ser utilizada para modelar todas as etapas do processo de desenvolvimento de software, bem como produzir todos os artefactos de software necessários a documentação dessas etapas.

Em termos práticos a UML é a convergência de diversas outras linguagens de modelagem utilizadas em diferentes processos de desenvolvimento de *software* propostos até então. Apesar de ser uma linguagem formal, a UML também é uma linguagem visual que tem como base diferentes tipos de diagramas (GUEDES, 2011).

A questão da existência de mecanismos de extensão, é uma das características interessantes da UML, estes mecanismos de extensão chamados *profiles* permitem que a UML possa ser estendido resultando na criação de novos tipos de diagramas. Diferentes tipos de *profiles* podem ser construídos.

Com base nos vários diagramas, a UML permite a definição e *design de threads* e processos, que permitem o desenvolvimento de sistemas distribuídos ou de programação concorrente. Também, permite a utilização dos chamados *patterns* e as descrições de colaborações.

Um dos tipos de diagramas particularmente úteis para modelarmos processos é os chamados diagramas de actividades. Por meio deles, é possível especificar uma sequência de procedimentos que compõem um processo no mundo real. Diagramas de actividade podem, portanto, ser utilizados para descrever o processo de desenvolvimento de *software*.

Outra característica da UML é a possibilidade que ela oferece de se poder criar uma descrição hierárquica dos processos, o que permite identificar se os processos são inúteis ou redundantes no modelo, descrevendo o relacionamento entre diferentes níveis de abstracção do mesmo.

Para implementar esse refinamento, cria-se um expediente para se definir abstractamente os componentes de um determinado modelo por meio de suas interfaces. Desta forma esses componentes são definidos somente em função das suas entradas e saídas.

Por fim, a semântica dos diagramas UML é determinada por uma linguagem de restrição chamada OCL, que determina de maneira não ambígua a interpretação a ser dada a seus diagramas.

Todas essas características fazem da linguagem UML uma linguagem de modelagem moderna, eficiente e conveniente para o desenvolvimento de especificações e definições com relação ao sistema em desenvolvimento (GUDWIN, 2015).

2.7.2 Elementos de Modelagem

Diagramas podem ser entendidos, de maneira simplificada, como grafos contendo nós, caminhos entre outros. Estes nós, podem ser apresentados em forma de símbolos gráficos. (GUDWIN, 2015).

Os caminhos podem ser representados por linhas contínuas ou descontínuas com possíveis decorações especiais nas pontas, e por fim, os textos podem aparecer em diferentes posições, dependendo da posição em que se encontrar, estes podem ter um significado diferente.

Esta característica espelha algum tipo de relacionamento entre os elementos envolvidos. Por exemplo, o relacionamento de conexão entre elementos do diagrama, modela algum tipo de relação entre os elementos conectados.

Relacionamentos de conexão, normalmente são representados por meio de linhas que ligam figuras bidimensionais, o relacionamento de inclusão, por exemplo de um símbolo, denota a inclusão de um determinado elemento em outro.

Por fim, a proximidade visual entre símbolos (por exemplo, um símbolo estar perto de outro símbolo ou figura bidimensional dentro de um diagrama), denota algum tipo de relação entre esses símbolos (GUDWIN, 2015).

Segundo GUDWIN (2015), os elementos de um diagrama podem ser: (i) **ícones** – são figuras gráficas de tamanho e formato fixo que não podem ser expandidas para ter algum tipo de conteúdo. Podem aparecer como participantes de um símbolo bidimensional, como terminadores de caminhos ou simplesmente isoladamente; (ii) **símbolos bidimensionais** - são figuras bidimensionais que podem ter tamanho variável e que podem ser expandidos de modo a conter outros elementos, tais como listas de *strings* ou outros símbolos bidimensionais. Podem ser divididos em compartimentos de tipo similar ou diferente; (iii) **caminhos** - são sequências de linhas cujas extremidades estão conectadas a outros elementos. Caminhos podem ter terminadores; e (iv) **strings** - são textos apresentando diversas informações cujo significado depende de onde aparecem.

2.7.3 Diagramas UML

Os diagramas UML podem, de uma maneira geral, se servir de todos estes expedientes de modelagem: estereótipos, *tagged values*, notas, pacotes, subsistemas e restrições. (GUDWIN, 2015). Segundo Sampaio (2007), existem treze tipos de diagramas da UML que estão divididos em três categoria: (i) **diagramas estruturais** – Inclui os diagramas de objecto, de classe, de componentes, de instalação, de pacotes e de estrutura; (ii) **diagramas comportamentais** – Inclui os diagramas de caso de uso, de transição de estados e de actividades; e (iii) **diagramas de interacção** – Inclui nesta categoria os diagramas de sequência, de interactividade, de comunicação e de tempo.

Neste trabalho usaremos os diagramas de classes, de caso de uso, de estado, de actividade e de sequência.

2.7.3.1 Diagrama de caso de uso

GUDWIN (2015) afirma que os diagramas de caso de uso são abstrações de pequenas histórias narrativas envolvendo a interacção entre um ou mais usuários (chamados de actores) e o sistema. A ideia é que estes casos de uso representem, por meio dessas pequenas histórias, as funcionalidades de um sistema.

São criados um conjunto de actores necessários para operar o sistema, e em seguida começa-se a descrever o fluxo dos acontecimentos, onde o actor executa uma acção e o sistema responde de alguma maneira a essa acção.

Para Nunes & O’Neil (2001), os diagramas de casos de uso utilizam as seguintes abstrações de modelação:

- a) **Actores** – representam uma entidade externa que interage com o sistema a ser desenvolvido, podendo ser um utilizador no sentido restrito da palavra ou um equipamento informático.
- b) **Caso de uso** – procura identificar a forma como se responde a um cliente ou evento em termos de processo ou caracterizar as funcionalidades que o sistema deve disponibilizar.
- c) **Relações** – é a forma de descrever a associação entre os casos de uso.

2.7.3.2 Diagrama de classes

Este diagrama ilustra e define a estrutura das classes utilizadas pelo sistema, determinando os atributos e métodos de cada classe (GUEDES, 2011).

Este diagrama provavelmente é o mais utilizado e um dos mais importantes da UML, serve de apoio para a maioria dos demais diagramas.

De uma forma simples, os diagramas de classes são grafos conectados por diversos tipos de relacionamentos estáticos. Estes ainda podem conter pacotes e outros tipos de elementos gerais. Em princípio, estes diagramas representam uma visão do modelo estrutural estático (GUDWIN, 2015).

De acordo com Nunes & O'Neill (2001), um diagrama de classe é composto pelos seguintes elementos: (i) Classes de objectos; (ii) Relações de Associação e Generalização; e (iii) Multiplicidade.

Nunes & O'Neill (2001), definem uma classe como uma representação de um conjunto de objectos que partilham a mesma estrutura e comportamento. Definem um atributo como sendo uma característica que os objectos possuem e que é representada por valores e os métodos como representação lógica do comportamento de um objecto.

Os atributos e métodos podem assumir 3 níveis de visibilidade: público (+) em que qualquer classe tem acesso ao elemento, protegido (#) em que qualquer descendente da classe pode aceder o elemento e privado (-) em que apenas a própria classe tem acesso ao elemento.

2.7.3.3 *Diagrama de sequência*

Um diagrama de sequência mostra uma interacção na forma de uma sequência temporal de mensagens sendo enviadas entre instâncias (GUDWIN, 2015). Nunes & O'Neill (2003), afirmam que as mensagens trocadas entre objectos representam a invocação de um serviço (operação) disponibilizado por um objecto, com o objectivo de despoletar uma acção ou actividade. Nunes & O'Neill (2003), descrevem os tipos de mensagens trocadas entre os objectos como:

- a) Mensagem síncrona: significa que o objecto emissor fica suspenso à espera de uma resposta, retomando posteriormente o controlo, e é usada quando o objecto emissor necessita de dados provenientes do objecto receptor, para continuar o seu processamento;
- b) Mensagem assíncrona: permite à operação emissora prosseguir o seu processamento, e é útil para ilustrar sistemas com processos concorrentes;
- c) Mensagem simples: é útil para quando ainda não está definido o tipo de mensagem ou o tipo não relevante;

- d) Mensagem de retorno: é usada para ilustrar o retorno da mensagem enviada que poderá ser um valor ou um sinal, esta é implícita para mensagens simples ou síncronas, então a sua representação é opcional.

Em particular, este ilustra as instâncias participando de uma interação por meio de *lifelines*, dessa forma um diagrama de sequência não clarifica as associações existente objectos, embora possamos concluir existindo uma mensagem de um objecto no diagrama é necessário que haja uma associação entre eles.

Os *lifelines* surgem com o intuito de representar de forma temporária o ciclo de vida de um objecto, executando desta forma um papel bem específico em uma interação. Assim sendo, cada objecto que participa na interação possui sua própria *lifeline*.

A representação de uma *lifeline* é feita por meio de uma linha vertical que sai de baixo do objecto e varrendo todo o espaço do diagrama. As setas entre diferentes *lifelines* denotam uma comunicação ocorrendo entre os objectos envolvidos em um instante de tempo.

A existência e a duração do objecto em um papel pode ser mostrada, mas o relacionamento entre os objectos não são demonstrados explicitamente como no caso dos diagramas de comunicação. Uma *lifeline* pode se dividir em duas ou mais *lifelines* concorrentes para expressar uma condicionalidade, ou seja, fluxos de sequência alternativos, dependentes de condições que são indicadas no diagrama

2.7.3.4 Diagrama de actividades

O diagrama de actividades é um diagrama UML utilizado para modelar o aspecto comportamental de processos. É o diagrama que mais sofreu mudanças em seu meta-modelo, desde seu surgimento na UML 1.0.

Neste diagrama, uma actividade é modelada como uma sequência estruturada de acções controladas potencialmente por nós de decisão e sincronismo. Em seu aspecto mais simples, um diagrama de actividades pode ser confundido com um fluxograma.

Entretanto, ao contrário de fluxogramas, os diagramas de actividades UML suportam diversos outros recursos, tais como as partições e os nós do tipo *fork* e *merge*, além da definição de regiões de interrupção, que permitem uma modelagem bem mais rica do que simplesmente um fluxograma (GUDWIN, 2015).

A partir da UML 2.0, foi considerado independente do diagrama de máquina de estados. O diagrama de actividade preocupa-se em descrever os passos a serem percorridos para a conclusão de uma actividade específica, podendo esta ser representada por um método com

certo grau de complexidade, um algoritmo, ou mesmo por um processo completo. O diagrama de actividade concentra-se na representação do fluxo de controlo de uma actividade.

2.7.4 Bootstrap

Desenvolvido por Mark Otto e Jacob Thornton, é um *framework* de *front-end* para o desenvolvimento rápido de aplicações e sites Web.

Este pacote contém HTML e modelos de *design* baseados em (Cascading Style Sheets) CSS para tipografia, *forms*, botões navegação e outros componentes de interface, bem como as extensões *JavaScript* operacionais.

Com compatibilidade garantida a todos os principais navegadores, suporta *Web design* responsivo. O que quer dizer que o *layout* da página Web ajusta-se dinamicamente, levando as características do dispositivo usado (*desktop*, tablet ou celular).

O *bootstrap* possui o código aberto e está disponível no *GitHub*, também é compatível com o HTML5 e CSS3, o que possibilita a criação de *layouts* responsivos, o que permite com que seu conteúdo seja organizado até 12 colunas que se comportam de maneira diferente para cada resolução.

Como qualquer outra ferramenta, *Bootstrap* possui vantagens e desvantagens, é importante conhecer e entender suas funcionalidades para saber os momentos e situações adequadas para utilizá-lo.

a) Vantagens:

- Possui documentação detalhada e de fácil entendimento
- É otimizado para o desenvolvimento de *layouts* responsivos
- Possui componentes suficientes para o desenvolvimento de qualquer *site* ou sistema Web com interface simples
- Facilita a criação e edição de *layouts* por manter padrões
- Funciona em todos os navegadores actuais (Chrome, Safari, Firefox, IE, Opera)

b) Desvantagens:

- Seu código terá de seguir os “padrões de desenvolvimento *Bootstrap*”
- Tema padrão e comum do *Bootstrap*: caso não se faça ajustes visuais, o projecto se parecerá com outros que também utilizam o *Bootstrap*
- Performance: por ser um *framework* grande contendo CSS, JS e *Fonts*, onde na maior parte dos casos, não irá utilizar nem a metade que o *framework* proporciona
- Dependência com *jQuery*

A estrutura é simples, o pacote contém três tipos diferentes de arquivos (*CSS*, *JavaScript* e *Fonts*), que vêm devidamente organizados em suas pastas. Montar um *layout* é fácil e rápido utilizando a documentação disponível no site.

Como toda a estrutura do *CSS* já vem definida, basta procurar o componente necessário e adicionar seu código (ou seja, copiar e colar) na sua página. Em poucos minutos o *layout* toma forma e está pronto para uso e o mesmo acontece com o *JavaScript*.

2.7.5 Sistema de Gestão de Banco de Dados

Segundo Elmasri & Navathe (2011), um sistema gerenciador de banco de dados (SGBD – *Database Management System*) é uma colecção de programas que permite aos usuários criar e manter um banco de dados.

O SGBD é um sistema de *software* de uso geral que facilita o processo de definição, construção, manipulação e compartilhamento de bancos de dados entre diversos usuários e aplicações.

Um sistema de gerenciamento de banco de dados implica a criação e manutenção de bases de dados, elimina a necessidade de especificação de definição de dados, age como interface entre os programas de aplicação e os ficheiros de dados físicos e separa as visões lógica e de concepção dos dados (Guimarães, 2011).

Segundo Guimarães (2011), existem basicamente três componentes de um SGBD:

- Linguagem de definição de dados (especifica conteúdos, estrutura a base de dados e define os elementos de dados);
- Linguagem de manipulação de dados (para poder alterar os dados na base);
- Dicionário de dados (guarda definições de elementos de dados e respectivas características).

Para Elmasri & Navathe (2011), as vantagens de usar a abordagem de um SGBD são:

- Controlo de redundância;
- Restringimento ao acesso não autorizado;
- Armazenamento persistente para objectos do programa;
- Oferecem *backups* e recuperação;
- Representam relacionamentos complexos entre dados;

O software que gerencia o dado é geralmente chamado de sistema gerenciador de banco de dados (SGBD). A seguir são apresentados alguns exemplos de sistemas de gerenciamento de banco de dados:

- Microsoft Sql Server;

- Acess;
- Oracle;
- MySql.

Para este trabalho será utilizado o SGBD – MySql gerenciador de banco de dados da Oracle, por ser uma ferramenta de manutenção de dados bastante simples e eficaz. As interfaces gráficas são bastante intuitivas, e sua licença é gratuita.

Este conjunto de ferramentas é utilizado em várias soluções corporativas e empresariais, possuindo boa aceitação do mercado no geral.

Segundo Niederauer (2006), o MySql é um SGBD relacional que utiliza a linguagem padrão SQL (Structured Query Language) e é largamente utilizado em aplicações para a Internet. É o mais popular entre os bancos de dados com código-fonte aberto.

Segundo hti (2016) as principais características do MySQL são:

- Escalabilidade e flexibilidade;
- Alto Desempenho;
- Suporte transaccional robusta;
- Protecção de Dados Strong;
- Liberdade Open Source.

2.7.6 Modelos de banco de dados

Segundo Elmasri & Navathe (2011), um modelo de banco de dados é uma coleção de conceitos que podem ser usados para descrever a estrutura de um banco de dados.

A Lucidchart.com (2019), sustenta afirmando que, um modelo de banco de dados para além de descrever a sua estrutura, inclui as relações que determinam como os dados podem ser armazenados e acedidos.

Guimarães (2011), apresenta quatro modelos de banco de dados a conhecer:

- Modelo Hierárquico;
- Modelo em Rede;
- Modelo Orientado a Objectos;
- Modelo Relacional.

De entre os vários modelos de banco de dados existentes, neste trabalho será utilizado o modelo relacional.

2.7.6.1 Modelo Relacional

O modelo relacional classifica dados em tabelas, também conhecidas como relações, cada uma das quais consiste em colunas e linhas. Cada coluna lista um atributo da entidade em questão, como ano de fabrico do veículo, matrícula, etc.

Para Elmasri & Navathe (2011), entidade é algo do mundo real com uma existência física (por exemplo, uma pessoa em particular, um carro, uma casa ou um funcionário), ou pode ser um objecto com existência conceitual (por exemplo, uma empresa, um cargo ou um curso universitário). Cada entidade possui atributos, que são as propriedades específicas que a descrevem.

3 METODOLOGIA

Esta secção apresenta a descrição dos procedimentos metodológicos que orientaram esta pesquisa para o alcance dos objectivos. Deste modo serão discutidas as implicações dos fundamentos teóricos e os objectivos do trabalho na escolha da metodologia de pesquisa, que a posterior será descrita e classificada de acordo com os métodos utilizados na recolha de dados.

3.1 Tipo de pesquisa

Para a concretização desta pesquisa seguiu-se o método da pesquisa mista, ou seja, combinação da pesquisa qualitativa e quantitativa.

De acordo com Creswell (2007, p.27) mista " é uma abordagem de investigação que combina ou associa a forma qualitativa e quantitativa".

Justifica-se a escolha desse método de modo a minimizar possíveis dificuldades na conjugação de prática de investigação quantitativa e qualitativa, tais pesquisas podem produzir resultados relevantes, assim como podem orientar caminhos promissores a serem explorados por pesquisadores.

No estudo em causa entende-se que as pesquisas qualitativas e quantitativas respondem de forma aprofundada o conhecimento da realidade, pelo facto de debruçar a razão da existência de certas coisas.

Para o efeito, fez-se um estudo de campo que, para Gil (2002), focaliza uma comunidade e desenvolve-se, por meio da observação directa das actividades do grupo do estudo e aplicação de entrevistas à informantes, na perspectiva de captar suas expirações e interpretações do que ocorre no grupo, como é o caso das regras, costumes e convenções que intervêm no processo de suas transformações.

No presente trabalho o estudo de campo foi realizado no Município de Maputo, o desenvolvimento do mesmo trabalho, foi por meio da observação directa e com a aplicação das entrevistas a informantes constituídos por munícipes condutores de veículos (*vide apêndice 2-roteiro de entrevista*).

3.2 População

Segundo Richardson (1999), "população ou Universo da pesquisa é o conjunto de elementos que possuem determinadas características, ou ainda, conjunto de indivíduos que trabalham em um mesmo lugar".

Assim, constituiu população ou universo deste trabalho, todos os munícipes condutores de veículos do Município de Maputo.

3.3 Amostra

Definida como sendo um subconjunto do universal da população, por Richardson (1999), nesta pesquisa recorreu-se a uma amostragem por conveniência, onde os elementos foram seleccionados com base em sua disponibilidade e acessibilidade.

De acordo com Oliveira e Ferreira (2014), no estudo de caso em questão, a seleção da amostra assume uma abordagem altamente específica, buscando deliberadamente uma ampla variedade em vez de uniformidade. Para o propósito deste estudo, a amostra consistiu em 21 munícipes que são condutores de veículos, os quais responderam a um inquérito online compartilhado via WhatsApp e e-mail.

Dentre os 21 munícipes que participaram do inquérito, 17 são do sexo masculino e os restantes 4 do sexo feminino. A seleção desses participantes foi intencional, baseada no critério de todos serem condutores de veículos. Quanto ao local e período da pesquisa, também houve uma seleção intencional. O local escolhido foi onde tradicionalmente ocorre o processo de cobrança do Imposto Autárquico sobre Veículos (IAV) na cidade de Maputo, o que facilitou a localização específica dos participantes para a coleta de dados. Quanto ao período, ele foi determinado pela entidade competente para o processo de cobrança do IAV.

3.3.1 Técnicas de recolha e organização de dados

Os dados que foram recolhidos são, em grande medida, dados qualitativos. Salientar que, nesta busca foram recolhidos dados primários e secundários.

Os primários constituem aqueles levantados antes da pesquisa e os dados secundários são aqueles levantados directamente no campo, durante a pesquisa.

3.3.2 Dados Primários

No processo de recolha de dados, o pesquisador recorreu às várias técnicas apropriadas da pesquisa qualitativa, nomeadamente: relatórios, questionário, inquérito, entrevistas e observação.

Devido a extensão dos intervenientes no processo e das limitações a vários níveis tais como: a natureza da pesquisa, o tempo e as limitações financeiras do pesquisador, não foi possível administrar o questionário a todos os técnicos de informática e correctores.

3.3.3 Dados Secundários

A utilização destes diferentes instrumentos constitui uma forma de obtenção de dados de diferentes fontes e tipos, os quais proporcionam a possibilidade de cruzamento/triangulação de informação.

3.3.4 Colecta de Dados

Para o efeito de recolha de dados, foram inquiridos 21 condutores de veículos que correspondem a 100 por cento da amostra. Estes foram escolhidos de forma aleatória sendo 17 homens que corresponde a 80,95 por cento e 7 mulheres que corresponde a 19,05.

Os inquéritos foram preenchidos por 15 condutores na ausência do pesquisador, de forma individual, de acordo com a disponibilidade de tempo de cada um dos inquiridos, de modo a não prejudicar o decurso normal das suas actividades laboral e para não influenciar as repostas dos respondentes.

Para os outros 06 inquiridos foram preenchidos na presença do pesquisador e no momento de de conversa descontraído uma vez esses inquiridos contavam com tempo disponível.

O trabalho de campo para a recolha de dados foi autorizado pelo Conselho Municipal da Cidade de Maputo (CMCM). A aplicação dos instrumentos de recolha de dados foi aceite por todos inquiridos.

4 APRESENTAÇÃO DE RESULTADOS

Nesta seção, apresentaremos os resultados obtidos a partir da análise dos dados coletados durante a realização deste estudo. Os resultados são apresentados em relação aos objetivos de pesquisa estabelecidos, buscando fornecer uma compreensão clara e abrangente das descobertas alcançadas.

Iniciamos esta seção com uma visão geral dos dados coletados, destacando características-chave da amostra, como tamanho, demografia e outras informações relevantes sobre os participantes do estudo. Em seguida, fornecemos uma análise descritiva dos dados, incluindo medidas resumidas e uma exploração inicial das tendências observadas.

4.1 Caracterização de Município de Maputo – local de estudo

O Município de Maputo situa-se na baía de Maputo a Norte do Estuário do Rio Maputo. Ocupa uma superfície de cerca de 383 km² que se distribuem entre o Norte e o Sul da baía, onde desaguam os rios Tembe, Matola, Umbelúzi e Infulene.

A população do Município de Maputo é de 1.127.565 habitantes (Perfil Estatístico do CMM - 2021), distribuída por sete Distritos Municipais (DM) e 63 bairros.

A missão do Município é de liderar o processo de elevação da qualidade de vida dos munícipes, criação de um ambiente atractivo aos investimentos e à geração de emprego, através da melhor prestação de serviços, da mobilização dos munícipes e da acção coordenada entre os diversos intervenientes. Dentro dessa missão possui como uma das suas actividades a área dos transportes e comunicação, onde encontramos a regulamentação do Imposto Sobre Veículos do Decreto nº 19/2002 de 23 de julho aprovado pelo conselho de ministros.

Os postos de conbrança podem ser localizados nos seguintes pontos:

Distrito Municipal KaMpfumu:

- Sede da Administração: Av. Eduardo Mondlane (esquina com Av. Tomás Nduda) Nº 511;
- Av. Karl Marx. Nº 173, R/C;
- Aterro da Maxaquene – Esquina entre Av. Zedequias Manganhela e Rua do Pântano;
- Av. Albert Lithuli Nº 63 R/c
- Av. Fernão Magalhães Nº 1252 R/c
- Av. 24 de Julho Nº 1449 R/c

- Rua da Rádio N° 2 R/c

Distrito Municipal Nhlamankulu

- Sede da Administração: Rua Irmãos Rubby N° 57
- Av. Joaquim Chissano N° 923

Distrito Municipal KaMaxaqueni

- Sede da Administração: Rua Particular da Malhangalene, N° 75 R/c

Distrito Municipal KaMavota

- Sede da Administração: Rua da Beira N°688

Distrito Municipal KaMubukwana

- Sede da Administração: Rua são Pedro 55

Distrito Municipal KaTembe

- Estrada Principal N° 403
- Distrito Municipal KaNyaka
- Mercado Peixe KaNyaka

4.1.1 Estrutura orgânica do Conselho Municipal de Maputo

O Conselho Municipal é um órgão executivo colegial do Município e exerce os seus poderes em conformidade com a Constituição, a lei e os seus regulamentos. Este órgão é constituído por um presidente eleito por sufrágio universal, igual, directo e secreto, para mandato de cinco (5) anos e por dezassete (17) vereadores por ele designados, sendo que sete (7) dirigem os Distritos Municipais. O CMM compreende também quadros de apoio ao presidente do CMM.

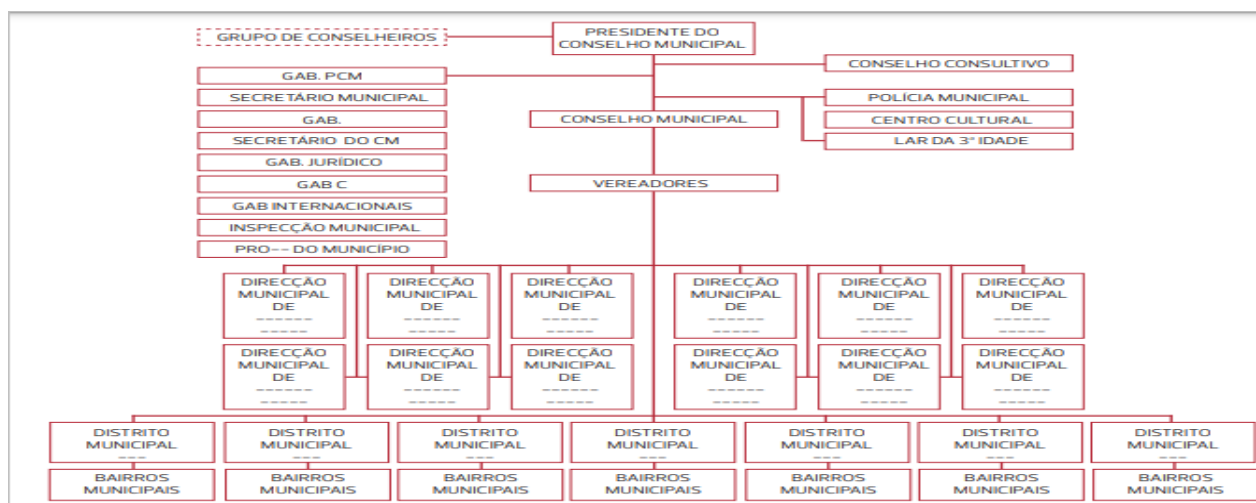


Figura 3: Estrutura orgânica do Conselho Municipal de Maputo Fonte: CMCM (2021)

4.2 Apresentação dos Resultados do Inquérito

Neste capítulo são apresentados e discutidos os principais resultados desta investigação. Os dados foram tratados e sistematizados através de um pacote estatístico o SPSS 2.0.

Os dados recolhidos por inquéritos, foram pré-codificadas e o seu tratamento foi analisado de forma comparativo com as respostas globais de diferentes categorias sociais, na base de Técnicas de redacção, de codificação e de exploração das perguntas, incluindo as escalas métricas e de atitude.

Procedeu-se à análise estatística descritiva dos dados obtidos uma vez que o principal objectivo do inquérito é servir de orientação para estabelecer as condições mais adequadas para realizar a cobrança de impostos.

Das 21 entrevistas realizadas, 17 eram do sexo masculino que corresponde a 80,95 por cento e 4 eram do sexo feminino que corresponde a 19,05 totalizando 100 por cento do tamanho de amostra com idades compreendidas entre 19 e 48 anos, residentes na cidade de Maputo.

Os resultados desta entrevista são apresentados em pontos de acordo com a estruturação do roteiro de entrevista elaborado pelo autor.

4.2.1 Dificuldades Enfrentadas

No que diz respeito as dificuldades enfrentadas nos postos de cobrança, verifica-se que maior parte da população (71,4%) depara-se com longas filas de espera. As dificuldades enfrentadas nos postos de cobrança estão representadas a figura 3. As longas filas de espera nos postos de cobrança é a dificuldade mais sofrida (71,4%), seguindo avarias de sistemas e erros de natureza humana (9,5%), deixar para depois (4,5%) e baixa qualidade no atendimento (5,1%).

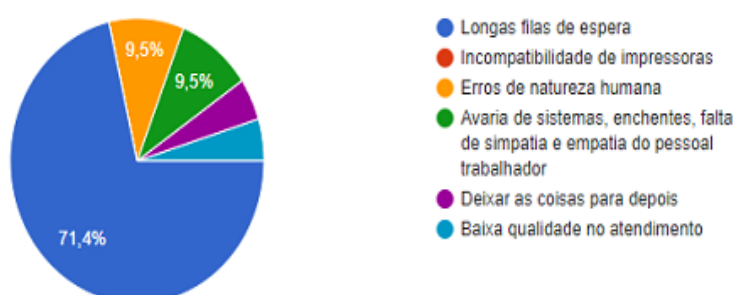


Figura 4: Distribuição percentual das dificuldades enfrentadas pela população nos postos de cobrança

4.2.2 Eficiência do Sistema

Em relação ao sistema de cobrança actual, é considerado razoável, tendo sido referido por 11 das pessoas inquiridas, que corresponde a 52,4%, seguido do (bom e péssimo) com uma percentagem de 23,8% para cada, que corresponde a 5 pessoas.

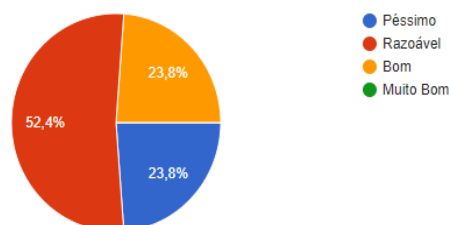


Figura 5: Distribuição percentual da opinião da população sobre o actual sistema de cobrança.

4.2.3 Atendimento

Tal como se observa na figura 5, a maior parte das pessoas (57,1%) considera o atendimento nos postos de cobrança razoável, seguido de 33,5% considera péssimo. Apenas 9,5% das pessoas inquiridas, referiram que o atendimento nos postos de cobrança é bom.

Das pessoas que afirmaram “enfrentar longas filas” de espera nos postos de cobrança, quando questionadas sobre o sistema actual, apenas 2 pessoas que corresponde 9.52 % afirmaram este é péssimo e 6 pessoas que corresponde a 28.57 % afirmaram, que o atendimento é péssimo.

Por conta disso surgiu a seguinte questão, a razão das longas filas nos postos de cobrança está aliada ao atendimento ou ao sistema de cobrança actual?

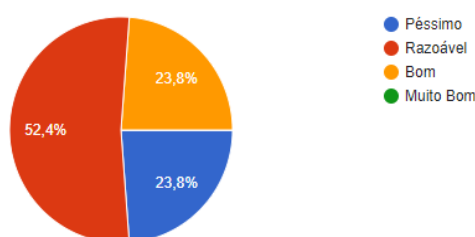


Figura 6: Distribuição percentual da opinião da população sobre o atendimento nos postos de cobrança.

Como pode se notar, o sistema actual (SCIVA), em uso no CMCM, não é flexível, pois com base nos dados colhidos este sistema, propicia as longas filas de espera, pois o pagamento ocorre dentro de um prazo previamente estabelecido, enquanto com a implementação deste sistema, não haverá longas filas pois o processo será realizado remotamente não havendo assim a necessidade do munícipe deslocar se até aos postos de cobrança.

Com base nestes resultados, vê-se a necessidade de se implementar uma solução em que o utilizador pode efectuar pagamentos em qualquer lugar sem ter de se fazer presente no posto

de cobrança e que não necessita da intervenção humana, com o objectivo de descongestionar os postos de cobrança, reduzindo desta forma as longas filas de esperas enfrentadas nos postos de cobrança.

Pode se afirmar ainda, que com a implementação do sistema proposto haverá uma melhoria na prestação de serviços e no atendimento ao munícipe, pois é possível ver com base nos dados colhidos, que os munícipes consideram mau o atendimento por parte dos colaboradores do CMCM.

Afirmativamente os munícipes estão convencidos que um novo sistema que permitisse lançar todos os documentos e validação de pagamentos online, tornaria o processo mais eficiente.

O Sistema actual tem um horário estabelecido das 7h30 min as 15h30min, o que diminuiu a janela de tempo para o atendimento. Em consequência disso, verifica se congestionamento. Contudo com a implementação do novo sistema, esta janela de tempo será maior pois estará disponível 24h sobre 24 h, o que ira permitir tenha mais tempo para efectuar o pagamento.

4.3 Características do sistema em uso para o processo de cobrança de impostos

O processo de cobrança de Impostos Autárquicos de Veículos (IAV) é submetido aos cidadãos por parte dos funcionários utilizadores de sistema, em recebedorias ou postos de cobranças. Sempre seguiu o mesmo padrão desde que o sistema foi concebido, a sua estrutura e regras básicas nunca foram alteradas. O processo de cobrança de IAV é baseado nas seguintes etapas:

- a) O cidadão apresenta-se perante o posto de cobrança, acompanhado do livrete do automóvel que deseja efectuar o pagamento, bilhete de identidade e número único de identificação tributária (NUIT).
- b) Identificado o cidadão o funcionário verifica o seu cadastro no sistema, verificado o cadastro no sistema, é emitido um aviso de pagamento
- c) Feito isso o cidadão recebe uma nota com preço a pagar e o respectivo número de conta para efectuar o depósito, ou pode pagar via POS.
- d) Em seguida, o cidadão dirige-se ao caixa de cobrança e efectua o pagamento e recebe o comprovativo de pagamento,
- e) Em seguida, o cidadão dirige-se à mesa de impressão com o respectivo comprovativo de pagamento;
- f) O funcionário consulta o aviso de pagamento emitido com base no código do cidadão e escolhe o documento efectua o pagamento no sistema e imprime o manifesto e entrega ao cidadão.
- g) Uma vez recebido o manifesto, o cidadão retira-se do local de cobrança.

A figura 7, mostra esse processo:

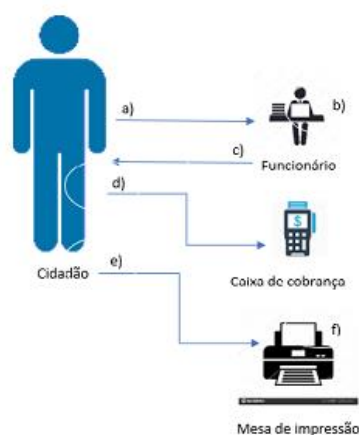


Figura 7: Processo de cobrança de imposto atual

4.4 Características do Sistema Proposto processo de cobrança de impostos

Baseado nos resultados da análise e descrição do actual processo de cobrança de impostos, constatou-se alguns constrangimentos, como por exemplo: longas filas de espera nos postos de cobrança, morosidade no processo de cobrança, o cidadão é obrigado a passar por vários sectores para obter o manifesto.

Todavia, vários dos problemas descritos anteriormente, decorrentes de tal processo poderiam ser evitados se uma solução que permitisse a cobrança de impostos a partir de qualquer lugar e a qualquer momento, como o caso de um sistema de cobrança de impostos *online*, acedido através de uma interface Web usando qualquer navegador e servir-se-á de uma base de dados relacional.

A princípio, os problemas acima descritos podem ser considerados mínimos em relação ao que o processo é capaz de fornecer. Com o sistema de cobrança *online* o cidadão poderá efectuar o pagamento de manifesto de qualquer lugar a qualquer momento sem ter de se deslocar até aos postos de cobrança. Para que tal solução vigore com sucesso, alcançando os objectivos desejados, a mesma deverá possuir meios de comunicação tanto com o servidor de banco de dados da instituição. O cidadão devera possuir um dispositivo electrónico com acesso à internet. A Figura 8, mostra o fluxograma do sistema proposto.

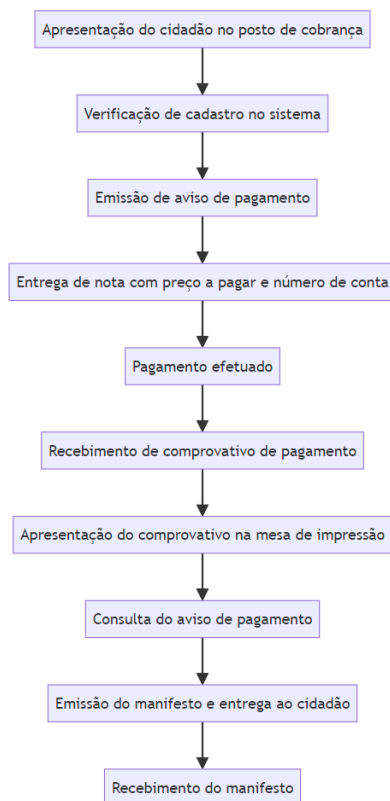


Figura 8: Fluxograma do sistema proposto

Desta forma, o acesso à aplicação se dará de maneira remota ao servidor de aplicações, que ficará responsável por gerir todas as operações de registo, até ao pagamento do manifesto. A figura 9 mostra a topologia de uma solução para a cobrança de imposto.

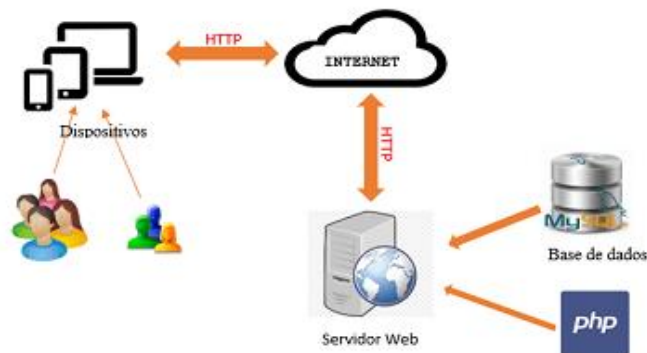


Figura 9: Modelo genérico do sistema proposto

4.5 Funcionalidades do Sistema

Nesta seção, descreveremos as funcionalidades principais do sistema, destacando suas capacidades e benefícios para os usuários. As funcionalidades são agrupadas de acordo com as áreas de atuação do sistema e são descritas de forma detalhada a seguir:

4.5.1 Requisitos Funcionais do Sistema

Tabela 1: Requisitos funcionais

Requisitos	Descrição
RF1	O sistema deve permitir a autenticação dos utilizadores.
RF2	O sistema deve permitir listar os tipos de automóveis, e permitir que os utilizadores escolham o automóvel que queiram efectuar o pagamento.
RF2	O sistema deve permitir emitir pagamento.
RF3	O sistema deve permitir consultar o estado de pagamento
RF4	O sistema deve permitir a consulta da situação do automóvel

4.5.2 Requisitos não funcionais

Tabela 2: Requisitos não funcionais

Requisitos	Descrição
RNF1	O sistema deve ser implementado na linguagem PHP com base de dados MySql.
RNF2	O sistema deve ser portátil, ou seja, deve ser utilizado em qualquer plataforma de hardware e de <i>software</i> .
RNF3	O sistema deve garantir impressões no formato PDF
RNF4	A interface do sistema de ser de fácil uso, esta deve tornar a experiência do utilizador fácil.

4.5.3 Diagrama de Casos de Uso

Os diagramas de caso de uso apresentam os processos realizados, este organiza e modela o comportamento do sistema, ilustra as funcionalidades do sistema, assim como os usuários que intervêm no sistema. Assim sendo, abaixo ilustramos os actores do sistema proposto:

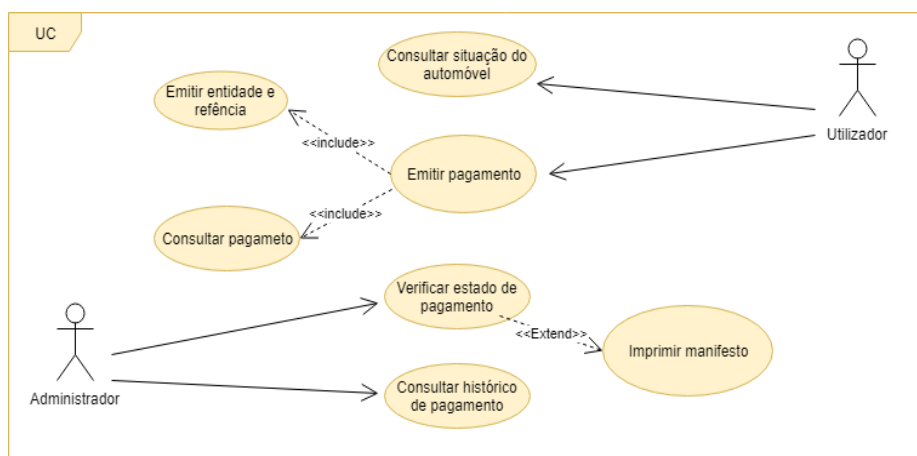


Figura 10: Diagrama de caso de uso

4.5.3.1 Descrição dos casos de uso

Tabela 3: Descrição dos casos de uso

Caso de Uso: Emitir pagamento	
Descrição:	Este caso de uso permite que os utilizadores emitam pagamento imposto.
Actores:	Utilizador
Pré-Condições:	- Automóvel cadastrado no sistema; - Autenticação do utilizador.
Pós-Condição:	- Obter talão de pagamento.
Fluxo Principal	
Acções do Actor	Sistema
1.Utilizador escolhe o tipo de automóvel que pretende fazer o pagamento; 3.Seleciona o automóvel que pretende efectuar o pagamento do registo e clica no botão "Emitir"; 5.Confirma; 7. Confirma os dados de pagamento.	2.Lista os automóveis associados ao utilizador; 4.Exibe uma tela pedindo a confirmação do utilizador; 6.Exibe uma tela com os dados de pagamento; 8. Exibe uma tela com entidade e referência

Tabela 4: Cenário do caso de uso consultar situação do automóvel

Caso de Uso: Consultar Situação do Automóvel	
Descrição:	Este caso de uso permite que o utilizador consulte a situação do automóvel
Actores:	Utilizador
Pré-Condições:	- Autenticação do utilizador; - Pagamento emitido; - Selecionar o automóvel;
Pós-Condição:	- Estado do automóvel verificado;
Fluxo Principal	
Acções do Actor	Sistema
1. Após seleccionar o automóvel, o cliente informa ano que pretende consultar e seleciona o botão “Consultar”;	2. Mostra uma tela com a situação do automóvel.

Tabela 5: Cenário do caso de uso consultar histórico de pagamento.

Caso de Uso: Consultar histórico de pagamento	
Descrição:	Este caso de uso permite que o administrador consulte o histórico de pagamentos.
Actores:	Administrador
Pré-Condição:	- Pagamentos emitidos;
Pós-Condição:	- Obter número de pagamentos emitidos.
Fluxo Principal	
Acções do Actor	Sistema
1. O administrador acede ao sistema; 3. Seleciona o botão;	2. O sistema uma exibe a tela principal, que apresenta um botão “Consultar histórico de pagamento”; 4. O sistema exibe uma tabela que contém a lista de todos os pagamentos efectuados no sistema.

Tabela 6: Cenário do caso de uso consultar estado de pagamento

Caso de Uso: Verificar estado de pagamento	
Descrição:	Este caso de uso permite que administrador verifique o estado do pagamento
Actores:	Administrador
Pré-Condição:	- Comprovativo de pagamentos; - Autenticação do Administrador
Pós-Condição:	- O administrador imprime o manifesto
Fluxo Principal	
Acções do Actor	Sistema
1. Pesquisa o automóvel; 3. Valida o pagamento; 5. Imprime o manifesto.	2. O sistema exibe os dados do automóvel e o seu estado; mensagem com status do pagamento.

4.5.4 Diagrama de Classe

Este diagrama, ilustra a estrutura estática das classes que servem de modelo para os objectos do sistema e as suas relações. No diagrama são apresentadas as classes usadas no sistema e seus respectivos nomes.

A relação das classes pode ser feita através de várias formas como associação, dependência, especificação ou por pacotes.

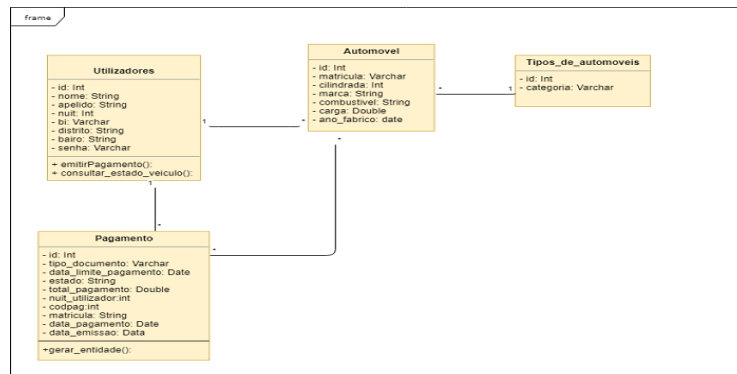


Figura 11: Diagrama de classe

4.5.5 Diagrama de Actividades

São diagramas que mostram o fluxo e a sequência das actividades onde capturam acções e seus resultados. Este normalmente é usado para demonstrar as actividades específicas do sistema.

Estes diagramas têm como objectivo demonstrar a perspectiva comportamental de um processo, uma actividade pode ser composta por uma sequência estruturada de acções.

4.5.5.1 Diagrama de Actividade do Caso de Uso Verificar estado de pagamento

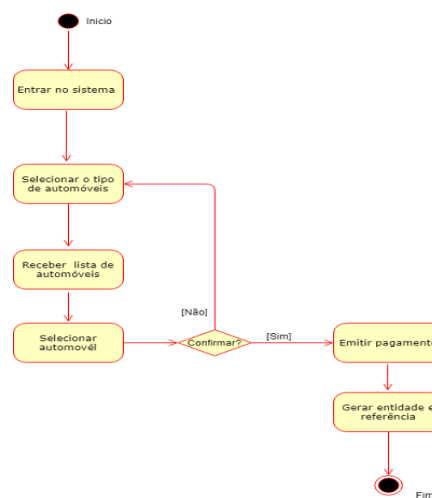


Figura 12: Diagrama de Actividade verificar estado de pagamento

4.5.5.2 Diagrama de Actividade do Caso de Uso Emitir pagamento

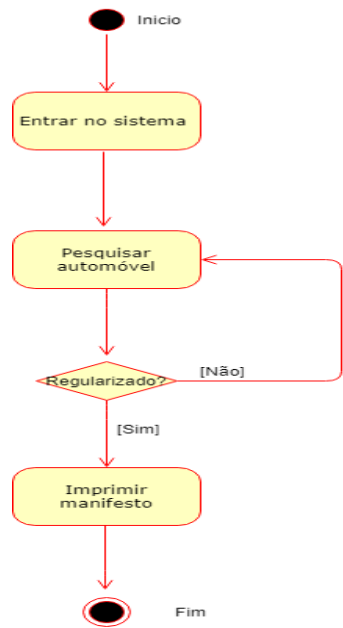


Figura 13: Diagrama de Actividade emitir pagamento

4.5.6 Diagrama de Sequência

Nesta secção vamos ilustrar como os eventos provocam transições de objeto para objeto. Uma vez que os eventos tenham sido identificados pelo exame de um caso de uso, faremos representação de como os eventos provocam o fluxo de um objeto para outro em função do tempo.

4.5.6.1 Diagrama de Sequência do Caso de Uso Emitir pagamento

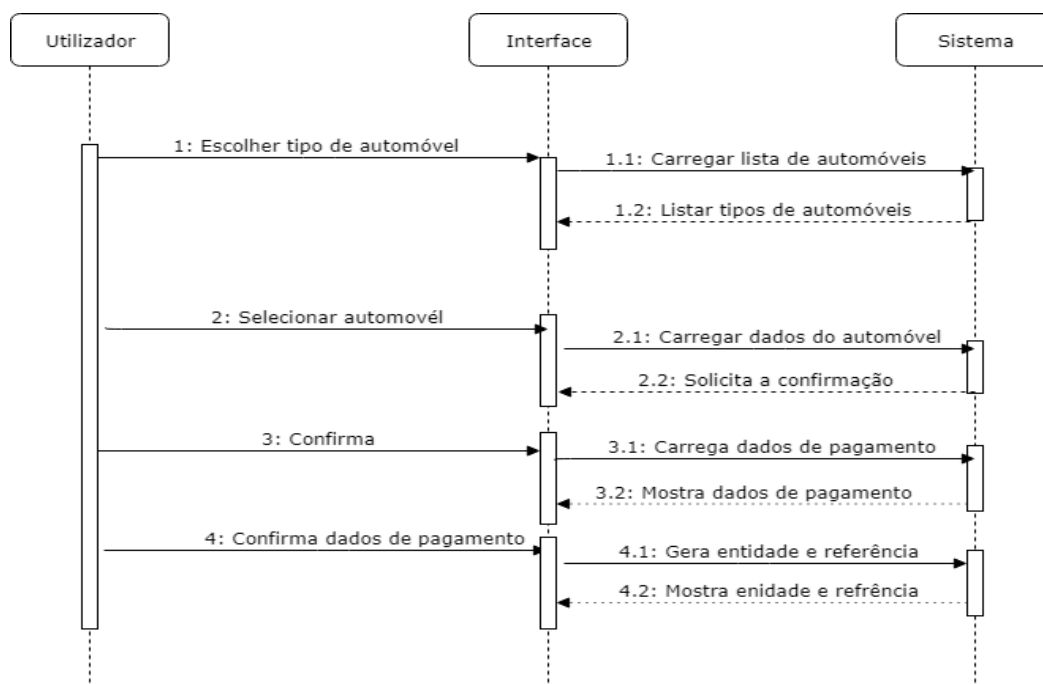


Figura 14: Diagrama de Sequência emitir pagamento

4.5.6.2 Diagrama de Sequência verificar estado de pagamento

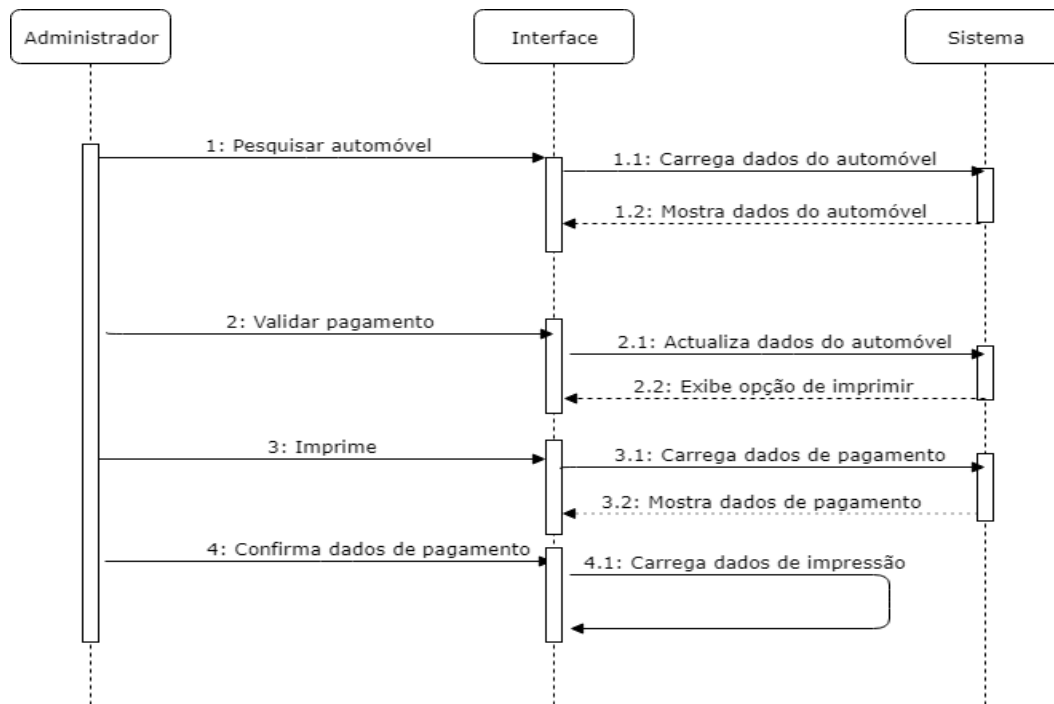


Figura 15: Diagrama de Sequência verificar estado pagamento

5 CONCLUSÕES

Neste trabalho foi apresentada uma proposta de um sistema de cobrança de impostos autárquicos de veículos de automóveis.

Em linhas gerais, concluímos que foram alcançados os objectivos da pesquisa pois desenvolveu-se um sistema de cobrança *online*, que pode servir de meio de cobrança de impostos. Assim está validada a hipótese 1 segundo a qual “com a implementação de um sistema de cobrança *online* poderá dinamizar o processo de cobrança, desta forma reduzindo grandes enchentes nos postos de cobrança”.

Neste caso, é refutada a segunda hipótese que desvaloriza a implementação de um sistema de cobrança *online* como alternativas para a cobrança de imposto. Igualmente, conseguimos descrever o sistema de cobrança *online*, que permite a emissão de pagamentos em tempo útil. Com este sistema, a qualidade de serviços fica melhorada, permitindo maior eficiência nas operações diárias, através do modelo electrónico, evitando que os cidadãos pecam a maior parte do seu tempo nos postos de cobrança, devido ao atendimento moroso por parte dos funcionários, indisponibilidade do sistema.

O actual sistema, apesar de ser informatizado, não está disponível aos cidadãos durante 24 horas ao dia, não permite mobilidade e flexibilidade no processo de cobrança, não possibilitando que os cidadãos emitam pagamentos de imposto a partir dos seus postos de trabalho, das suas casas usando qualquer dispositivo móvel.

6 RECOMENDAÇÕES

Sugere-se que nas próximas pesquisas relacionadas com o tema leve-se em consideração as seguintes recomendações:

a) Ao CMCM

- Implementar soluções inovadoras para processos de cobranças de impostos.
- Treinamento do pessoal envolvido para melhor compreensão do sistema.

b) Aos Municípios

- Que utilizem as tecnologias móveis para facilitar as actividades do seu dia a dia.

7 BIBLIOGRAFIA

- Albino, D., Souza, A. de & Prado, E. (2013), Benefícios alcançados através de um modelo de Gestão Ágil de Projeto em uma empresa de jogos eletrônicos. TCC de Administração, Usp - Universidade de São Paulo: São Paulo.
- Anselmo, F. (2002), PHP e MySQL para Windows. Florianópolis: Visual Books.
- Battisti, J. (2001), SQL Server 2000: Administração e Desenvolvimento. Rio de Janeiro:
- Bezerra, E. (2017), Princípio de Analise e Projectos com UML - 2ª Edição. Rio de Janeiro: Campus.
- Buyens, J. (2002), Aprendendo MySQL e PHP. São Paulo: MAKRON BOOKS
- Carneiro, J. (2016), Gestão e Programação de Sistemas Informáticos
- Converse, T., Morgan, C. & Park, J. (2004), PHP5 and MySQL Bible, Hoboken: Ed. Wiley.
- Correia, H. & Tafner, A. (2006), Análise Orientada a Objetos. Florianópolis: Visual Books.
- Dall'Oglio, P. (2007), PHP Programando com Orientação a Objetos: Inclui Design Patterns— 1ª Edição. São Paulo: Novatec.
- Fagundes, B. (2005), Framework para Comparação e Análise de Métodos Ágeis. Dissertação de Mestrado. Universidade Federal de Santa Catarina: Florianópolis.
- Gilmore, J. (2008), Dominando PHP e MySQL: do iniciante ao profissional. Rio de Janeiro: Alta Books.
- Goodrich, T. & Tamassia, R. (2013), Introdução à Segurança de Computadores. Porto Alegre: Bookman.
- Gudwin Ricardo R. (2015), Engenharia de Software – Uma visão prática. 2ª Edição, DCAFEEC-UNICAMP.
- GUEDES, A. (2011.), UML2 Uma Abordagem Prática, 2º edição. São Paulo:
- Larman, Craig (2003), Utilizando UML e padrões: uma introdução à análise e projeto orientados a objetos. São Paulo: Bookman, (2003:608).
- Laudon, C. & Laudon P. (2001), Gerenciamento de Sistemas de Informação - 3ª Edição. Rio de Janeiro: LTC.
- Melnikof, Reginaldo Arakaki e Edilson de Andrade Barbosa – São Paulo: Person Addison.
- Uchoa, Q. (2005), Segurança Computacional. UFLA/FAEPE: 2ª. Edição.

Mutumucio, I. V. (2008). Métodos de Investigação: Apontamentos. Centro de desenvolvimento académico. Maputo.

Richardson, Roberto Jarry. (1999) Pesquisa Social: Métodos e técnicas, 3ª Edição Editora Atlas S.A São Paulo.

Richardsom, R. (2009). Pesquisa social, métodos e técnicas. São Paulo: 3ª Edição Revista e Ampliada. Editora Atlas;

Oliveira, R. & Fernandes, N. (2006), Apache Instalação, Configuração Gestão Servidores Web FCA. Editora de Informática.

Ozsu, M. & Valduriez, P. (1999), Princípios de Sistemas de Banco de Dados Distribuídos 2ª edicao. Prentice Hall.

Pender, Tom (2014), UML, A Bíblia, Tradução Daniel Vieira - Rio de Janeiro: Campus.

Pimentel, R. (2015), Projeto de Software Usando a UML

Rocha, A. (2007), PHP – ASP – JSP: Desenvolvendo web sites dinâmicos.

Rumbaugt, J., Jacobson I. and Rumbaugh J. (2004), The Unified Modeling Language – 2º Edition, Pearson Education: Boston.

Sampaio, C. & Neto, R. (2003) Material sobre UML

Sampaio, C. (2007), História de UML

Silva, P. (2007), UML 2 em Modelagem Orientada a Objetos. Florian Polis: Visual Books.

SOMMERVILLE, Lan (2007), Engenharia de Software -8ª Edição, tradução de Selma

7.1.Webgrafia

Vivas, M., Aplicações Web utilizando PHP<<http://www.mauricio.vivas.com.br>>. Consultado em Março de 2019.

Costa, I. (2013) UML – Diagrama de Sequência. <<http://pt.slideshare.net/ItaloCosta4/umldiagrama-de-sequencia>>. Acesso em Abril de 2019.

Apache.org. The Apache Software Foundation 2019<<http://www.apache.org/>>. Consultado em Janeiro de 2019.

Santos, F. (2010) Controle de Acesso, de quem é a responsabilidade? <<https://www.trustsign.com.br/portal/blog/controle-de-acesso-dequem-e-aresponsabilidade/>>. Consultado em Maio de 2019.

Nunes, M. & O'Neill, H. (2003), "Fundamental de UML", FCA, 2ª Edição. Grandi, A., Put your data safe <<http://www.pdf.apogeeonline.com/ebook/2002/90025/pdf/PGP.pdf>>. Consultado em Abril de 2019.

Mota, F. (2003). Programação Orientada a Objetos & Delphi: Parte I – Uma Introdução sobre OO. Salvador <<http://www.geocities.com/fmotaol/oo1.htm>>. Consultado em Março 2019.

Marcel, M.; Padrão de Arquiteturas da Web <<http://marcelmesmo.blogspot.com/2011/10/padroes-de-arquitetura-da-web.html#.XO1b2ugzbiU>>. Consultado em Maio de 2019

Martinez, Marina (2015) UML<<http://www.infoescola.com/engenharia-de-software/uml/>>. Consultado em Junho de 2019.

Azevedo, F. O que é Sistema de Informação? Disponível em: <https://portal.unigranrio.edu.br/blog/o-que-e-sistemas-de-informacao> A Acesso em: nov. 2020.

Fileto, R. (2006), Sistemas Cliente-Servidor <<http://www.inf.ufsc.br/~r.fileto/Disciplinas/BDAvancado/Aulas/03-ClienteServidor.pdf>> Consultado em Abril de 2019.

8 ANEXOS

8.1 Anexo 1– Autorização para a recolha de dados


MUNICIPIO DE MAPUTO
CONSELHO MUNICIPAL
PELOURO DE DESCENTRALIZAÇÃO, BOA GOVENAÇÃO E RECURSOS HUMANOS

Exma.Senhora
Prof.Doutora. Sandra Lopes Estrela Brito
Escola Superior de Gestão, Ciências e Tecnologias
Universidade Politécnica

MAPUTO
/04/2019

867 /SG/363 /DPDRH/2019

Assunto: Pedido de Recolha de Dados

1. Comunica-se a V.Excia que por Despacho de 18 de Abril de 2019, do Exmo Senhor Presidente do Conselho Municipal de Maputo, foi autorizado o pedido de recolha de dados, Subordenada ao Tema: " Sistema de pagamento de impostos sobre Veiculos," para o senhor **Soares de Sousa Adriano Chiroute**, estudante do curso de Licenciatura em Engenharia Informática e de Telecomunicações.

2. Assim, a estudante deverá apresentar-se na Secretaria da Direcção Municipal de Recursos Humanos, para procedimentos administrativos subsequentes.

Com os melhores cumprimentos


Armando Novo 119-04-26/1815-07
/Téc. Sup. em Adm. Púb. N1/

AMF 26.04.2019

8.2 Apêndice 1– Inquérito



Universidade Politécnica A POLITÉCNICA

Escola Superior de Gestão, Ciências e Tecnologias

Trabalho de Conclusão de Curso

Inquérito sobre processo de cobrança de Imposto- Cidade de Maputo

1. Sexo:
a) Masculino ____ b) Feminino _____
2. Idade:
a) Menor que 18 ____ b) 19-28 _____ c) 29-48 ____ d) 49 ou mais _____
3. Cidade:

4. Alguma vez pagou impostos autárquicos sobre veículos de automóveis?
a) Sim ____ b) Não ____
5. Qual é a sua opinião sobre o actual sistema de cobrança de impostos?
a) Péssimo ____ b) Razoável _____ c) Bom ____ d) Muito bom _____
6. Como classifica o atendimento nos postos de cobrança de impostos?
a) Péssimo ____ b) Razoável _____ c) Bom ____ d) Muito bom _____
7. Quais são as dificuldades que tem enfrentando nos postos de cobrança de impostos?
a) Longas filas de espera _____
b) Incompatibilidade de impressoras _____
c) Erros de natureza humana _____
d) Outras:

8. Que medidas sugere para a melhoria do sistema actual?

9. Acredita que a existência de um sistema online de cobrança de impostos, seria uma mais valia?
a) Sim ____ b) Não _____ c) Talvez _____

8.3 Apêndice 2 – Interfaces Gráficas do Sistema

A primeira janela após a execução do sistema é a do login, onde são solicitadas as credenciais para que o sistema possa abrir as janelas subsequentes, os dados de acesso são previamente registrados e a atribuição de privilégios é feita na instituição. A janela login conta os campos NUIT e Palavra-passe como podemos observar na figura abaixo, caso o preenchimento destes campos seja feito de forma correcta o usuário passa para a próxima tela. No caso de se fazer inserção de dados não correctos, será exibida uma mensagem de erro e solicitando que usuário entre em contacto com a instituição.



Autenticação

NUIT:

Palavra-passe:

Entrar

Uma vez autenticado no sistema, o utilizador é encaminhado para uma nova tela onde são listados os automóveis em seu nome. O utilizador selecciona o automóvel que pretende efectuar o pagamento de IAV. Posteriormente carrega em "Emitir".

Tipos de Automóveis

- Automóveis ligeiros
 - Toyota Raum AFZ729MC
 - Honda Fit AFU400MC
- Automóveis pesados de carga
- Automóveis pesados de passageiros

Emitir

Após clicar em emitir será exibida uma próxima janela com dados de pagamento onde poderá conferir os dados, feito isso o utilizador carrega em emitir pagamento conforme ilustra a figura abaixo.

Dados de Pagamento

Matricula	Entidade	Data-emissão	Tipo	Total	Data-limite	Multa
AFZ729MC	José Titos	29/05/2019	Ligeiro	150.00	29/07/2019	0.00

Emitir pagamento

Feita a emissão de pagamento é exibido o talão de pagamento, com este talão de pagamento o utilizador pode se deslocar até ATM mais próxima e efectuar o pagamento ou mesmo recorrendo as plataformas de pagamento online disponibilizadas pelos bancos comercias.

TALÃO DE PAGAMENTO

Dados pessoais	Informação para pagamento
Nuit: 987654321	Entidade:90001
Entidade: José Titos	Referência: 07742280563
	Valor: 150,00
	Vencimento: 29/07/2019