**UNIVERSIDADE TÉCNICA DE ANGOLA**

**FACULDADE DE ENGENHARIAS**

**DEPARTAMENTO DE ENSINO E INVESTIGAÇÃO DAS TECNOLOGIAS DE INFORMAÇÃ E COMUNICAÇÃO-DEITIC**

**CURSO DE ENGENHARIA INFORMÁTICA**

**(DESENVOLVIMENTO DE UM SISTEMA DE GESTÃO DE MUNÍCIPES PARA ADMINISTRAÇÃO MUNICIPAL DE MALANJE)**

**ALEKSEI JUSTINO MATEUS DONDO DA SILVA**

**LUANDA-ANGOLA**

**MES/2020**

**UNIVERSIDADE TÉCNICA DE ANGOLA**

**FACULDADE DE ENGENHARIAS**

**DEPARTAMENTO DE ENSINO E INVESTIGAÇÃO DAS TECNOLOGIAS DE INFORMAÇÃ E COMUNICAÇÃO-DEITIC**

**CURSO DE ENGENHARIA INFORMÁTICA**

**(DESENVOLVIMENTO DE UM SISTEMA DE GESTÃO DE MUNÍCIPES PARA ADMINISTRAÇÃO MUNICIPAL DE MALANJE)**

**ALEKSEI JUSTINO MATEUS DONDO DA SILVA**

Trabalho de fim de curso apresentado a Universidade Técnica de Angola, como requisito para obtenção do título de licenciado em Engenharia Informática.

Orientador: Eng. Martinho Joaquim Kassequele. Licenciado

**LUANDA-ANGOLA**

**MES/2020**

**ALEKSEI JUSTINO MATEUS DONDO DA SILVA**

**(DESENVOLVIMENTO DE UM SISTEMA DE GESTÃO DE MUNÍCIPES PARA ADMINISTRAÇÃO MUNICIPAL DE MALANJE)**

**Aprovado**\_\_\_\_/\_\_\_\_\_/\_\_\_\_\_

BANCA EXAMINADORA

Presidente do Júri.................................................................Assinatura..............................

Primeiro Vogal....................................................................Assinatura...............................

Segundo Vogal....................................................................Assinatura...............................

#### DEDICATÓRIA

Dedico este trabalho a **minha esposa** e **filhas**, aos **meus pais** e a todos que sempre estiveram ao meu lado desde o início e que me auxiliaram sempre para que, mesmo com todas as dificuldades, eu nunca desviasse do meu caminho.

#### AGRADECIMENTOS

Em primeiro lugar a **Deus**, que sempre me deu força e coragem nos momentos mais difíceis.

Á toda **minha família** e **amigos**, por sempre me estimularem a crescer cada vez mais e me auxiliarem naquilo que precisei no momento.

Ao orientador **Lic. Martinho Kassequele**, e co-orientador **Msc. Amiraldes Xavier**, que muito me inspiraram e apoiaram nessa árdua trajectória.

Aos funcionários da Administração Municipal de Malanje, na pessoa da **Directora Mara** que sempre disponibilizaram seu precioso tempo para responder e esclarecer certas questões.

Á DPPF/Minint/Malanje, na pessoa do **Director Fernando Zage** e colegas que sempre souberam controlar a minha ausência no local de serviço.

Aos meus preciosos colegas de sala, **Manuel Lopes**, **Zola Panzo**, **Edgar André**, **Ariclenes Quíssua**, **Teófilo Félix** e **Nelson Sofonias**, que sempre me apoiaram moral e psicologicamente a não desistir dessa luta.

#### EPÍGRAFE

*Talvez não tenha conseguido fazer o melhor, mas lutei para que o melhor fosse feito. Não sou o que deveria ser, mas Graças a Deus, não sou o que era antes (Marthin Luther King).*

#### RESUMO

O controle de munícipes de um determinado município tem sido uma tarefa árdua e custosa por parte dos órgãos do aparelho do estado. As Administrações têm se deparado com este grande empecilho quando se pretende emitir documentos e controlar a estatística dos munícipes por localidades ou bairros. Esta é uma tarefa que as administrações municipais precisam realizar devidos as actividades que os munícipes realizam ao longo de sua estadia enquanto residente num município.

Dada a relevância, o gerenciamento de munícipes é indispensável pois providencia maior controlo da série de documentos que se emite para cada munícipe.

Nesta senda, o trabalho aqui apresentado, descreve uma alternativa eficiente e eficaz que permite as Administrações Municipais a terem maior rigor e controlo dos munícipes que habitam naquele município e o histórico de documentação dos mesmos. Esta alternativa traduz-se em desenvolver um sistema de gerenciamento de munícipes (SiGMun), aplicação de survey, para a recolha de dados, utilizando as técnicas de engenharia de software para realização da análise dos requisitos, utilização da linguagem uml para elaboração dos diagramas e consequentemente efectuar os testes da aplicação. De modos a desenvolver um software de qualidade, utilizamos os princípios que regem as boas práticas de desenvolvimento de software, tais como: aplicação do padrão arquitetural ASP.net MVC, aplicação do bootstrap para termos um website responsivo, ajustável a qualquer tela.

**Palavras Chaves:** Gerenciamento de Munícipes; Administração Municipal de Malanje; Controlo Estatístico, Emissão de Documentos;

#### ABSTRACT

The control of residents of a given municipality has been an arduous and costly task on the part of the organs of the state apparatus. Administrations have faced this great obstacle when it comes to issuing documents and controlling the statistics of residents by localities or neighborhoods. This is a task that municipal administrations need to carry out due to the activities that residents carry out throughout their stay as a resident in a municipality.

Given the relevance, the management of citizens is essential as it provides greater control over the series of documents that are issued for each citizen.

In this way, the work presented here, describes an efficient and effective alternative that allows the Municipal Administrations to have greater rigor and control of the citizens who live in that municipality and their documentation history. This alternative translates into the development of a citizen management system (SiGMun), a survey application, for data collection, using software engineering techniques to perform the requirements analysis, use the uml language to prepare the diagrams and consequently carry out the application tests. In order to develop quality software, we use the principles that govern good software development practices, such as: application of the ASP.net MVC architectural standard, application of boostrap to have a responsive website, adjustable to any screen.

**Key words:** Citizens Management; Municipal Administration of Malanje; Statistical Control, Issuance of Documents;

#### LISTA DE SIGLAS E ABREVIATURAS

|  |  |
| --- | --- |
| AES------------- | - |
| AJAX----------- | Asynchronous Javascript and Xml |
| API-------------- | Application Programming Interface |
| ASP-------------- | Active Server page |
| BACK-END---- | Parte de Retaguarda, lado do Servidor |
| CASE----------- | Computer -Aided Software Engineering |
| CIM------------- | Cartão de Identificação do Munícipe |
| CLR------------- | Common Language Runtime |
| CRUD----------- | -Create Read Update Delete |
| CSS-------------- | Folha de Estilo em Cascata |
| DADI------------ | Definition Architecture Design Implementation |
| DB--------------- | Data Base |
| DCL------------- | Data Control Language |
| DDL------------- | Data Definition Language |
| DLL------------- | Dynamic Link Library |
| DML------------ | Data Manipulation Language |
| DQL------------- | Data Query Language |
| DTL------------- | Data Transaction Language |
| EAI-------------- | Enterprise Application Integration |
| FDD------------- | Feature Driven Development |
| FK--------------- | Foreign Key |
| FRONT-END-- | Interface frontal, lado do cliente |
| GUI-------------- | -Graphical User Interface |
| HTML----------- | -HyperText Markup Language |
| IDE-------------- | -Integrated Development Environment |
| IEEE------------- | -Instituto de Engenheiro Electrónicos e Electricistas |
| IIS--------------- | -Internet Information Service |
| ISO | International Standardizattion Organization |
| JSON------------ | JavaScript Object Notation |
| LDAP----------- | -Lightweight Directory Access Protocol |
| LDF------------- | -Log Data File |
| LINQ------------ | -Language Integrated Query |
| MDF------------- | -Master Data File |
| MVC------------ | -Model View Controller |
| OO--------------- | -Orientação a Objectos |
| OOP------------- | -Programação Orientada a Objectos |
| PK--------------- | -Primary Key |
| REST----------- | -Representation State Transfer |
| RF--------------- | -Requisitos Funcionais |
| RN-------------- | -Regras de Negócio |
| RNF------------- | -Requisitos Não Funcionais |
| SDLC----------- | -Software Development Life Cycle |
| SGBD----------- | -Sistema de Gerenciamento de Banco de Dados |
| SGDBR--------- | -Sistema de Gerenciamento de Banco de Dados Relacional |
| SI---------------- | -Sistema de Informação |
| SiGMun-------- | -Sistema de Gerenciamento de Munícipes |
| SOAP---------- | -Single Object Application Protocol |
| SPA-------------- | -Single Page Application |
| SQL------------- | -Structured Language Query |
| UC--------------- | -Use Case |
| UML------------ | -Unified Modeling Language |
| URL------------- | -Uniform Resource Locator |
| WWW----------- | -World Wide Web |
| XML------------ | -Extensible Markup Language |
| XP--------------- | -Extreme Programming |

#### LISTA DE FIGURAS

[Figura 1:Modelo de cartão de Munícipe (CIM) 20](#_Toc52449484)

[Figura 2:Metodologia de Pesquisa Adotada. 21](#_Toc52449485)

[Figura 3:Diagrama de Casos de Uso. 28](#_Toc52449486)

[Figura 4:Diagrama Entidade-Relacionamento. 35](file:///C:\Users\SoftDev\Documents\GitHub\SiGMunPro\Monografia\Word\TFC%20Final%20-Antes%20do%20Tutor.docx#_Toc52449487)

[Figura 5:Diagrama Lógico. 36](#_Toc52449488)

[Figura 6: Diagrama de Classes 38](#_Toc52449489)

[Figura 7:Diagrama de Sequência Cadastrar Usuários 39](#_Toc52449490)

[Figura 8:Diagrama de Sequência Cadastrar Munícipe 39](#_Toc52449491)

[Figura 9:Diagrama de Sequência Cadastrar Informações de Familiares 40](#_Toc52449492)

[Figura 10:Diagrama de Sequência Cadastrar Morada 41](#_Toc52449493)

[Figura 11:Diagrama de Sequência Cadastrar Grau Académico 41](#_Toc52449494)

[Figura 12:Diagrama de Sequência Emitir Documento 42](#_Toc52449495)

[Figura 13:Diagrama de Sequência Emitir Cartão CIM 43](#_Toc52449496)

[Figura 14:Diagrama de Instalação 44](#_Toc52449497)

[Figura 15:Arquitetura Lógica do Sistema 46](#_Toc52449498)

[Figura 16: Linguagens do SQL 49](#_Toc52449499)

[Figura 17:Arquitetura do Entity Framework 51](#_Toc52449500)

[Figura 18:Modelo Arquitetural MVC. 53](#_Toc52449501)

[Figura 19:Código Qr gerado pela Sistema 55](#_Toc52449502)

[Figura 20:Código de Barras Gerado pelo sistema 56](#_Toc52449503)

[Figura 21:Diagrama de Distribuição de Pacotes Nuget 57](#_Toc52449504)

[Figura 22:Modelo de Objectos do ADO.Net 58](#_Toc52449505)

[Figura 23: Modelo de Desenvolvimento iterativo e incremental 61](#_Toc52449506)

[Figura 24:Ciclo de Vida de Desenvolvimento de Software 63](#_Toc52449507)

#### LISTA DE TABELAS

[Tabela 1.3.2: Requisitos Funcionais 23](#_Toc52556247)

[Tabela 2.3.3: Requisitos Não Funcionais 25](#_Toc52556248)

[Tabela 3.3.4: Regras de Negócios 26](#_Toc52556249)

[**Tabela 4.4.3.1:** Detalhamento do Caso de Uso Cadastrar Usuário 29](#_Toc52556250)

[**Tabela 5.4.3.2:** Detalhamento do Caso de Uso Cadastrar Munícipe 30](#_Toc52556251)

[**Tabela 6.4.3.3:** Detalhamento do Caso de Uso Cadastrar Familiar do Munícipe 31](#_Toc52556252)

[**Tabela 7.4.3.4:** Detalhamento do Caso de Uso Cadastrar Morada 32](#_Toc52556253)

[**Tabela 8.4.3.5:** Detalhamento do Caso de Uso Cadastrar Grau Académico 33](#_Toc52556254)

[**Tabela 9.4.3.6:** Detalhamento do Caso de Uso Emitir Documento 34](#_Toc52556255)

[**Tabela 10.4.3.7:** Detalhamento do Caso de Uso Emitir Cartão CIM 34](#_Toc52556256)

[Tabela 11:Usuários 69](#_Toc52556257)

[Tabela 12:Afirmações de usuários 69](#_Toc52556258)

[Tabela 13:Login de Usuário 70](#_Toc52556259)

[Tabela 14:Tipos de Documentos 70](#_Toc52556260)

[Tabela 15:Provincias 70](#_Toc52556261)

[Tabela 16:Perfis de usuário 71](#_Toc52556262)

[Tabela 17:Parente de Munícipe 71](#_Toc52556263)

[Tabela 18:Municipio 71](#_Toc52556264)

[Tabela 19:Municipe 72](#_Toc52556265)

[Tabela 20:Morada 72](#_Toc52556266)

[Tabela 21:Grau Parentesco 73](#_Toc52556267)

[Tabela 22:Géneros 73](#_Toc52556268)

[Tabela 23:Formas de Pagamento 73](#_Toc52556269)

[Tabela 24:Estado Civil 73](#_Toc52556270)

[Tabela 25:Finalidade dos Documentos 74](#_Toc52556271)

[Tabela 26:Comunas 74](#_Toc52556272)

[Tabela 27:Bancos 74](#_Toc52556273)

[Tabela 28:Bairros 75](#_Toc52556274)

[Tabela 29:Permissões 75](#_Toc52556275)

[Tabela 30:Operações 75](#_Toc52556276)

#### ÍNDICE

[INTRODUÇÃO 1](#_Toc53531462)

[DEFINIÇÃO DO PROBLEMA 2](#_Toc53531463)

[JUSTIFICATIVA 2](#_Toc53531464)

[OBJECTIVOS 3](#_Toc53531465)

[OBJECTIVOS GERAIS 3](#_Toc53531466)

[OBJECTIVOS ESPECÍFICOS 3](#_Toc53531467)

[DELIMITAÇÃO 3](#_Toc53531468)

[ESTRUTURA DO TRABALHO 4](#_Toc53531469)

[CAPÍTULO: 1- FUNDAMENTAÇÃO TEÓRICA 5](#_Toc53531470)

[1.1. Administração Municipal 5](#_Toc53531471)

[1.1.1. Estrutura Orgânica 5](#_Toc53531472)

[1.2. Secretaria da Administração Municipal 5](#_Toc53531473)

[1.3. Classificação Documental 6](#_Toc53531474)

[1.3.1. Atestado de Residência 6](#_Toc53531475)

[1.3.2. Atestado de Pobreza 6](#_Toc53531476)

[1.3.3. Agregado familiar 6](#_Toc53531477)

[1.3.4. Licença de condução 7](#_Toc53531478)

[1.3.5. Registo de velocípedes 7](#_Toc53531479)

[1.3.6. Cartão CIM 7](#_Toc53531480)

[CAPÍTULO: 2- METODOLOGIA 9](#_Toc53531481)

[2.1. Tipo de Pesquisa 9](#_Toc53531482)

[2.2. Campo de Estudo 9](#_Toc53531483)

[2.3. Levantamento de Requisitos 10](#_Toc53531484)

[2.3.1. Análise de Requisitos 10](#_Toc53531485)

[2.3.2. Requisitos Funcionais 10](#_Toc53531486)

[2.3.3. Requisitos Não Funcionais 11](#_Toc53531487)

[2.3.4. Regras de Negócio 12](#_Toc53531488)

[2.4. Análise do Sistema 12](#_Toc53531489)

[Diagramas do Sistema 12](#_Toc53531490)

[Diagrama de caso de Uso 13](#_Toc53531491)

[2.4.1. Descrição dos Principais Casos de Uso 13](#_Toc53531492)

[2.4.2. Diagrama Entidade-Relacionamento 18](#_Toc53531493)

[2.5. Diagrama Lógico 18](#_Toc53531494)

[2.5.1. Diagrama de Classe 19](#_Toc53531495)

[2.5.2. Diagrama de Sequência 20](#_Toc53531496)

[2.5.3. Diagrama de actividade 23](#_Toc53531497)

[2.5.4. Diagrama de Instalação 26](#_Toc53531498)

[2.6. Ferramentas de Desenvolvimento 27](#_Toc53531499)

[2.7. Arquitectura lógica do Sistema 28](#_Toc53531500)

[2.8. Desenvolvimento de base de Dados 29](#_Toc53531501)

[2.8.1. Criação de base de dados 29](#_Toc53531502)

[2.9. Tecnologias envolvidas no Projecto 30](#_Toc53531503)

[2.9.1. Base de Dados 30](#_Toc53531504)

[2.9.2. Sistemas de gestão de base de dados 30](#_Toc53531505)

[2.9.3. SQL 31](#_Toc53531506)

[2.9.4. Procedimentos armazenados 32](#_Toc53531507)

[2.9.5. Entity Framework 6.0 33](#_Toc53531508)

[2.10. Programação 34](#_Toc53531509)

[2.10.1. Asp.NET 4.8 34](#_Toc53531510)

[2.10.2. Asp.net MVC 34](#_Toc53531511)

[2.10.3. Booststrap 35](#_Toc53531512)

[2.10.4. Jquery 35](#_Toc53531513)

[2.10.5. Validações 35](#_Toc53531514)

[2.10.6. Scaffolding 36](#_Toc53531515)

[2.10.7. Qr Code 37](#_Toc53531516)

[2.10.8. Bar Code 37](#_Toc53531517)

[2.10.9. Nuget 38](#_Toc53531518)

[2.10.10. ASP.NET Identity 2.1 39](#_Toc53531519)

[2.10.11. Ado.NET 39](#_Toc53531520)

[2.10.12. Json 40](#_Toc53531521)

[2.11. Fase de Testes 40](#_Toc53531522)

[2.11.1. Teste de Unidade 41](#_Toc53531523)

[2.11.2. Teste de Sistema 41](#_Toc53531524)

[2.11.3. Teste de Aceitação 41](#_Toc53531525)

[2.12. Metodologias de desenvolvimento de software 42](#_Toc53531526)

[2.12.1. Refatoração 43](#_Toc53531527)

[2.12.2. Ciclo de vida do desenvolvimento do sistema 44](#_Toc53531528)

[CAPÍTULO: 3- RESULTADOS 45](#_Toc53531529)

[3.1. Descrição do Projecto 45](#_Toc53531532)

[3.2. Interfaces do Sistema 45](#_Toc53531533)

[3.2.1. Tela do Administrador 45](#_Toc53531534)

[3.2.2. Tela de Cadastro de Usuário 46](#_Toc53531535)

[3.2.3. Tela de cadastro do Munícipe 46](#_Toc53531536)

[3.2.4. Tela de Emissão de Documento 46](#_Toc53531537)

[3.2.5. Tela de emissão de cartão CIM 46](#_Toc53531538)

[CONCLUSÃO E RECOMENDAÇÕES 47](#_Toc53531539)

[CONCLUSÃO 47](#_Toc53531540)

[RECOMENDAÇÕES 47](#_Toc53531541)

[REFERÊNCIAS BIBLIOGRAFICAS 48](#_Toc53531542)

[LIVROS 48](#_Toc53531543)

[SITES 48](#_Toc53531544)

[APÊNDICE I- Dicionário de Dados 1](#_Toc53531545)

[APÊNDICE II – ROTEIRO DE ENTREVISTA 8](#_Toc53531546)

[APÊNDICE III- EXEMPLO DE ENTRVISTA 8](#_Toc53531547)

[ANEXOS 8](#_Toc53531548)

[GLOSSÁRIO 9](#_Toc53531549)

# INTRODUÇÃO

As tecnologias de informação e comunicação surgiram para facilitar a vida do homem. Desde a criação do Eniac, isto em 1946, o esforço físico e mental do homem passou a ser dividido por máquinas. Com a evolução dos computadores, os softwares também evoluíram também afim de dinamizar e automatizar os processos que eram realizados unicamente com o esforço do homem.

Os softwares são programas de computador e sua documentação associada, esses produtos podem ser desenvolvidos para um cliente específico ou um mercado em geral (SOMMERVILLE, 2011).

Um software é uma sequência de instruções escritas para serem interpretadas por um computador através de um compilador, com objectivo de exercer tarefas especificas a qual foi desenvolvido e assim poupar o esforço da mente humana.

Neste âmbito, queremos aqui apresentar o desenvolvimento de um software para Administração Municipal de Malanje, cuja finalidade é auxiliar este órgão da Administração Local do Estado no que diz respeito ao munícipe, tendo em vista que tal órgão é responsável por controlar as moradias e emitir documentos que os munícipes precisam para interagir com outras entidades do estado.

Assim, pensando unicamente no atendimento eficaz e eficiente do cidadão malanjino e como uma das melhores hipóteses para resolução do problema de enchentes no atendimento, demora na obtenção dos documentos, inexistência de mecanismo de controlo e histórico documental, apresentamos o SiGMun (Sistema de Gerenciamento de Munícipes).

## DEFINIÇÃO DO PROBLEMA

Actualmente a Administração Municipal de Malanje utiliza a forma de emissão ou criação de documentos tradicional, isto é, os funcionários utilizam a ferramenta Microsoft Word para a criação dos documentos que o munícipe solicita e posteriormente armazenam em ficheiros. Desta forma, cada solicitação, que é feita, o funcionário rescreve o mesmo ficheiro, alterando apenas os dados para combinar com novas informações passadas pelo munícipe através da cópia do bilhete de identidade e as anotações feitas pelo munícipe.

O sistema tradicional, tem criado vários embaraços, principalmente quando o sistema operacional da máquina danifica, chegando em certas situações a perderem-se os ficheiros armazenados na mesma máquina. Por outro lado, por causa do uso constante de dispositivos de armazenamento portátil (Pendrive), que muitas das vezes transportam aos computadores que se conectam, vírus e outras pragas informáticas que podem deteriorar os ficheiros existentes na máquina.

Outrossim, com o sistema tradicional certas questões como: obter histórico de solicitação dos munícipes, morada dos munícipes mais ou menos precisa, era quase impossível de controlar. Mas sim, controla-se apenas quantos documentos foram emitidos, mas sem identificar de forma precisa os titulares, isto através da numeração sequencial que se coloca em cada documento que é emitido.

Perante tal senário, levanta-se a seguinte questão para a pesquisa deste trabalho:

Como solucionar por meios tecnológicos os problemas encontrados na Administração Municipal de Malanje, no que se refere ao atendimento e controlo dos munícipes?

## JUSTIFICATIVA

Partindo do princípio que os softwares existem para tornar a árdua actividade do homem mais compreensível, este trabalho justificasse pela necessidade de não haver no campo de estudo um software específico para a gestão de munícipes que proporcione as funcionalidades que façam com que a gestão dos munícipes seja mais eficiente, prática e eficaz.

Portanto, o desenvolvimento deste software trará melhorias no quesito atendimento aos munícipes, no controlo de suas actividades a nível documental, isto é, suas requisições, seu agregado familiar, sua moradia e possivelmente seus títulos académicos. Desta forma, proporcionará também a Administração Municipal de Malanje, maior rapidez e facilidade no atendimento dos pedidos, visto que para emissão de algum documento o funcionário precisará apenas passar o cartão CIM ao leitor de código de Barras ou QR para obter os dados do munícipe e assim prosseguir com a solicitação. Para a Administração, isto trará maior produtividade na obtenção de resultados e em contrapartida, maior satisfação do munícipe.

Por outro lado, desenvolver este software me proporcionará um alto nível de conhecimento das ferramentas que serão utilizadas e melhoria das aptidões como programador.

## OBJECTIVOS

## OBJECTIVOS GERAIS

Desenvolver um sistema web para gerenciamento de munícipes, garantindo a emissão de seus documentos bem como o controlo estatístico, histórico a nível de solicitações, emissão de relatórios e controlo das suas respectivas moradias.

## OBJECTIVOS ESPECÍFICOS

* Efectuar o levantamento e analise de requisitos.
* Modelar o sistema usando a linguagem UML para criar o diagramas.
* Desenvolver uma base de dados para a persistência dos dados de acordo as necessidades, utilizando a ferramentas SQL Server.
* Codificar a aplicação utilizando a linguagem C#.
* Testar e mostrar as funcionalidades do sistema.

## DELIMITAÇÃO

Este estudo delimitasse na Administração Municipal Malanje, mais concretamente no Departamento Administrativo, na Área do Guiché, que é a área que interage directamente com os munícipes para quaisquer situações que queiram ver esclarecida.

## ESTRUTURA DO TRABALHO

Para melhor enquadramento e situação no relatório, este trabalho foi dividido em capítulos conforme segue a estrutura abaixo:

Capítulo 1- Fundamentação Teórica –este capítulo é reservado para abordagens mais teórica do projecto, apresentamos aqui as formas e procedimentos existentes actualmente dão solução a problemática de atendimento ao munícipe.

Capítulo 2- Metodologia – neste capítulo, apresentamos as técnicas, os procedimentos e os métodos de pesquisa usados para elaboração deste projecto, desde o levantamento de requisitos, analise e apresentação dos diagramas. Fizemos também a descrição das tecnologias envolvidas e as razões para aplica-las.

Capítulo 3- Resultados – neste capítulo apresentamos o projecto de forma detalhada e funcional, como foi construído a aplicação para auxilia a gestão dos munícipes, apresentação das principais telas do sistema e outros aspectos importantes.

Capítulo 4- Conclusão e Recomendação

# CAPÍTULO: 1- FUNDAMENTAÇÃO TEÓRICA

* 1. Administração Municipal

Administração Municipal é o órgão desconcentrado da administração do Estado na província que visa assegurar a realização das acções do estado a nível do município, na execução das suas competências. Cabe a Administração Municipal, promover e orientar o desenvolvimento económico e social e assegurar a prestação de serviços públicos da respectiva área geográfica (MAT, 2007).

* + 1. Estrutura Orgânica

A estrutura orgânica da Administração Municipal compreende os seguintes órgãos e serviços:

1. No Domínio do planeamento e orçamento
2. Serviços de Apoio técnico:
3. Secretaria da Administração Municipal.
4. Repartição de Estudos e Planeamento.
5. Serviços de Apoio Instrumental
6. Serviços Desconcentrado da Administração Municipal

Nesta estrutura orgânica, focamo-nos nos Serviços de Apoio técnico mais precisamente na Secretaria da Administração Municipal, que é a divisão desta entidade que se relaciona directamente com o atendimento ao cidadão.

* 1. Secretaria da Administração Municipal

A secretaria da Administração Municipal é o serviço que se ocupa da generalidade das questões administrativas, gestão do pessoal, património, orçamento e relações públicas e transportes.

Deste modo, para Administração Municipal de Malanje, o munícipe é todo aquele ser que procura ou solicita os serviços da administração. O mesmo é considerado munícipe quando acarreta consigo um documento de identificação que neste caso é o bilhete de identidade.

A Administração Municipal de Malanje, disponibiliza uma gama de documentos para que o munícipe possa interagir com outros órgãos do estado, dentre estes documentos podemos destacar: Atestado de Residência; Atestado de Pobreza, Auto de Ocorrência, Agregado Familiar, Licença de Condução de Velocípedes com Motor, Registo de Velocípedes entre outros.

Sendo estes documentos acima citados com maior índice de solicitações, na qual nos leva a desenvolver este sistema que inicialmente dará solução a estes problemas.

* 1. Classificação Documental
     1. Atestado de Residência

O atestado de residência é um documento emitido para o munícipe pela administração municipal que atesta o munícipe reside num determinado município, comuna e bairro, num prazo de dois anos. Este documento, pode ser emitido para diversos fins como comprovação de residência junto das instituições públicas e privadas.

* + 1. Atestado de Pobreza

O atestado de pobreza é um documento emitido pela administração municipal para comprovar que um munícipe não tem condições de pagar os custos exigidos com:

* Assessoria jurídica;
* Isenção das taxas cobradas em cartórios;
* Certidão de casamento;
* Certidão de divorcio;
* Isenção de taxas de inscrição e manuais cobrados em concursos públicos;
* Renegociação de dividas e
* Redução dos custos de propina e matrícula escolar, etc.
  + 1. Agregado familiar

O agregado familiar é um documento que confirma ou atesta o grau de parentesco de pessoas que residem sobre responsabilidades económico-financeira de um cidadão.

Para obtenção desta documentação, o munícipe terá que dirigir-se com a sua documentação de identificação e a documentação de seu agregado a Administração Municipal e seguir os tramites.

* + 1. Licença de condução

A licença de condução é documento que habilita o cidadão a conduzir velocípedes com o sem motor não superior a 100C.C (centímetros cúbicos) e com idade superior a 17 anos.

* + 1. Registo de velocípedes

O registo de velocípedes ou cartão de registo de velocípedes é um documento emitido pela administração municipal que autoriza e legaliza o velocípede com motor e cilindragem não superior a 100 C.C a transitar na via pública. Para sua obtenção, o cidadão, deverá apresentar seu documento de identificação, comprovativo de aquisição do velocípede e o termo de compra e venda devidamente reconhecido pela conservatória dos registos do cartório.

* + 1. Cartão CIM

O cartão CIM (Cartão de Identificação de Munícipe), é uma documentação nova, apresentada como proposta enquanto decorria a entrevista com os funcionários da administração municipal de Malanje.

Este cartão visa identificar o munícipe dentro e fora do seu município e não só, também facilitar a emissão dos documentos que o mesmo solicita a Administração Municipal. O cartão é composto por: identificação da Entidade emissora do cartão, o número de munícipe (NM), os dados de identificação do munícipe como, nome completo, e morada, o código de barras e o código QR conforme ilustra a figura abaixo:



Figura 1:Modelo de cartão de Munícipe (CIM)

# CAPÍTULO: 2- METODOLOGIA

## Tipo de Pesquisa

A metodologia adoptada para este trabalho foi dividida em 4 etapas, conforme apresentado na figura 2.1. A primeira parte foi a escolha do campo de estudo para que seja possível a realização da pesquisa. A segunda deu-se por determinar o projecto e pesquisa do survey. A terceira foi a realização da aplicação do survey na empresa seleccionada e por último, a realização da consolidação dos resultados.

A pesquisa em forma survey é um tipo quantitativo que é utilizado quando deseja-se obter dados referentes a determinado grupo sobre algum assunto. A utilização da mesma dá-se principalmente por questionários e de acordo com o formato do survey pode ter como pressuposto ser: explanatórios, em que são testadas as hipóteses; exploratório, em que o foco é um maior conhecimento sobre determinado tema; e descritivo, onde estabelecer características e padrões são o ponto principal.

O tipo survey abrangido neste trabalho é descritivo, no qual foi realizado uma entrevista com os funcionários da Administração Municipal que teve a duração de 2 dias

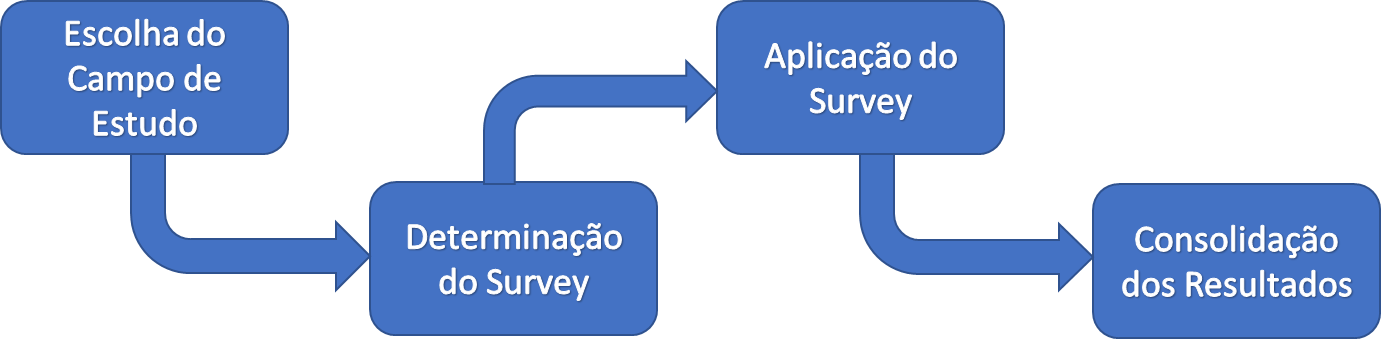


Figura 2:Metodologia de Pesquisa Adotada.

## Campo de Estudo

O campo de estudo desta pesquisa é a Administração Municipal de Malanje e está localizada no bairro azul, centro da cidade.

## Levantamento de Requisitos

Os requisitos de um sistema são as descrições do que o sistema deve fazer, os serviços que oferece e as restrições a seu funcionamento. Esses requisitos reflectem as necessidades dos clientes para um sistema que serve a uma finalidade determinada (SOMMERVILLE, 2011).

Nesta fase procedemos a recolha de informações relativas ao projecto, de modo que foram obtidas por meio de entrevistas e aplicação de questionários estruturado aos funcionários da Administração Municipal de Malanje.

* + 1. Análise de Requisitos

Segundo a IEEE a análise de requisitos é o processo que envolve o estudo das necessidades do usuário para se encontrar uma definição correcta ou completa do sistema ou dos requisitos do software. Essa análise de requisitos é vital para o desenvolvimento do sistema, pois ela vai determinar o sucesso ou fracasso do projecto (QUITERIO,2012).

Desta forma, os requisitos de software são frequentemente classificados como: Requisitos funcionais e requisitos não funcionais.

* + 1. Requisitos Funcionais

Os requisitos funcionais são as reais necessidades do cliente. São as funcionalidades que satisfazem o desejo dos futuros utilizadores do sistema.

Conforme Marinela (2013), “Descrevem funcionalidade ou serviços que o sistema deve fornecer, como irá reagir a entradas particulares e como irá se comportar em determinadas situações.”

|  |  |
| --- | --- |
| Código | Descrição |
| RF1 | O sistema deve permitir gerir os usuários e seus perfis. |
| RF2 | O sistema deve permitir gerir as finalidades dos documentos. |
| RF3 | O sistema deve permitir gerir os munícipes. |
| RF4 | O sistema deve permitir gerir moradas. |
| RF5 | O sistema deve permitir emitir documentos |
| RF6 | O sistema deve permitir efectuar pagamento. |
| RF7 | O sistema deve permitir emitir relatórios. |
| RF8 | O sistema deve permitir efectuar consultas. |

Tabela 1.3.2: Requisitos Funcionais

* + 1. Requisitos Não Funcionais

Os Requisitos não funcionais são as restrições impostas para manter o melhor funcionamento do sistema. Segundo Sommerville (2011): [são restrições aos serviços ou funções oferecidas pelo sistema]. Desta forma, conseguimos extrair os seguintes:

|  |  |  |
| --- | --- | --- |
| Código | Atributos de Qualidade | Descrição |
| RNF1 | Usabilidade | O sistema possui uma interface amigável com as seguintes características:  Idioma do sistema em Português, botões com ícones que representam a funcionalidade, área de trabalho atractiva que permite que os usuários se adaptem facilmente.  Conta ainda com uma selecção de cores que não cansam a visão mesmo em longas horas de visualização. |
| RNF2 | Segurança | O sistema possui um mecanismo de autenticação por login e autorização por permissões de acesso.  E possui também um algoritmo de criptografia AES. |
| RNF3 | Confiabilidade | O sistema é capaz de manter o seu normal funcionamento mesmo em situações de grandes requisições. |
| RNF4 | Manutenibilidade | O sistema foi concebido com boas praticas de OO e arquitectura em 3 camadas que facilita a manutenção, o teste e muito mais… |
| RNF5 | Escalabilidade | O sistema está preparado para escalabilidade vertical ou horizontal de acordo as necessidades dos usuários. |
| RNF6 | Portabilidade | O sistema está preparado para ser executado em outras plataformas diferente da que foi desenvolvida e com recurso que permitam o ajuste em diferentes telas de dispositivos. |

Tabela 2.3.3: Requisitos Não Funcionais

* + 1. Regras de Negócio

As regras de negócio são critérios e restrições estabelecidas pelas empresas para impulsionar o sucesso da mesma.

Regras de negócio tornam-se requisitos, ou seja, podem ser implementadas em um sistema de software como uma forma de requisitos desse sistema (WIKIPEDIA, 2020).

|  |  |
| --- | --- |
| Código | Descrição |
| RN1 | -O sistema não deve permitir o cadastro com campos obrigatório vazio. |
| RN2 | -O sistema não deve permitir o cadastro de usuários com o mesmo nome de login |
| RN3 | -O sistema não deve permitir o cadastro de documento já cadastrados(duplicação). |
| RN4 | -O sistema não deve permitir o cadastro de munícipe com o documento de identificação caducada. |
| RN5 | - O sistema não deve permitir o cadastro de munícipe com a data de nascimento superior a data actual. |
| RN6 | -O sistema não deve permitir a emissão de documentos do munícipe caso a documentação de identificação esteja expirada. |
| RN7 | - O sistema não deve permitir a emissão de documentos do munícipe caso este não tenha efectuado o pagamento. |

Tabela 3.3.4: Regras de Negócios

Análise do Sistema

A qualidade de um produto de software, aumenta à medida que se aplica a análise de seu desenvolvimento, pois quando os atributos de uso do sistema são bem claros e de certo modo a eficácia do produto final é emitente. Desta forma a análise permite-nos transformar todas ou quase todas as necessidades dos usuários do sistema em requisitos que farão funcionar o nosso sistema.

Diagramas do Sistema

Um diagrama é um gráfico que apresenta informações de forma esquematizada e relacionada a alguma área do saber. Após a recolha e análise dos requisitos, apoiamo-nos as ferramentas que possibilitam a criação de diagramas na linguagem na linguagem uml, para assim compor os digramas do sistema.

Diagrama de caso de Uso

Um diagrama de caso de uso é um diagrama que descreve a relação entre actores e casos de utilização de um dado sistema, é um diagrama que permite dar a visão global e de alto nível do sistema, sendo fundamental a definição correcta da sua fronteira (SILVA, 2020).

Vale recordar que, para realização dos casos de uso conforme mostra a figura abaixo, é necessário que os utilizadores estejam logado no sistema, com as credenciais devidamente autorizadas.

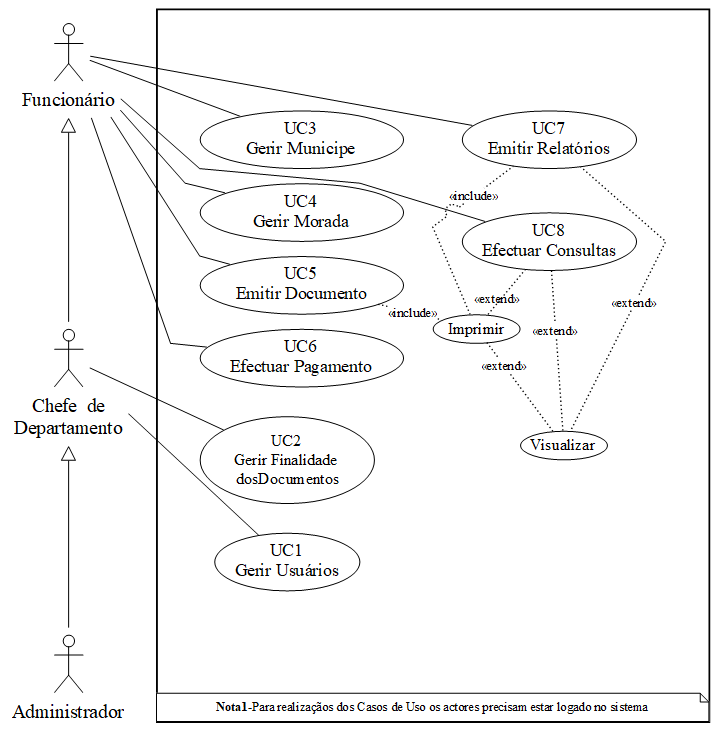


Figura 3:Diagrama de Casos de Uso.

* + 1. Descrição dos Principais Casos de Uso

A tabela abaixo, descreve o caso de uso cadastrar usuário, pois são estes usuários que irão efectuar a maior parte das actividades no sistema. Para tal o administrador deverá também conceder acesso a cada usuário que for cadastrado de modo que o mesmo possa usar o sistema com a permissões concedidas.

|  |  |
| --- | --- |
| Gerir Usuários | |
| Descrição: | Este caso de uso permite gerir os usuários que irão manipular o sistema. |
| Pré-Condição: | 1. Estar logado no sistema. 2. Ter permissão para realização do caso de uso. |
| Pós-Condição: | Dados geridos com sucesso |
| Actores: | Administrador e Chefe de Departamento |
| Fluxo Principal: | 1. O actor acede o menu Sistema. 2. Clicar no menu usuários. 3. O sistema apresenta a tela de usuários cadastrados o ator tem as opções de cadastrar, alterar ou excluir um usuário. 4. Para cadastrar ator deve clicar em no botão “Cadastrar”. 5. O sistema apresentará a tela de cadastro de usuário com os campos para serem preenchidos. O usuário deve também seleccionar o perfil e escolher o estado. 6. O usuário deve clicar no botão “guardar”. |
| Fluxo Alternativo: | No 6º passo, o sistema pode apresentar a mensagem de erro e retorna ao 5º passo para que se possa corrigi-lo. |

Tabela 4.4.3.1: Detalhamento do Caso de Uso Cadastrar Usuário

|  |  |
| --- | --- |
| Gerir Munícipe | |
| Descrição: | Este caso de uso permite o funcionário gerir os munícipes. |
| Pré-Condição: | 1. Estar logado no sistema. 2. Ter permissão para realização do caso de uso. |
| Pós-Condição: | Dados do Munícipe gerido com sucesso |
| Actores: | Administrador, Chefe de Departamento e Funcionário. |
| Fluxo Principal: | 1. O actor acede o menu Munícipe. 2. Clicar no menu Listar. 3. O sistema apresenta a tela de Munícipes cadastrados, o ator tem as opções de cadastrar, alterar e excluir um munícipe. 4. Para cadastrar, o actor deve clicar em “cadastrar”. 5. O sistema apresentará a tela de cadastro de Munícipe com os campos para serem preenchidos. O usuário deve também seleccionar o Tipo de Documento de Identificação, o estado civil e o género. 6. O actor deve clicar no botão “guardar”. |
| Fluxo Alternativo: | No 6º passo, o sistema pode apresentar a mensagem de erro e retorna ao 5º passo para que se possa corrigi-lo. |

Tabela 5.4.3.2**:** Detalhamento do Caso de Uso Cadastrar Munícipe

|  |  |
| --- | --- |
| Gerir Moradas | |
| Descrição: | Este caso de uso permite o funcionário gerir as moradas. |
| Pré-Condição: | 1. Estar logado no sistema. 2. Ter permissão para realização do caso de uso. |
| Pós-Condição: | Morada gerida com Sucesso |
| Actores: | Funcionários |
| Fluxo Principal: | 1. O actor acede o menu munícipe. 2. Clicar no menu listar. 3. O sistema apresenta a tela de munícipes cadastrados. 4. O actor seleciona o munícipe e clica no ícone . 5. O sistema apresentará a tela de cadastro de morada. O actor deve seleccionar o município, a comuna, o bairro e preencher os campos rua, zona e número da casa. 6. O actor deve clicar no botão “guardar”. |
| Fluxo Alternativo: | No 6º passo, o sistema pode apresentar a mensagem de erro e retorna ao 5º passo para que se possa corrigi-lo. |

Tabela 7.4.3.4**:** Detalhamento do Caso de Uso Cadastrar Morada

|  |  |
| --- | --- |
| Emitir Documento | |
| Descrição: | Este caso de uso permite o funcionário emitir documentos. |
| Pré-Condição: | 1. Estar logado no sistema. 2. Ter permissão para realização do caso de uso. |
| Pós-Condição: | Documento emitido com Sucesso |
| Actores: | Funcionários |
| Fluxo Principal: | 1. O actor acede o menu Secretaria. 2. Clicar no Submenu Emitir. 3. O sistema apresenta a tela de Seleção de tipos de Documentos a ser emitido. 4. O actor deve selecionar o tipo de Documento, e a finalidade. 5. O actor deve ler o código de barras, código Qr ou digitar o número do munícipe e clicar em buscar. 6. O Sistema apresentará os dados do munícipe. 7. O actor deve selecionar a forma de pagamento e introduzir os dados. 8. O actor deve clicar em “emitir documento”. |
| Fluxo Alternativo: | No 6º passo, o sistema pode apresentar a mensagem de erro e retorna ao 5º passo para que se possa corrigi-lo.  No 7º passo, o sistema pode apresentar a mensagem de erro e retorna ao 6º passo para que se possa corrigi-lo.  No 8º passo, o sistema pode apresentar a mensagem de erro e retorna ao 5º passo para que se possa corrigi-lo. |

Tabela 9.4.3.6: Detalhamento do Caso de Uso Emitir Documento

|  |  |
| --- | --- |
| Efectuar Pagamento | |
| Descrição: | Este caso de uso permite o funcionário emitir documentos. |
| Pré-Condição: | 1. Estar logado no sistema. 2. Ter permissão para realização do caso de uso. |
| Pós-Condição: | Pagamento efetuado com sucesso |
| Actores: | Funcionários |
| Fluxo Principal: | 1. O actor acede o menu Secretaria. 2. Clicar no Submenu Emitir. 3. O sistema apresenta a tela de Seleção de tipos de Documentos a ser emitido. 4. O actor deve selecionar o tipo de Documento, e a finalidade. 5. O actor deve ler o código de barras, código Qr ou digitar o número do munícipe e clicar em buscar. 6. O Sistema apresentará os dados do munícipe. 7. O actor deve selecionar a forma de pagamento e introduzir os dados. 8. O actor deve clicar em “emitir documento”. |
| Fluxo Alternativo: | No 7º passo, o sistema pode apresentar a mensagem de erro e retorna ao 6º passo para que se possa corrigi-lo. |

Tabela 10.4.3.7: Detalhamento do Caso de Uso Efectuar Pagamento

* + 1. Diagrama Entidade-Relacionamento

O diagrama entidade-relacionalmente ou modelo entidade-relacionamento é um modelo conceitual usado para descrever objectos envolvidos no domínio de um sistema a ser construído, incluindo seu atributos e relacionamentos, o MER permite representar de forma abstrata a estrutura que irá construir o banco de dados. Neste ponto podemos ilustrar com a seguinte figura o MER do nosso sistema:



Figura 4:Diagrama Entidade-Relacionamento.

## Diagrama Lógico

“Um diagrama logico é um diagrama que possui conceitos que os usuários são capazes de entender, ao mesmo tempo em que não está distante do modelo físico do banco de dados.

Neste nível o projecto é independente do SGDB e consiste na especificação logica dos dados em um formato adequado ao SGDB escolhido, os tipos de dados, são completamente definidos” (REIS, 2018)

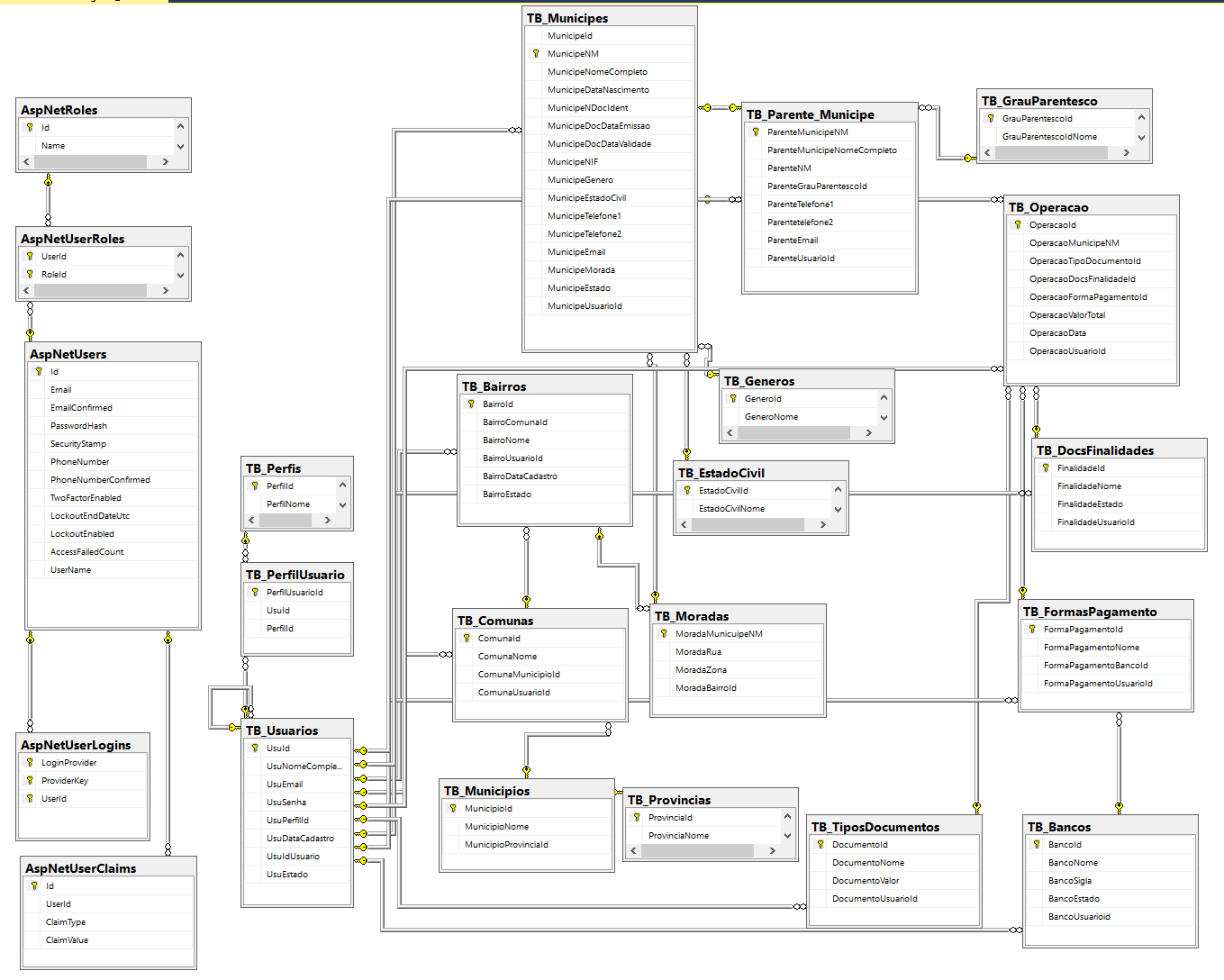


Figura 5:Diagrama Lógico.

* + 1. Diagrama de Classe

Apoiando-se as palavras de Silva; Videira (2001, p. 186) constatamos que

[…] Os diagramas de classes são usados para modelar a estrutura de um sistema. Estes modelos são também designados por vista do desenho estático do sistema e são usados tipicamente em três situações, (1) para modelar o vocabulário de um sistema; (2) para modelar colaborações simples; e (3) para modelar o desenho de um esquema de uma base de dados.

Assim sendo, através da análise do sistema consegue-se extrair o diagrama apresentado na figura abaixo:

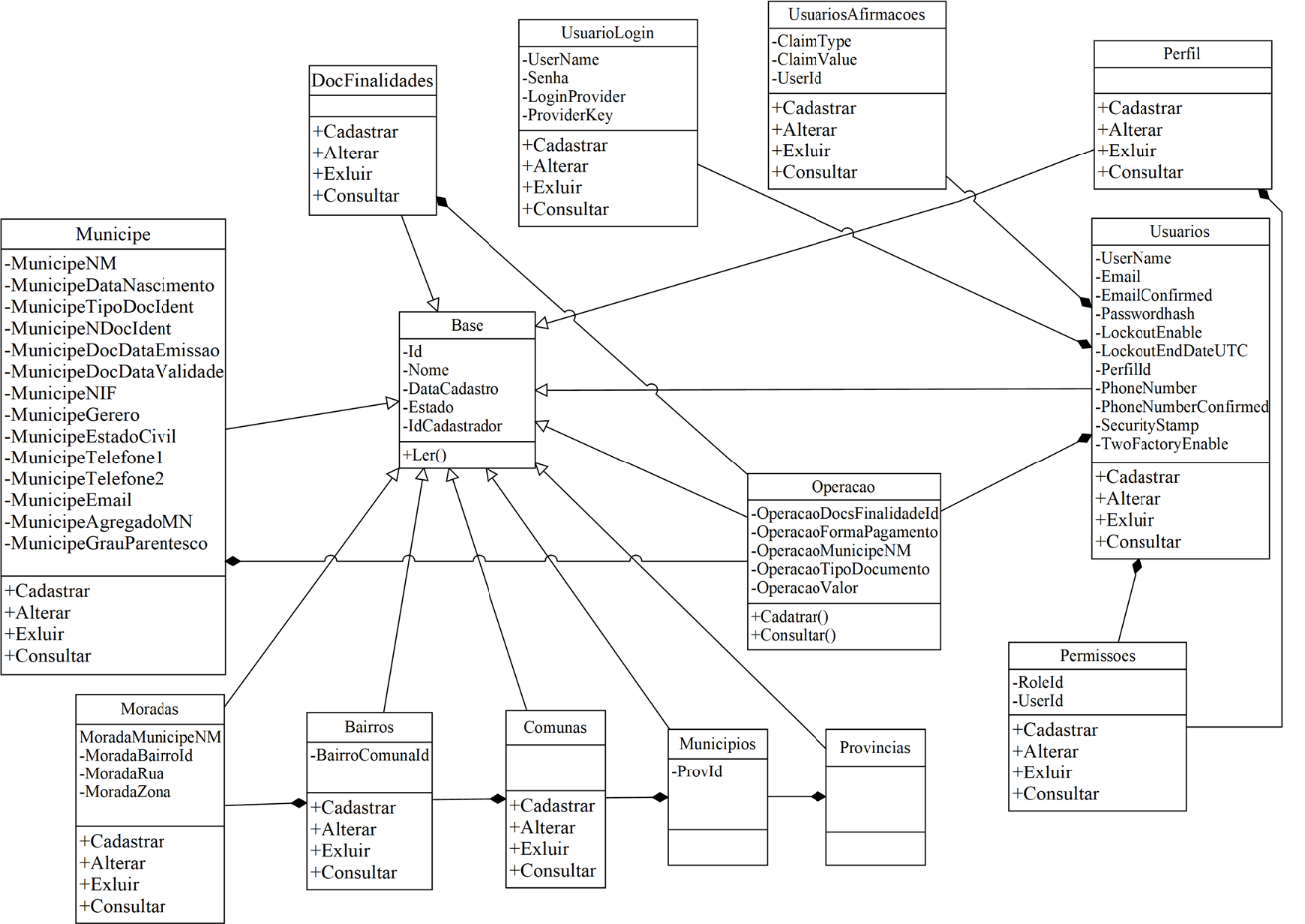


Figura 6: Diagrama de Classes

* + 1. Diagrama de Sequência

Silva; Videira (2001), afirma que o diagrama de sequência ilustra uma interacção segundo uma visão temporal, é representado através de duas dimensões: a dimensão horizontal, que representa um conjunto de objectos intervenientes; e a dimensão vertical que representa o tempo.

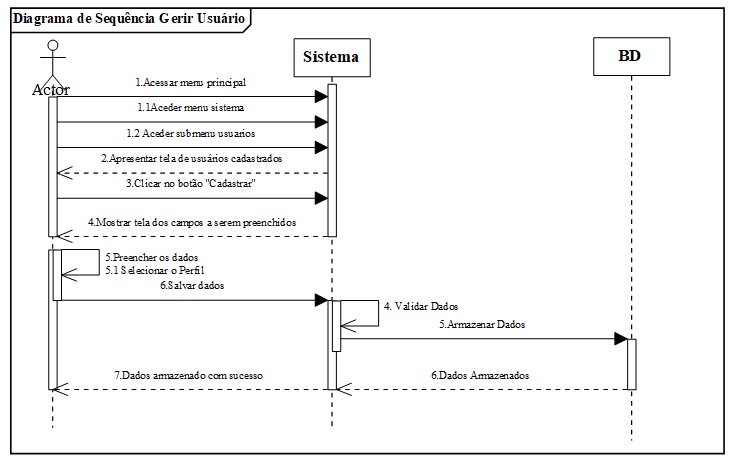


Figura 7:Diagrama de Sequência Gerir Usuários

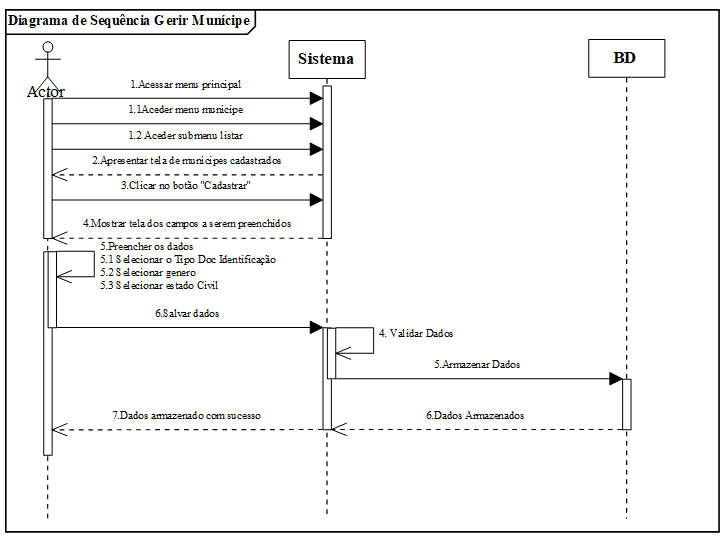


Figura 8:Diagrama de Sequência Gerir Munícipe

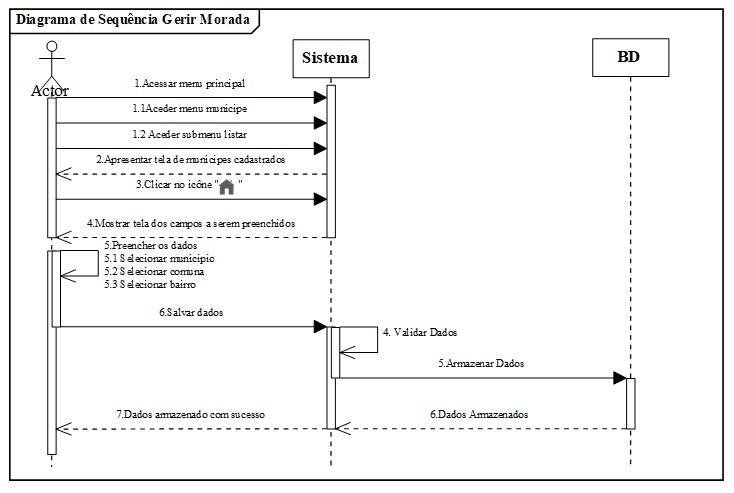


Figura 9:Diagrama de Sequência Gerir Morada

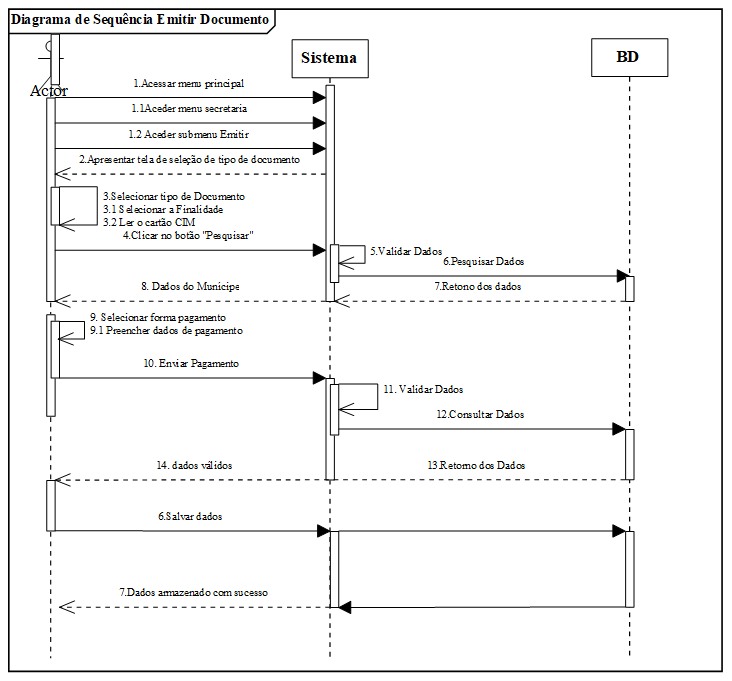


Figura 10:Diagrama de Sequência Emitir Documento

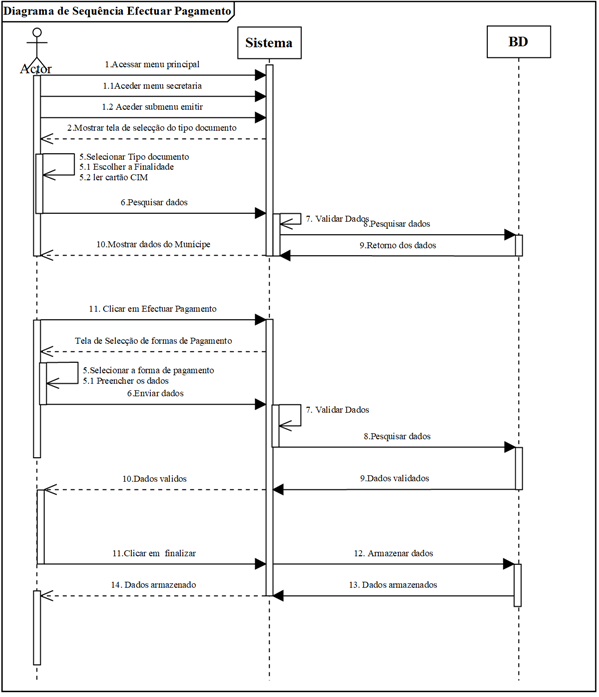


Figura 11:Diagrama de Sequência Efectuar Pagamento

* + 1. Diagrama de actividade

Um diagrama de actividades é um caso particular de um diagrama de estados, no qual todos ou a maioria dos estados são estados de actividade e todas ou a maioria das transações são desencadeadas pela conclusão das actividades dos estados anteriores (SILVA e VIDEIRA, 2001).

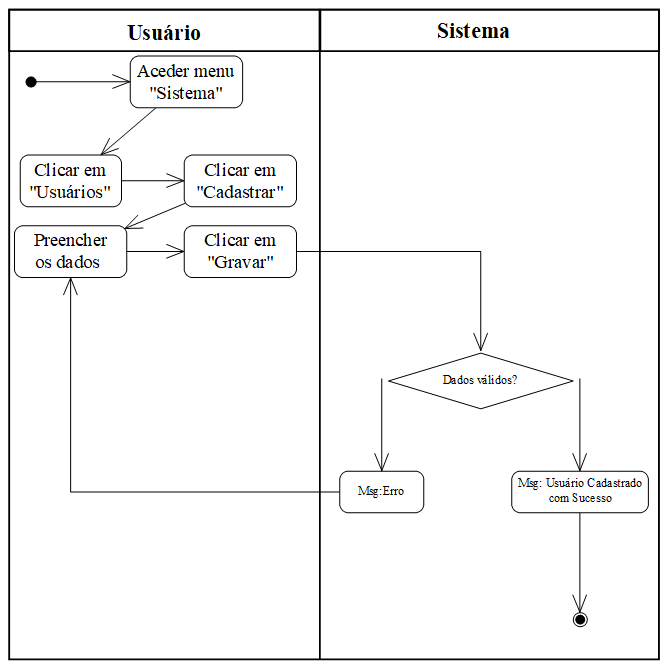


Figura 12:Diagrama de Actividade Cadastrar Usuário.

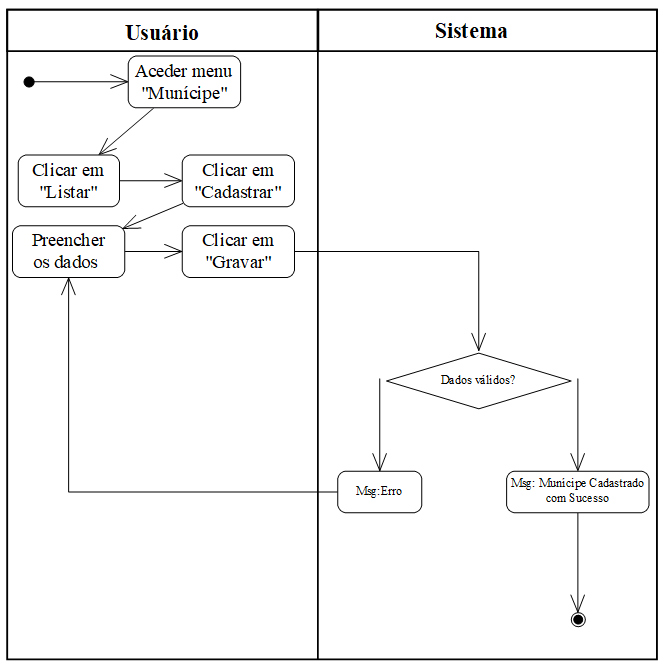


Figura 13:Diagrama de Actividade cadastrar munícipe

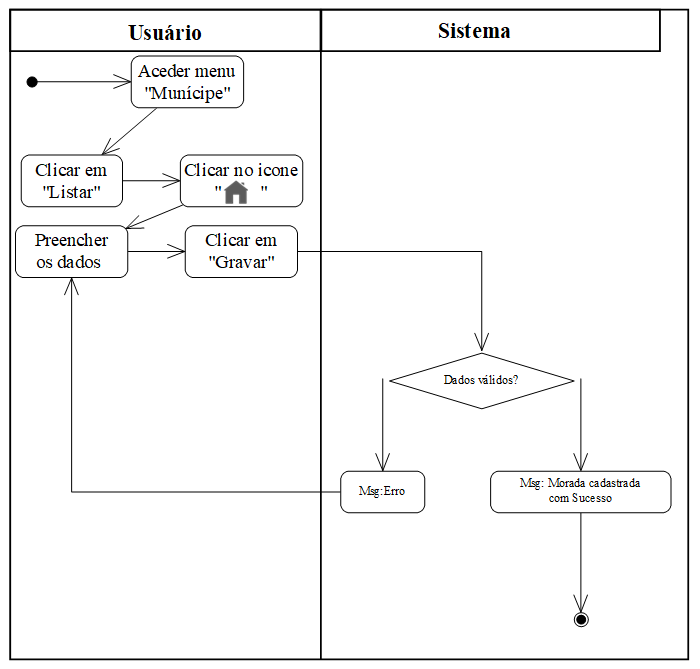


Figura 14:Diagrama de Actividade Cadastrar Morada.

.

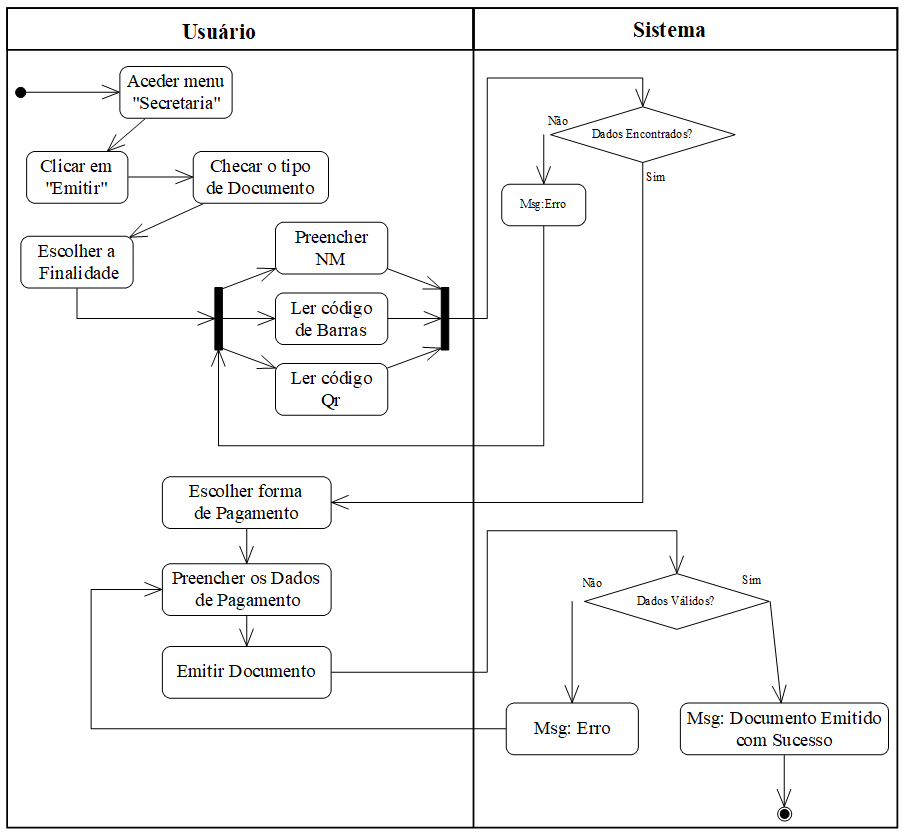


Figura 15:Diagrama de Actividade Emitir Documento.

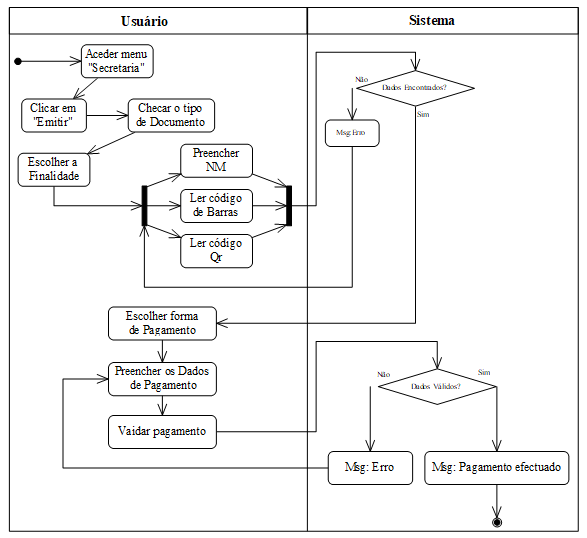


Figura 16:Diagrama de Actividade Efectuar Pagamento

* + 1. Diagrama de Instalação

Segundo Ramos (2013): […] um diagrama de instalação ilustra a configuração dos elementos de processamento e dos componentes software, processos e objectos neles suportados.

Outrossim, Silva; videira (2001) afirma que um diagrama de instalação, também designado nalgumas circunstâncias por diagramas de distribuição, consiste num conjunto de nós ligados por associações de comunicação.



Figura 17:Diagrama de Instalação

## Ferramentas de Desenvolvimento

As ferramentas de desenvolvimento de software servem para diferentes aspectos do desenvolvimento de software, como a codificação propriamente dita, compilação, teste, depuração, armazenamento de dados, controle de versões e debugs, implantação e execução da aplicação desenvolvida. Deste modo as ferramentas que serviram de suporte para a construção deste sistema são:

* Visual Estúdio 2019 - IDE- Integrated Development Environment ou seja Ambiente de Desenvolvimento Integrado da Microsoft que possibilita os usuários escreverem seus códigos em uma determinada linguagem e para então serem traduzidas em comandos para o computador (blog Impacta, 2017).
* SQL Server 2019 - é um SGDB desenvolvido e mantido pela Microsoft para armazenamento, actualização e recuperação de dados computacionais, incluindo os relacionamentos e permitindo toda administração de forma local ou remota (Portal GSTI).
* DBforge Studio for SQL Server – é um IDE poderoso para gerenciamento, administração, desenvolvimento, relatórios de dados, analise e muito mais do Microsoft SQL server. Desenvolvedores de SQL e BDAs que executam tarefas complexas de banco de dados podem usar a ferramenta GUI para acelerar quase qualquer experiência de banco de dado, como projectar banco de dados, escrever códigos SQL, comparar bancos de dados, sincronizar esquemas e dados, gerar dados de teste significativos e muito mais (DEVART, 1997).
* GitHub Desktop - é um sistema de gerenciamento e projectos e versões de código criado para desenvolvedores e é um dos maiores depósitos online de trabalhos colaborativos do mundo. O GitHub desktop é um aplicativo criado pela Microsoft especificamente para computadores com Windows (MARQUES, 2019).
* JCPicker- é uma ferramenta para capturar a cor de qualquer pixel em qualquer área da tela e exibir seu código em uma variedade de formatos como: html, rgb, hex, hsb, hsv, hsl, hwb, cmy, cmyk e delphi (AnnyStudio, 2003).
* Enterprise Architect- é uma plataforma de desenvolvimento colaborativa para modelagem, design e gerenciamento baseada em uml e padrões similares. Uma solução completa e empresarial para visualização, analise, modelagem, teste e manutenção de uma grande variedade de sistema, softwares, processos e arquitecturas (software.com.br).
* BrModelo – é uma ferramenta de apoio ao projecto de um BD relacional desenvolvida pelo GBD/UFSC e consiste num aplicativo que permite a definição de modelagens conceituais e logicas para BDs relacionais com facilidade e independência de SGBD (RAMOS e MENNA, 2011).
* StarUML - é um modelador de software sofisticado destinado a suportar modelagem ágil e concisa (STARUML).

## Arquitectura lógica do Sistema

O sistema foi desenvolvido usando uma arquitectura em camadas, que visa a criação de aplicativos modulares, de forma que a camada mais alta se comunica com a camada mais baixa e assim por diante, fazendo com que uma camada seja dependente apenas da camada imediatamente abaixo, conforme podemos observar na figura 18.

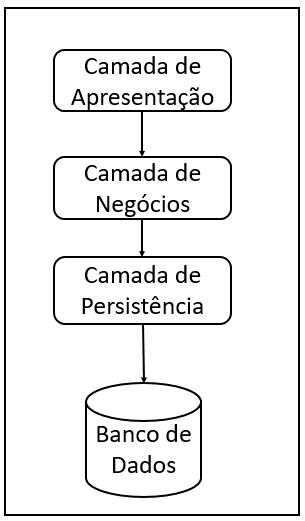


Figura 18: Arquitetura em 3 camadas

* Camada de Apresentação- logica de interface do usuário (GUI). O código responsável pela apresentação e controle da página e tela de navegação forma a camada de apresentação.
* Camada de Negócios- camada referente a implementação de regras de negócio ou requisitos do sistema.
* Camada de Apresentação-camada responsável por armazenamento e recuperação dos dados quando solicitado. O seu objectivo é garantir uma independência da fonte de dados (arquivos, bancos de dados, etc.) e ao mesmo tempo manter as informações entre diferentes sessões de uso.

Não obstante a adoção da arquitetura em camadas, neste projecto fizemos uma mesclagem entre as camadas acima referenciadas e o padrão arquitetural MVC, onde obtivemos os resultados conforme mostra a figura 19.



Figura 19:Arquitetura Lógica do Sistema

## Desenvolvimento de base de Dados

O desenvolvimento e implementação de uma base de dados é uma parte essencial do processo de engenharia de software. As mesmas são concebidas numa estrutura conceitual e modelo físico, tendo em conta a escalabilidade. As bases de dados relacionais são especificadas e implementadas respeitando as regras de normalização no sentido de minimizar a redundância dos dados.

* + 1. Criação de base de dados

Para a criação da base de dados deste projecto, utilizamos o Microsoft SQL Server 2019 Developer Edition que é uma solução de persistência de dados da Microsoft. Escolhemos o SQL server porque é umas das soluções que atende as características do nosso projecto. Nesta sua versão implementou algumas melhorias como:

* Big Data+ Analytics;
* Melhorias de Perfomance;
* Segurança Avançada-Computação Confidencial;
* Disponibilidade de missão critica-alto tempo de uptime;
* Melhorias no Multiplataforma;
* Suporte ao SQL Server 2019 no Azure data Studio
* Novas mensagens na sys.messages;
* Novos objectos de sistema e DMW´s (DIRCEU, 2013).

## Tecnologias envolvidas no Projecto

* + 1. Base de Dados

Segundo (JONES, 2013) base de dados é um conjunto de dados relacionados de acordo com uma ou várias regras e objectivo específico. Consiste numa colecção de dados estruturados e armazenados de forma persistente. Eis algumas vantagens de uso:

* Diminuição de espaço físico ocupado;
* Maior integridade dos dados
* Menor redundância;
* Mais facilidade na partilha de dados
* Maior facilidade de manutenção;
* Isolamento entre objectos de dados, oque protege a integridade da origem dos dados;
* Facilidade de mudança na criação de diferentes mapas com diferentes objectivos.
  + 1. Sistemas de gestão de base de dados

Segundo (JONES, 2013) Um sistema de gestão de bases de dados é uma aplicação ou um conjunto de aplicações informáticas utlizadas para definir, aceder e gerir os dados existentes numa base de dados. Seu objectivo é registar e manter a informação que for considerada necessária á organização/pessoa que gere o sistema, disponibilizando-a automaticamente para os mais diversos fins.

Através de um SGDB é pode-se realizar um vasto conjunto de operações de gestão de dados das quais é possível destacar:

* Inserção, edição e apagamento de registos;
* Critérios de visualização da informação;
* Indexação e ordenação da informação contida nos registos;
* Operações estatísticas sobre os dados;
* Criação de ecrãs de apresentação
* Automatização de funções;
* Programação.
  + 1. SQL

O SQL é uma linguagem padrão universal para manipular bancos de dados relacionais através dos SGBDs, isso significa que todos os sistemas de gerenciamento de banco de dados relacionais oferecem uma interface para acessar o banco de dados utilizando a linguagem SQL, embora como algumas variações.

A linguagem estruturada de consultas é utilizada com SGBD e executar varia tarefas como inserir e alterar, criar objectos no banco de dados, gerenciar usuários, consultar informações, controlar transacções, etc.

A linguagem SQL está divida em 4 agrupamento conforme ilustra a figura:

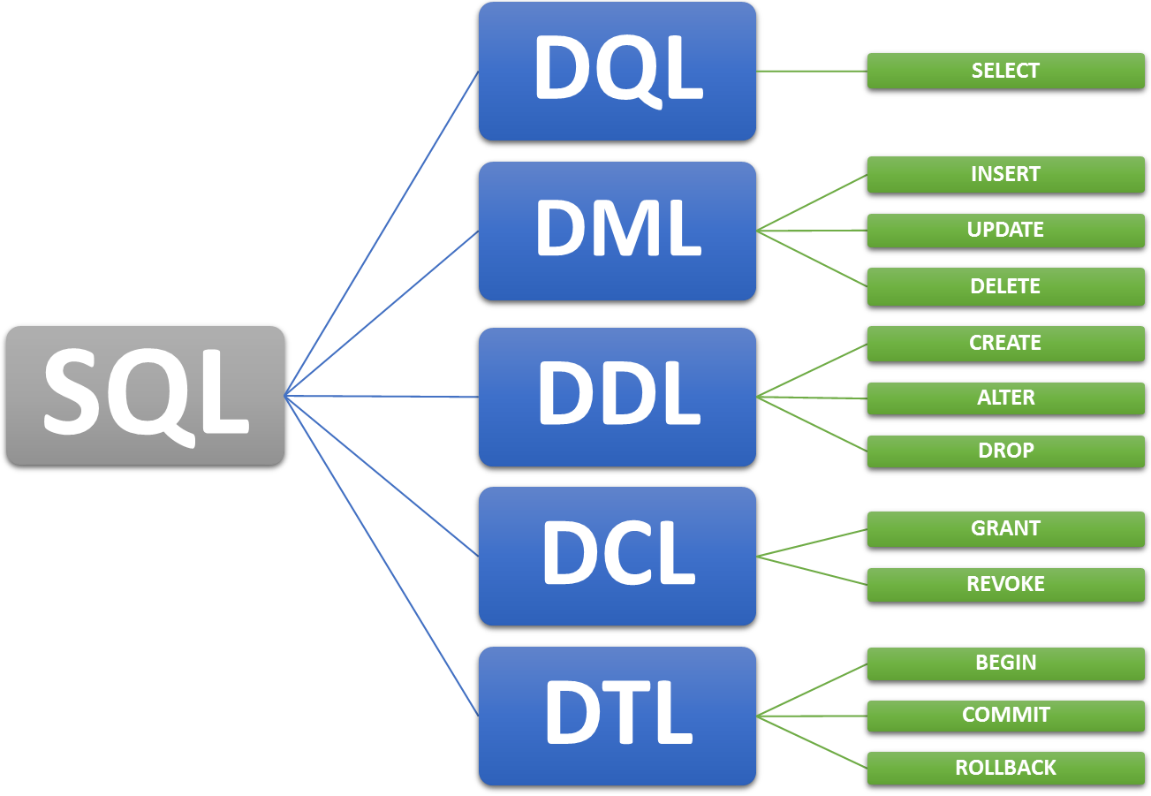


Figura 20: Linguagens do SQL

Fonte: [www.devmedia.com.br](http://www.devmedia.com.br), Agosto,2020.

* DML-Data Manipulation Language, é um subconjunto mais utilizado da linguagem, pois através dela operamos os dados dos bancos de dados e compreende os seguintes comandos: INSERT, UPDATE, DELETE, SELECT.
* DLL-Data Manipulation Language- subconjunto da linguagem que gerência a estrutura do banco de dados e compreende os seguintes comandos: CREATE, ALTER e DROP.
* DCL-Data Control Language-subconjunto para controlar o acesso aos dados e possui basicamente dois comandos que permite ou bloqueia o acesso de usuário a dados: GRANT e REVOVE.
* DTL-Data Transation Language subconjunto que fornece mecanismo para controlar transacções no banco de dados, compreende 3 comandos que são: BEGIN TRANSATION, COMMIT e ROLLBACK (ALVES, 2007).
  + 1. Procedimentos armazenados

Kassequele citou que procedimento armazenando é um conjunto de instruções para solucionar um determinado problema (informação pessoal)[[1]](#footnote-1).

Um procedimento armazenado é um grupo de uma ou mais instruções Transact-SQL ou uma referência a um método CLR do Microsoft .Net framework. Eis algumas vantagens da sua aplicabilidade:

* Tráfego de rede de servidor/cliente reduzido;
* Segurança mais forte;
* Reutilização de código;
* Manutenção facilitada;
* Desempenho aprimorado (MICROSOFT CORPORATION, 2017).
  + 1. Entity Framework 6.0

O Entity framework é uma ferramenta da Microsoft, pertencente ao pacote ado.net que permite ao desenvolvedor trabalhar com banco de dados relacional de maneira atrata, focando nas regras do negócio e abstraindo a camada de banco de dados (LEITE, 2015).

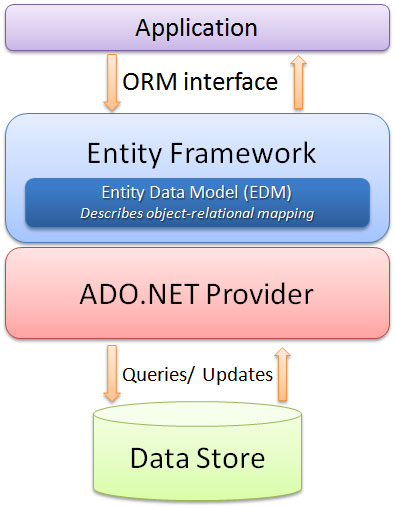


Figura 21:Arquitetura do Entity Framework

Fonte: [www.redspark.io](http://www.redspark.io), Agosto,2020.

A figura acima, apresenta a arquitectura do ORM e existem 3 formas de se trabalhar com o Entity Framework, a saber:

* Database-Firt: nesta forma, o banco de dados já existe, as classes (entidades), são criadas baseadas nele;
* Model-First: o banco de dados não existe, é criado um modelo conceitual e através dele é gerado um script para a criação.
* Code-First: talvez a forma que melhor utiliza os recursos do Entity Framework. O Banco ainda não existe, são criadas as classes simples, através delas toda logica da entidade é criada e com este modelo é gerado o banco de dados.

## Programação

* + 1. Asp.NET 4.8

O asp.net é uma tecnologia de scripting do lado do servidor, esta tecnologia permite colocar numa página web, scripts que irão ser executados por um servidor. ASP significa Active Server Page e corre no servidor IIS que vem instalado e é gratuito nos servidores Windows.

Na sua versão 4.8 implementa melhorias de qualidade e confiabilidade e correcção de vários problemas da versão 4.7.2 (MICROSOFT, 2019).

* + 1. Asp.net MVC

O MVC é uma arquitectura que divide uma aplicação em três partes: Model- responsável pela manutenção do estado dos dados(armazenamento dos dados); View- responsável pela interface com o usuário(não deve possuir nenhuma logica de aplicação nem acesso a dados); e o Controler- que controla a logica da aplicação, manipulando os dados da Model e escolhendo a view a ser utilizada. Esta arquitectura permite a clara separação dos papeis de cada componente, oque facilita a execução de teste unitários na aplicação, entre outros benefícios (ONEDA, 2018).



Figura 22:Modelo Arquitetural MVC.

Esta arquitetural tem como principal objectivo isolar mudanças na GUI, evitando que estas mudanças acarretem em mudanças na cama de negócios da aplicação. E apresenta algumas vantagens como, facilitação na manutenção onde as mudanças na logica de negócio têm menos probabilidades de quebrar a logica da apresentação e facilidade no desenvolvimento por times multidisciplinares, onde os desenvolvedores podem criar códigos de negócios robusto e os designers interfaces gráficas utilizáveis e envolventes.

* + 1. Booststrap

Boostrap é um framework front-end que facilita a vida dos desenvolvedores web a criar sites com tecnologia mobile(responsivo) sem ter que digitar uma linha de CSS para fazer acontecer, não é atoa que o termo inglês bootstrap significa “inicialização”, algo que possui um ponto de partida (COSTA, 2014). Neste projecto foi utilizado para garantir um visual agradável e responsivo.

* + 1. Jquery

O Jquery é uma biblioteca leve, rápida e cheia de recursos JavaScript, que facilita a manipulação de eventos, animações, elementos HTML e utilização de Ajax (BRASIL, 2019).

* + 1. Validações

A validação de software ou mais genericamente, verificação e validação(V&V), tem a intenção de mostrar que um software se adequa a suas especificações ao mesmo tempo que satisfaz as especificações do cliente do sistema (SOMMERVILLE, 2011).

Para validação dos campos do formulário contidos no projecto, utilizados a biblioteca Jquery, conforme ilustra a figura abaixo:

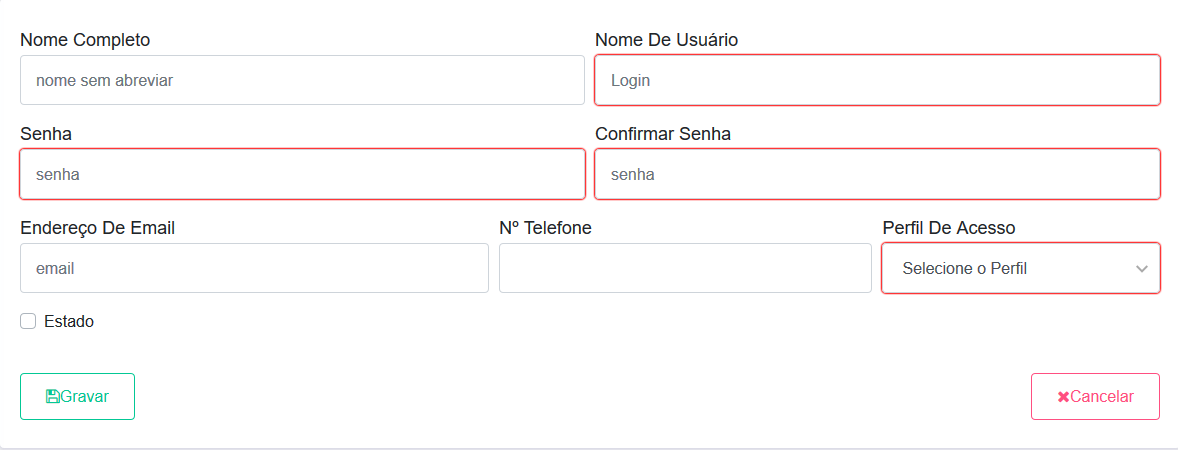


Figura 23:Formulário com validação dos campos.

* + 1. Scaffolding

Maniero afirmou que o scaffold é uma técnica antiquíssima de geração de código baseado em gabaritos de operações comuns que costumam ser usadas em aplicações. Ao contrário do que muita gente acredita, provavelmente fruto de ter aprendido por receita de bolo, existem várias técnicas para obter o resultado. É possível realizar isto em tempo de execução ou em tempo de desenvolvimento, gerando código que são agregados ao projecto como se fossem escritos por uma pessoa (Informação Pessoal)[[2]](#footnote-2).

Esta técnica permitiu-nos gerar rapidamente os códigos para efectuação do CRUD das entidades que fazem parte do nosso projecto.

* + 1. Qr Code

O Qr Code ou código Qr é um código de barras ou podemos chamar de código barrametrico bidimensional que pode ser facilmente escaneado usando a maioria dos telefones celulares equipados com câmara. Esse código é convertido em texto interativo, uma url, um número de telefone, uma localização georreferenciada, um email, um contacto ou uma SMS.

O padrão japonês para o código qr, JIS X 0510, foi lançado em janeiro de 1999 e corresponde ao padrão internacional ISSO/IEC 18004, tendo sido aprovado em 2000.

Neste projecto, implementamos o código Qr para garantir a autenticidade e a veracidade da documentação emitida, sendo que ao ler o código, o mesmo retornara as informações que estiverem gravados no documento. O Qr também providencia um atendimento rápido e fácil ao munícipe.



Figura 24:Código Qr gerado pela Sistema

* + 1. Bar Code

O código de barras é uma representação gráfica de dados numéricos ou alfanuméricos. A decodificação (leitura) dos dados é realizada por um tipo de scâner, o leitor de código de barras, que emite um raio vermelho que percorre todas as barras. Onde a barra se for escura, a luz é absorvida; onde a barra for clara (espaços), a luz é refletida novamente ao leitor. Os dados capturados nessa leitura óptica são compreendidos pelo computador, que por sua vez converte-os em letras ou números legíveis aos humanos.



Figura 25:Código de Barras Gerado pelo sistema

* + 1. Nuget

Segundo NASCIMENTO (2020) “publica que Nuget é um gerenciador de dependências para a plataforma .NET. Ele define como os pacotes desta plataforma são criadas, publicados e consumidos, fornecendo ferramentas para cada uma dessas funções.

Na pratica um pacote Nuget é um arquivo compactado com a extensão .nuppkg que contém um código compilado (DLL), outros arquivos relacionados a este código (como imagens, etc.) e um arquivo de configuração que contém informações sobre o pacote, como numero de versão, criador, etc. estes pacotes .nupkg podem ser publicados em repositórios públicos ou privado onde os desenvolvedores podem consumi-los, adicionando-os em seus projectos e utilizando no código as funcionalidade que fornecerem”.

Para facilitar a distribuição dos pacotes, o Nuget mantem um repositório publico, o Nuget.org, onde qualquer desenvolvedor pode publicar, pesquisar e consumir pacotes.

Independentemente do tipo do repositório, ele funciona como a ligação dentre os criadores dos pacotes e os desenvolvedores que os consome. Podemos ilustrar isso com a figura abaixo:



Figura 26:Diagrama de Distribuição de Pacotes Nuget

Fonte: [www.treinaweb.com.br](http://www.treinaweb.com.br), Maio,2020.

* + 1. ASP.NET Identity 2.1

Segundo Pires (1983) “O ASP.Net Identity é um componente de autenticação e autorização que foi disponibilizado pela Microsoft junto com ASP.NET MVC 5, em substituição ao ASP.NET MemberShip e Simple MemberShip Systems. Na sua versão 2.1 que foi lançada em 2013, trouxe algumas características desde a sua primeira versão como:

* Parte do ONE ASP.NET;
* Customização do Perfil do usuário simplificado (Escrito em Code First);
* Controle de persistência de dados (EF ou outros);
* Totalmente testável (Unity Tests);
* Role Provider (Separação de acessos por Perfis);
* Claims Based;
* Autenticação com redes sociais (FB, Twitter, Google+ e Microsoft Acoounts);
* Integração com Active Directoty (On-Premisses e Azure);
* Integração com OWIN (OWIN Middleware Based);
* Entregue via NuGet (Nuget Everwhere)”.

Recorremos a este componente para providenciar o mecanismo autenticação e autorização dos usuários que farão uso no sistema.

* + 1. Ado.NET

Conforme Húngaro (2014), “O ADO.NET é um conjunto de classes do .NET Framework, desenvolvidas para facilitar o acesso das aplicações à bases de dados de diversos tipos, especialmente bancos de dados como Access, SQL Server, Oracle”.

Pensando no mundo conectado ao qual vivemos hoje, a Microsoft desenvolveu o ADO.NET com as características especiais conforme mostra a figura:



Figura 27:Modelo de Objectos do ADO.Net

Fonte: [www.linhadecodigo.com.br](http://www.linhadecodigo.com.br), Maio,2020.

Neste trabalho usamos este conjunto de classes, pois devido a sua implementação, manuseio e testabilidade, quanto a persistência dos dados de forma que eficaz e garantindo maior desempenho tanto no transmissão e recepção dos dados.

* + 1. Json

Segundo o Site Wikipedia.org, “JSON é um acrónimo de Javascript Object Notation, é um formato compacto de padrão aberto independente, de troca de dados simples e rápida(pasing) entre sistemas, especificado por Douglas Crockford em 2000, que utiliza o texto legível a humanos, no formato atributo-valor (natureza auto-descritiva).

Diferenças o Json em relação ao XML:

* Não é uma linguagem de marcação. Não possui tags de abertura e fechamento;
* Representa as informações de forma mais compacta.
* Não permite a execução de instruções de processamento, enquanto é possível em XML.
* É tipicamente destinado para troca de informações, enquanto xml possui mais aplicações”.

## Fase de Testes

Relativamente aos testes e consequentemente a qualidade do software, surgiu então a norma ISSO/IEC 9126, com objectivo de avaliar a qualidade do produto de software. Esta norma é composta por vários atributos e métricas que devem ser abordados em um software para que ele seja caracterizado um “software de qualidade”.

O teste é destinado a mostrar que um programa faz o que é proposto e para descobrir os defeitos do programa antes do uso. Quando se testa o software, o programa é executado usando dados fictícios e os resultado do teste são verificados à procura de erros , anomalias ou informações sobre atributos não funcionais do programa (SOMMERVILLE, 2011).

Entende-se como teste de software a investigação do software ou parte do mesmo a fim de fornecer informações sobre a sua qualidade em relação ao contexto em que o mesmo deve operar. Podemos então concluir que testar = Executar/usar o software (informação verbal)[[3]](#footnote-3).

* + 1. Teste de Unidade

O teste de unidade ou teste unitário é o processo de testar os componentes de programa, como métodos ou classes de objecto. As funções individuais ou métodos são o tipo mais simples de componente (SOMMERVILLE, 2011).

Os testes unitários em softwares são efectuados durante a fase de codificação pelo próprio programador ou um code reviewer[[4]](#footnote-4). Deste modo, a medida que fomos codificando as classes, objectos e métodos, fomos testando para garantir a operacionalização correcta. Esse teste foi realizado de forma manual com objectivo de corrigir os bugs mais cedo possível enquanto durar a fase de desenvolvimento e para também poupar custos com re-works e correcções tardias.

* + 1. Teste de Sistema

O teste de sistema verifica se os componentes são compatíveis, se interagem correctamente e transferem os dados certos no momento certo, por suas interfaces (SOMMERVILLE, 2011).

Entende-se também como testes de sistema o tipo de testes para verificar o comportamento completo só software baseado no documento de especificação de requisitos do software, onde seu principal foco é avaliar os requisitos funcionais do negócio.

Neste ponto, testávamos todas as funcionalidades do sistema de modo a certificar-se que tais funcionalidades eram executadas e cumpriam com aquele que é o propósito da sua implementação.

* + 1. Teste de Aceitação

Entende-se como testes de aceitação o tipo de testes realizado pelo cliente e um grupo selecto de utilizadores finais (Keys-Users) para certificar o software em relação aos requisitos previamente acordados. Este teste acontece na fase final da fase de testes, antes que a aplicação seja movida para o ambiente de produção ou lançada para o mercado (BENGUI, 2019).

Infelizmente, por questões alheias a nossa vontade, não nos foi possível efectuar este tipo de teste visto que o a Administração Municipal de Malanje, se encontra a trabalhar com um número reduzido de efectivos por conta da pandemia que assola o mundo, em especial o nosso país.

## Metodologias de desenvolvimento de software

Já faz alguns anos que o desenvolvimento de software deixou de ser sinonimo de apenas código, hoje em dia, sabe-se que é necessária a utilização de uma metodologia de trabalho.

Entende-se de metodologia de desenvolvimento de software a maneira ou forma de se utilizar um conjunto coerente e coordenado de métodos para atingir um objectivo, de modo que se evite, tanto quanto possível, a subjectividade na execução do trabalho. O dicionário define como um conjunto de métodos, regras e postulado empregados por disciplina, ou um conjunto de procedimentos.

O objectivo da metodologia de desenvolvimento é definir de forma clara “quem” faz “o que”, “quando”, “como”, e até mesmo “onde”, para todos os que estejam envolvidos directamente ou não com o desenvolvimento de software.

Para a realização deste projecto, optou-se pelo modelo iterativo e incremental.

O modelo iterativo incremental é uma metodologia de desenvolvimento de software que define um subconjunto de requisitos e utiliza o modelo em cascata para a sua realização. Cada porção do ciclo segue o projecto de arquitectura inicial como guia, mas com uma abordagem bem menor. Uma vez satisfeita os requisitos e os objectivos da iteração forem completos, o desenvolvimento segue para a próxima iteração, a figura abaixo ilustra a sequência dos incrementos.

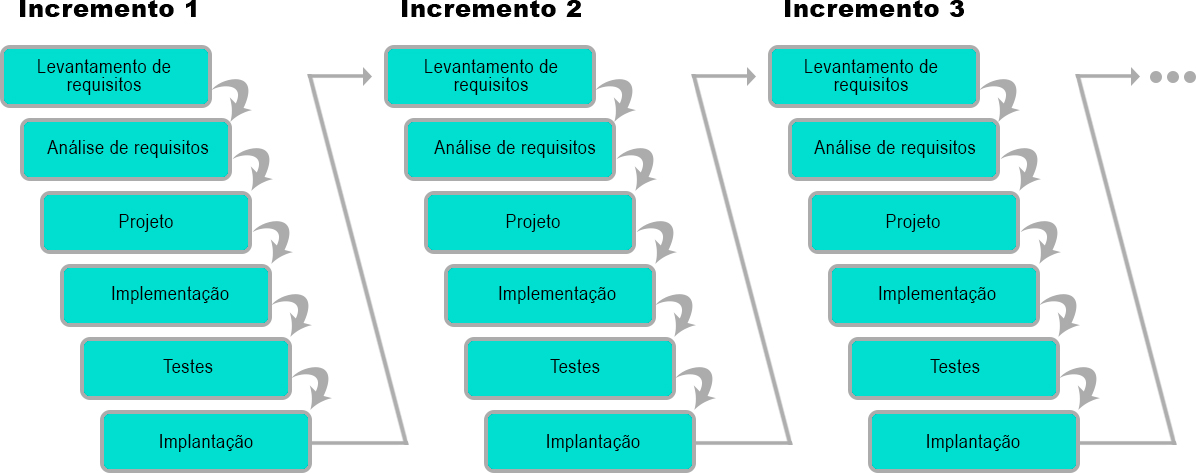


Figura 28: Modelo de Desenvolvimento iterativo e incremental

Fonte: [www.simulado.estacio.br](http://www.simulado.estacio.br), Agosto,2020.

A implementação de um processo de desenvolvimento iterativo e incremental está na base das metodologias ágeis, como a Scrum. A ideia é que a criação de um software seja pautada por vários ciclos curtos, em funcionalidades são introduzidas, feedbacks colectados e requisitos revistos. Assim é possível atingir um maior nível de satisfação do cliente e garantir que o resultado final esteja dentro do esperado.

A adoção de um processo de desenvolvimento iterativo e incremental pode trazer uma serie de vantagens para o negócio, elas vão além da melhora dos produtos entregues ao cliente. A seguir anotamos alguns benefícios ou vantagens de implementação deste modelo:

* Criação de um fluxo de entrega de softwares em que os requisitos são apresentados em pequenas partes funcionais;
* Maior capacidade de acompanhar a evolução do desenvolvimento da aplicação;
* Identificação precisa de erros e falhas durante a criação do sistema;
* Redução de riscos a cada etapa do projecto;
* Maior capacidade de modificar a direção de um projecto;
* Criação de soluções de software com maior valor agregado;
* Capacidade de otimizar a ferramenta continuamente;
* Escopo de software mais flexível;
* Mais agilidade e produtividade no dia a dia de cada time.
  + 1. Refatoração

A Refatoração é o processo de alterar um software de uma maneira que não mude o seu comportamento externo e ainda melhore a sua estrutura interna. Ele é utilizado para manter um software bem projectado mesmo com o decorrer do tempo e as mudanças que ele vira a sofrer.

Ao decorrer da codificação e implementação de novas funcionalidades, somos refatorando alguns trechos de códigos de funcionalidades já existentes com intuído de deixar o código limpo, organizado e por consequência minimizar a chance de introduzir novos bugs na aplicação.

* + 1. Ciclo de vida do desenvolvimento do sistema

Segundo a norma NBR ISSO/IEC 12207:1998, define ciclo de vida como ”estrutura contendo processos, actividades e tarefas envolvidas no desenvolvimento, operação e manutenção de um produtos de software, abrangendo a vida só sistema desde a definição se seus requisitos até o térmico de seu uso” (MACORATTI, 2008).

A principal função do ciclo de vida do desenvolvimento do software é indicar as fazes, actividades, entregas e responsabilidades de cada envolvido no processo de desenvolvimento só software, dentre elas, as principais fases:

De forma geral, o CVDS possui cinco fases principais em qualquer um dos modelos de desenvolvimento adotados, a seguir temos essas fases representadas em um diagrama:

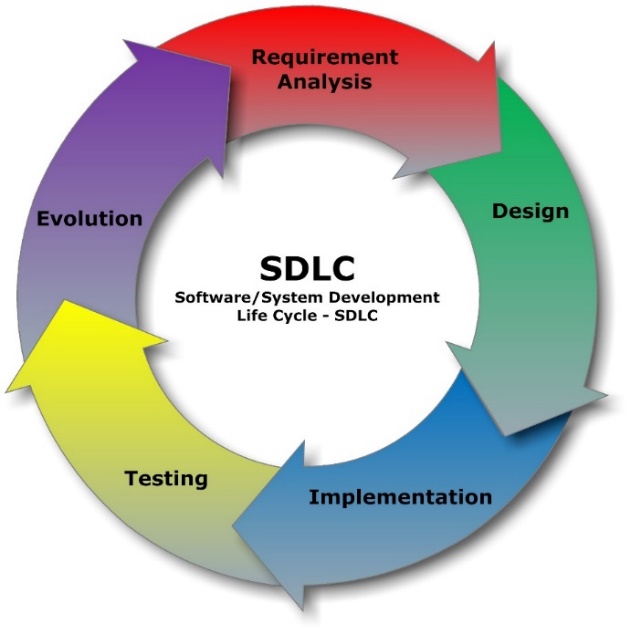


Figura 29:Ciclo de Vida de Desenvolvimento de Software

Fonte: [www.commons.wikipedia.org](http://www.commons.wikipedia.org), Agosto, 2020.

1. Fase de Requisitos: nesta fase, faz-se o levantamento dos requisitos mínimos, estuda-se a viabilidade e define-se o modelo a ser usado.
2. Fase de Projecto: envolve actividades de concepção, especificação, design da interface, prototipação, design da arquitetura.
3. Fase de Implementação: tradução para uma linguagem de programação das funcionalidades definidas durante a fases anteriores.
4. Fase de testes: realização de teste no que foi desenvolvido de acordo com os requisitos;
5. Fase de produção: implantação em produção do produto final.

# CAPÍTULO: 3- RESULTADOS



## Descrição do Projecto

Como já frisado em alguns pontos, este projecto traduz-se no desenvolvimento de um sistema de gerenciamento de munícipes, o mesmo é constituído por apenas um módulo Administrativo que é responsável por interagir e resolver os problemas dos munícipes.

Neste modulo, procuramos seguir todos os requisitos necessários e levantados para emissão dos documentos mais solicitados pelo munícipe ate a esta data, conforme referido anteriormente.

Procuramos também adotar um design de fácil aprendizagem e manipulação, pois o principal foco deste sistema é a emissão dos documentos e registo dos munícipes.

## Interfaces do Sistema

* + 1. Tela do Administrador

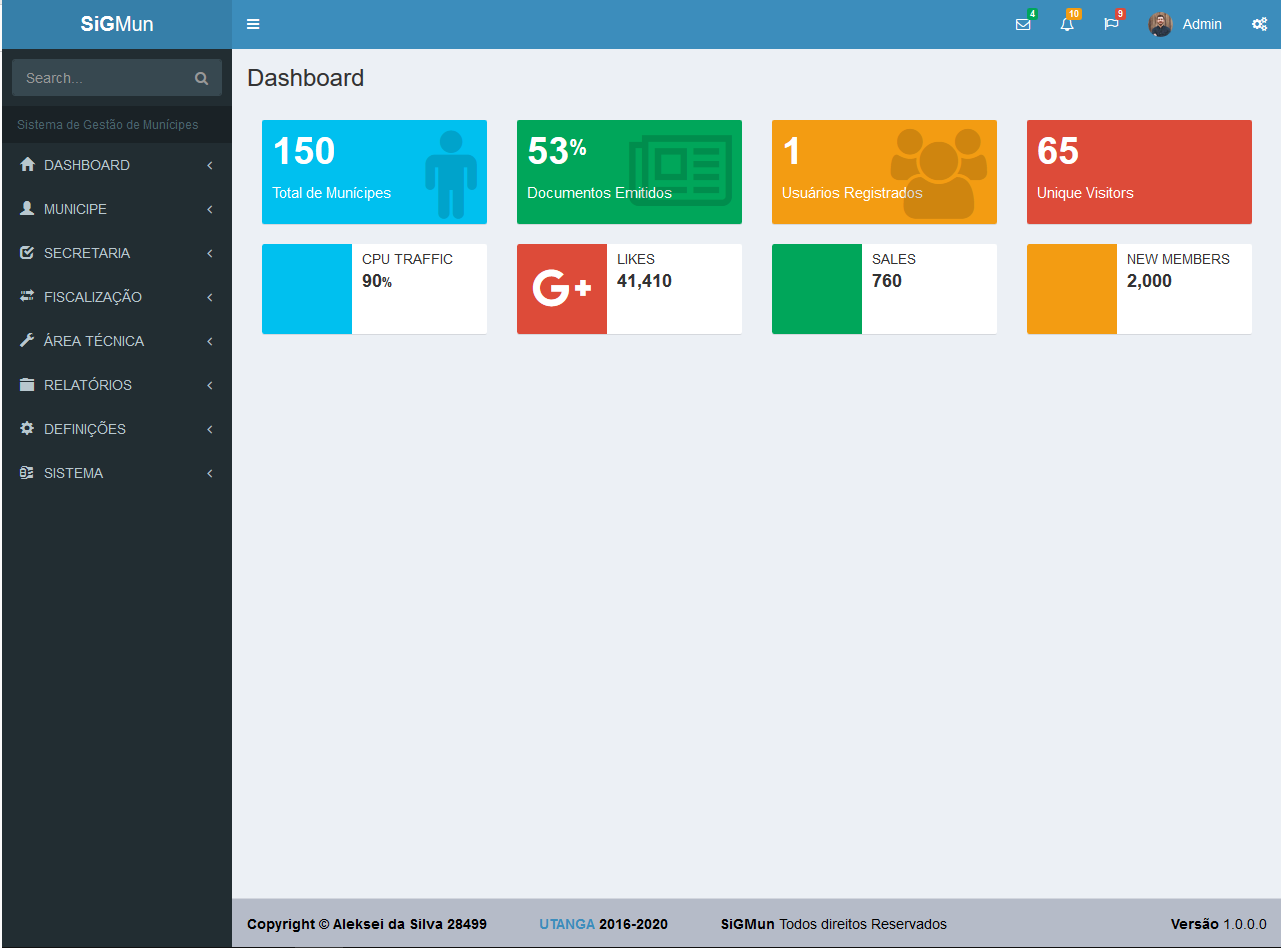


Figura 30:Tela principal do perfil Administrador

* + 1. Tela de Cadastro de Usuário

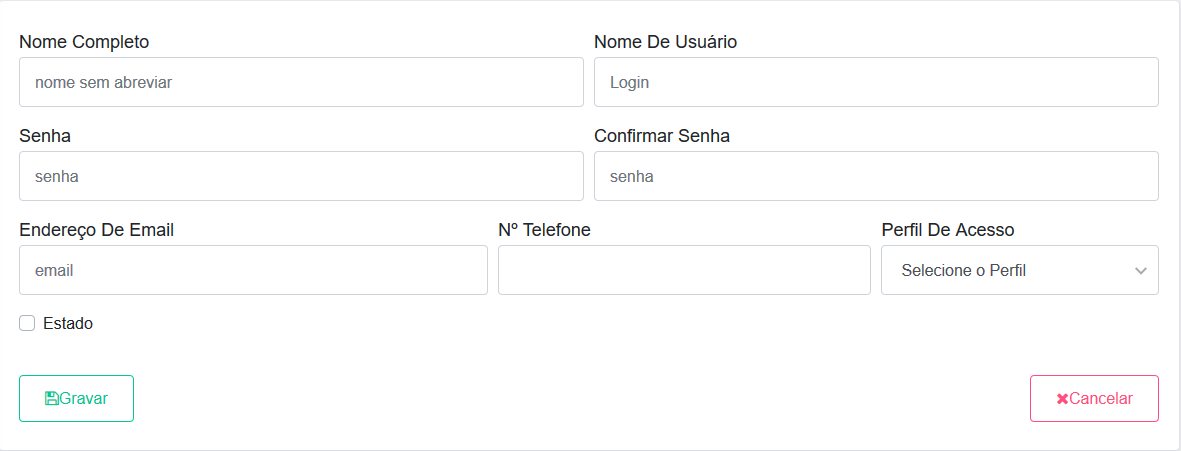


Figura 31: tela de Cadastro de Usuário.

* + 1. Tela de cadastro do Munícipe

Por Preencher

* + 1. Tela de Emissão de Documento

Por Preencher

* + 1. Tela de emissão de cartão CIM

Por Preencher

# CONCLUSÃO E RECOMENDAÇÕES

# CONCLUSÃO

Visto que o nosso projecto consta o desenvolvimento de um sistema de gerenciamento de munícipes para Administração Municipal de Malanje, em virtude dos factos mencionados, concluímos que as expectativas foram alcançadas, de modos que, atende aos requisitos ora levantados.

O projecto foi desenvolvido com boas praticas de desenvolvimento de softwares de qualidade, de tal ponto que não atendemos os requisitos de formas simples, mas providenciamos a qualidade do nosso sistema e tais ações contribuem para maior facilidade em manutenções acréscimo de funcionalidades futuras.

Mesmo com o cumprimento dos requisitos levantados, ainda assim a nossa aplicação conta com alguns desafios que poderão encarar os usuários a nível de manuseio dos meios tecnológicos envolvidos. Contudo essas dificuldades podem ser ultrapassadas pelos utilizadores mediante as formações de uso e manipulação do sistema e da tecnologia envolvida.

# RECOMENDAÇÕES

Sendo este projecto, uma aplicação que dará o auxílio no atendimento aos munícipes de forma simplificada, recomenda-se que o sistema, seja implantado em um servidor de aplicação dedicado, visto que em modo de produção, várias máquinas em simultâneo estarão solicitando informações neste servidor.

Recomenda-se também que se desenvolva novos módulos para Área Técnica, Fiscalização e Gestão de Pessoal, esses módulos irão sustentar e automatizar todas as tarefas dos demais departamentos da Administração Municipal de Malanje.

Recomendamos também a construção de um serviço baseado em Rest (Web API), para que outras entidades governamentais e privadas possam consumir e consultar os dados dos munícipes.

Tendo em conta que o trabalho foi delimitado a Administração Municipal de Malanje, recomendamos que posteriormente seja feita continuidade dos estudos para que possa ser implementado a nível do pais.

# REFERÊNCIAS BIBLIOGRAFICAS

# LIVROS

MAT. **Projecto Portal do Governo**. Conselho de Ministros. Luanda: [s.n.]. 2007. p. 22.

BENGUI, G. F. **Qualidade nos Sistema de Informação**. Universidade Tecnica de Angola. Capolo II, p. 8. 2019.

CORTÉS, M. I. **Fundamentos de Engenharia de Software**. 1º. ed. Fortaleza: Secretaria de Apoio às Tecnologias Educacionais, 2013.

RAMOS, L. ; MENNA, O. S. **Portabilização da Ferramenta de Modelagem de Banco de Dados Relacional brModelo**. UNIVERSIDADE FEDERAL DE SANTA CATARINA. Florianópolis - SC, p. 13. 2011.

RAMOS, R. A. **UML-Aula III Diagrama de Estado, Actividades, Componentes e Instalação**. CECOMP. [S.l.], p. 42. 2013.

SILVA, A. M. R. D.; VIDEIRA, C. A. E. **UML Metodologias e Ferramentas CASE**. 1º. ed. Porto-Lisboa: Centro ATLANTICO, 2001.

SOMMERVILLE, I. **Engenharia de Software**. 9º. ed. São Paulo: Pearson Education, 2011.

# SITES

ALVES, G. F. D. O. Dicas de Programação. **dicasdeprogramacao.com.br**, 2007. Disponivel em: <https://dicasdeprogramacao.com.br/o-que-e-sql/>. Acesso em: 26 Junho 2020.

ANNYSTUDIO. **AnnyStudio**, 2003. Disponivel em: <https://annystudio.com/software/colorpicker/#manual>. Acesso em: 26 Junho 2020.

BLOG Impacta. **blog Impacta**, 2017. Disponivel em: <https://www.impacta.com.br/blog/2017/12/11/voce-sabe-o-que-e-visual-studio/>. Acesso em: 26 Junho 2020.

BRASIL, A. KingHost. **KingHost**, 2019. Disponivel em: <https://king.host/wiki/artigo/o-que-e-jquery/>. Acesso em: 28 Junho 2020.

COSTA, G. tutorialwebdesign. **tutorialwebdesign**, 2014. Disponivel em: <http://www.tutorialwebdesign.com.br/o-que-e-bootstrap/>. Acesso em: 28 Junho 2020.

DESCONHECIDO. Wikipedia. **Wikipedia**, 2020. Disponivel em: <https://pt.wikipedia.org/wiki/Regras\_de\_neg%C3%B3cio>. Acesso em: 23 Julho 2020.

DEVART. Devart. **Devart**, 1997. Disponivel em: <https://www.devart.com/dbforge/sql/studio/>. Acesso em: 26 Juno 2020.

DIRCEU, R. dirceuresende. **dirceuresende**, 2013. Disponivel em: <https://www.dirceuresende.com/blog/sql-server-2019-lista-de-novidades-e-novos-recursos/>. Acesso em: 26 Junho 2020.

JONES. Jones`Blog. **Bases de Dados Acess**, 2013. Disponivel em: <https://espretuguess.wordpress.com/o-que-e-uma-base-de-dados/>. Acesso em: 26 Junho 2020.

LEITE, T. Redspark. **Redspark.io**, 2015. Disponivel em: <https://www.redspark.io/entendendo-o-entity-framework/>. Acesso em: 26 Junho 2020.

MACORATTI,. Macoratti.net. **Macoratti**, 2008. Disponivel em: <http://www.macoratti.net/17/09/net\_slcd1.htm>. Acesso em: 29 Junho 2020.

MARQUES, B. Hostinger Tutoriais. **Hostinger Tutoriais**, 2019. Disponivel em: <https://www.hostinger.com.br/tutoriais/o-que-github/>. Acesso em: 26 Junho 2020.

MICROSOFT. Microsoft. **suporte microsoft**, 2019. Disponivel em: <https://support.microsoft.com/pt-br/help/4531182/kb4531182>. Acesso em: 26 Junho 2020.

MICROSOFT CORPORATION. Microsoft. **Microsoft Docs**, 2017. Disponivel em: <https://docs.microsoft.com/pt-br/sql/relational-databases/stored-procedures/stored-procedures-database-engine?view=sql-server-ver15>. Acesso em: 26 Junho 2020.

NASCIMENTO, W. M. TreinaWeb. **TreinaWeb**, 2020. Disponivel em: <https://www.treinaweb.com.br/blog/o-que-e-o-nuget/>. Acesso em: 8 Maio 2020.

ONEDA, R. linhadecodigo. **linhadecodigo**, 2018. Disponivel em: <http://www.linhadecodigo.com.br/artigo/1602/consideracoes-iniciais-sobre-o-aspnet-mvc-framework.aspx>. Acesso em: 26 Junho 2020.

PORTAL GSTI. **Portal GSTI**. Disponivel em: <https://www.portalgsti.com.br/sql-server/sobre/>. Acesso em: 26 Junho 2020.

SOFTWARE.COM.BR. **software.com.br**. Disponivel em: <https://software.com.br/p/enterprise-architect>. Acesso em: 26 Junho 2020.

STARUML. staruml. **staruml**. Disponivel em: <https://docs.staruml.io/user-guide/readme>. Acesso em: 26 Junho 2020.

# APÊNDICE I- Dicionário de Dados

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Nome Físico:** | Tabela de Usuários | | | | | |
| **Nome Lógico:** | TB\_Usuarios | | | | | |
| **Descrição:** | Armazena os dados dos usuários | | | | | |
|  | | | | | | |
| **Descrição** | **Nome Campo** | **Tipo dados** | **Tamanho** | **Nulo** | **PK** | **FK** |
| Identificador de Usuário | Id | Int |  | Não | Sim | Não |
| Nome Completo | NomeCompleto | nvarchar | max | Sim | Não | Não |
| Nome de Login | UserName | nvarchar | 256 | Não | Não | Não |
| Email | Email | nvarchar | 256 | Sim | Não | Não |
| Confirmação de Email | EmailConfirmed | bit | 1 | Não | Não | Não |
| Palavra-passe criptografada | Passwordhash | nvarchar | max | Não | Não | Não |
| Segurança do Usuário | SecurityStamp | nvarchar | Max | Sim | Não | Não |
| Número de Telefone | PhoneNumber | nvarchar | max | Sim | Não | Não |
| Confirmação do nº Telefone | PhoneNumberConfirmed | bit | 1 | Não | Não | Não |
| Habilitação de Dois Factores | TwoFactorEnabled | bit | 1 | Não | Não | Não |
| Data de Termino de Sessão | LockoutEndDateUtc | datetime |  | Sim | Não | Não |
| Habilitar controle de termino de Sessão | LockoutEnabled | bit | 1 | Não | Não | Não |
| Contagem de Tentativas de Acesso negado | AccessFailedCoint | int |  | Não | Não | Não |
| Estado (Activo/Inativo) |  |  |  |  |  |  |
| Data de Cadastro | DataCadastro | Datetime |  | Não | Não | Não |
| Identificação do Cadastrador | IdCadastrador | int |  | Não | Não | Sim |

Tabela 11:Usuários

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Nome Físico:** | Tabela de Afirmações de usuários | | | | | |
| **Nome Lógico:** | TB\_Usuarios\_Afirmações | | | | | |
| **Descrição:** | Tabela que armazena os Dados de afirmações dos usuários | | | | | |
|  | | | | | | |
| **Descrição** | **Nome Campo** | **Tipo dados** | **Tamanho** | **Nulo** | **PK** | **FK** |
| Identificador da afirmação | Id | int |  | Não | Sim | Não |
| Identificador do usuário | UserId | int |  | Não | Não | Sim |
| Tipo de Afirmação | ClaimType | nvarchar | max | Sim | Não | Não |
| Valor da Afirmação | ClaimValue | nvarchar | max | Sim | Não | Não |

Tabela 12:Afirmações de usuários

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Nome Físico:** | Tabela de Login de Usuários | | | | | |
| **Nome Lógico:** | TB\_UsuarioLogin | | | | | |
| **Descrição:** | Tabela que armazena os dados de login dos usuários | | | | | |
|  | | | | | | |
| **Descrição** | **Nome Campo** | **Tipo dados** | **Tamanho** | **Nulo** | **PK** | **FK** |
| Provedor de Login | LoginProvider | nvarchar | 128 | Não | Sim | Não |
| Chave do Provedor | ProviderKey | nvarchar | 128 | Não | Sim | Não |
| Identificador de Usuário | UserId | int |  | Não | Sim | Não |
| Endereço de Email | Email | nvarchar | max | Sim | Não | Não |
| Senha | Senha | nvarchar | max | Sim | Não | Não |

Tabela 13:Login de Usuário

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Nome Físico:** | Tabela de Tipos de Documentos | | | | | |
| **Nome Lógico:** | TB\_TiposDocumentos | | | | | |
| **Descrição:** | Tabela que armazena os Tipos de Documentos | | | | | |
|  | | | | | | |
| **Descrição** | **Nome Campo** | **Tipo dados** | **Tamanho** | **Nulo** | **PK** | **FK** |
| Identificador do documento | DocumentoId | int |  | Não | Sim | Não |
| Descrição do Documento | DocumentoNome | varchar | max | Não | Não | Não |
| Valor do Documento | DocumentoValor | decimal | 18, 2 | Não | Não | Não |
| Identificação do Cadastrador | DocumentoUsuarioId | int |  | Não | Não | Não |
|  |  |  |  |  |  |  |

Tabela 14:Tipos de Documentos

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Nome Físico:** | Tabela de Provincias | | | | | |
| **Nome Lógico:** | TB\_Provincias | | | | | |
| **Descrição:** | Tabela que armazena os dados de Províncias | | | | | |
|  | | | | | | |
| **Descrição** | **Nome Campo** | **Tipo dados** | **Tamanho** | **Nulo** | **PK** | **FK** |
| Identificador da província | ProvinciaId | int |  | Não | Sim | Não |
| Descrição da província | ProvinciaNome | varchar | 50 | Não | Não | Não |

Tabela 15:Provincias

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Nome Físico:** | Tabela de Perfis | | | | | |
| **Nome Lógico:** | TB\_Perfil | | | | | |
| **Descrição:** | Tabela que armazena os perfis dos usuários | | | | | |
|  | | | | | | |
| **Descrição** | **Nome Campo** | **Tipo dados** | **Tamanho** | **Nulo** | **PK** | **FK** |
| Identificador do Perfil | Id | int |  | Não | Sim | Não |
| Estado do perfil | Estado | bit | 1 | Não | Não | Não |
| Data de cadastro | DataCadastro | datetime |  | Não | Não | Não |
| Descrição do Perfil | Name | nvarchar | 256 | Não | Não | Não |
| Identificação do Cadstrador | IdCadastrador | int |  | Não | Não | Sim |

Tabela 16:Perfis de usuário

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Nome Físico:** | Parente de Munícipe | | | | | |
| **Nome Lógico:** | TB\_Parente\_Municipe | | | | | |
| **Descrição:** | Tabela que armazena os dados de familiares do Munícipe | | | | | |
|  | | | | | | |
| **Descrição** | **Nome Campo** | **Tipo dados** | **Tamanho** | **Nulo** | **PK** | **FK** |
| Identificador | ParenteMunicipeNM | bigint |  | Não | Sim | Não |
| Nome do Familiar | ParenteMunicipeNomeCompleto | Varchar | 100 | Não | Não | Não |
| NM do Familiar | ParenteNM | bigint |  | Sim | Não | Não |
| Grau parentesco do Familiar | ParenteGrauParentescoId | int |  | Não | Não | Não |
| Telefone do Familiar | ParenteTelefone1 | varchar | 15 | Sim | Não | Não |
| Telefone do Familiar | Parentetelefone2 | varchar | 15 | Sim | Não | Não |
| Email do familiar | ParenteEmail | Varchar | 50 | Sim | Não | Não |
| Identificação do cadastrador | ParenteUsuarioId | Int |  | Não | Não | Não |

Tabela 17:Parente de Munícipe

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Nome Físico:** | Tabela de Municípios | | | | | |
| **Nome Lógico:** | TB\_Municipios | | | | | |
| **Descrição:** | Tabela que armazena os dados dos Municípios | | | | | |
|  | | | | | | |
| **Descrição** | **Nome Campo** | **Tipo dados** | **Tamanho** | **Nulo** | **PK** | **FK** |
| Identificador do município | MunicipioId | Int |  |  | Sim | Não |
| Descrição do Município | MunicipioNome | Varchar | 50 | Não | Sim | Não |
| Identificador da Província | MunicipioProvinciaId | Int |  | Sim | Não | Sim |

Tabela 18:Municipio

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Nome Físico:** | Tabela de Munícipes | | | | | |
| **Nome Lógico:** | TB\_Municipes | | | | | |
| **Descrição:** | Tabela que armazena os dados dos munícipes | | | | | |
|  | | | | | | |
| **Descrição** | **Nome Campo** | **Tipo dados** | **Tamanho** | **Nulo** | **PK** | **FK** |
| Identificador do Munícipe | MunicipeId | Int |  | Não | Sim | Não |
| Número de Munícipe | MunicipeNM | Bigint |  | Não | Sim | Não |
| Nome completo | MunicipeNomeCompleto | Varchar | 100 | Não | Não | Não |
| Data de Nascimento | MunicipeDataNascimento | Date |  | Não | Não | Não |
| Número do Documento Identificação | MunicipeNDocIdent | Varchar | 50 | Não | Não | Não |
| Data de Emissão do Documento | MunicipeDocDataEmissao | Date |  | Não | Não | Não |
| Data de Validade do Documento | MunicipeDocDataValidade | Date |  | Não | Não | Não |
| Número de Contribuinte | MunicipeNIF | Varchar | 50 | Não | Não | Não |
| Género | MunicipeGenero | Int |  | Não | Não | Sim |
| Estado Civil | MunicipeEstadoCivil | Int |  | Não | Não | Sim |
| Número de Telefone 1 | MunicipeTelefone1 | Varchar | 15 | Sim | Não | Não |
| Número de Telefone 2 | MunicipeTelefone2 | Varchar | 15 | Sim | Não | Não |
| Email | MunicipeEmail | Varchar | 50 | Sim | Não | Não |
| Morada | MunicipeMorada | Int |  | Não | Não | Sim |
| Estado (Ativo/Inativo) | MunicipeEstado | Bit |  | Não | Não | Não |
| Identificação do cadastrador | MunicipeUsuarioId | Int |  | Não | Não | Sim |

Tabela 19:Municipe

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Nome Físico:** | Tabela de Moradas | | | | | |
| **Nome Lógico:** | TB\_Moradas | | | | | |
| **Descrição:** | Tabela que armazena as moradas | | | | | |
|  | | | | | | |
| **Descrição** | **Nome Campo** | **Tipo dados** | **Tamanho** | **Nulo** | **PK** | **FK** |
| Identificador da Morada | MoradaMunicuipeNM | Int |  | Não | Sim | Sim |
| Rua | MoradaRua | Varchar | 200 | Não | Não | Não |
| Zona | MoradaZona | int |  | Sim | Não | Não |
| Bairro | MoradaBairroId | Int |  | Não | Não | Sim |

Tabela 20:Morada

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Nome Físico:** | Tabela de Grau Parentesco | | | | | |
| **Nome Lógico:** | TB\_GrauParentesco | | | | | |
| **Descrição:** | Tabela que armazena os graus parentesco | | | | | |
|  | | | | | | |
| **Descrição** | **Nome Campo** | **Tipo dados** | **Tamanho** | **Nulo** | **PK** | **FK** |
| Identificador | GrauParentescoId | Int |  | Não | Sim | Não |
| Descrição | GrauParentescoIdNome | Varchar | 50 | Não | Não | Não |

Tabela 21:Grau Parentesco

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Nome Físico:** | Tabela de Géneros | | | | | |
| **Nome Lógico:** | TB\_Generos | | | | | |
| **Descrição:** | Tabela de Armazena os Géneros | | | | | |
|  | | | | | | |
| **Descrição** | **Nome Campo** | **Tipo dados** | **Tamanho** | **Nulo** | **PK** | **FK** |
| Identificador | GeneroId | Int |  | Não | Sim | Não |
| Descrição | GeneroNome | Varchar | 50 | Não | Não | Não |

Tabela 22:Géneros

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Nome Físico:** | Tabela de Formas de Pagamento | | | | | |
| **Nome Lógico:** | TB\_FormasPagamento | | | | | |
| **Descrição:** | Tabela que armazena as formas de pagamentos | | | | | |
|  | | | | | | |
| **Descrição** | **Nome Campo** | **Tipo dados** | **Tamanho** | **Nulo** | **PK** | **FK** |
| Identificador | FormaPagamentoId | Int |  | Não | Sim | Não |
| Descrição | FormaPagamentoNome | Varchar | 50 | Não | Não | Não |
| Identificador do Banco | FormaPagamentoBancoId | Int |  | Não | Não | Sim |
| Identificador do cadatrador | FormaPagamentoUsuarioId | Int |  | Não | Não | Sim |
|  |  |  |  |  |  |  |

Tabela 23:Formas de Pagamento

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Nome Físico:** | Tabela de Estado Civil | | | | | |
| **Nome Lógico:** | TB\_EstadoCivil | | | | | |
| **Descrição:** | Tabela que armazena o estado civil | | | | | |
|  | | | | | | |
| **Descrição** | **Nome Campo** | **Tipo dados** | **Tamanho** | **Nulo** | **PK** | **FK** |
| Identificador | EstadoCivilId | Int |  | Não | Sim | Não |
| Descrição | EstadoCivilNome | Varchar | 20 | Não | Não | Não |

Tabela 24:Estado Civil

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Nome Físico:** | Tabela de Finalidade dos Documentos | | | | | |
| **Nome Lógico:** | TB\_DocsFinalidades | | | | | |
| **Descrição:** | Tabela que armazena as finalidades dos Documentos | | | | | |
|  | | | | | | |
| **Descrição** | **Nome Campo** | **Tipo dados** | **Tamanho** | **Nulo** | **PK** | **FK** |
| Identificado | FinalidadeId | Int |  | Não | Sim | Não |
| Descrição | FinalidadeNome | Varchar | Max | Não | Não | Não |
| Estado (Activo/Inativo) | FinalidadeEstado | Bit |  | Não | Não | Não |
| Identificação do Cadastrador | FinalidadeUsuarioId | Int |  | Não | Não | Sim |

Tabela 25:Finalidade dos Documentos

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Nome Físico:** | Tabela de Comunas | | | | | |
| **Nome Lógico:** | .TB\_Comunas | | | | | |
| **Descrição:** | Tabela que armazena os dados das comunas | | | | | |
|  | | | | | | |
| **Descrição** | **Nome Campo** | **Tipo dados** | **Tamanho** | **Nulo** | **PK** | **FK** |
| Identificador | ComunaId | Int |  | Não | Sim | Não |
| Descrição | ComunaNome | Varchar | 50 | Não | Não | Não |
| Identificador do Municipio | ComunaMunicipioId | Int |  | Não | Não | Sim |
| Identificação do cadastrador | ComunaUsuarioId | Int |  | Não | Não | Sim |

Tabela 26:Comunas

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Nome Físico:** | Tabela de Bancos | | | | | |
| **Nome Lógico:** | TB\_Bancos | | | | | |
| **Descrição:** | Tabela que armazena os dados dos bancos | | | | | |
|  | | | | | | |
| **Descrição** | **Nome Campo** | **Tipo dados** | **Tamanho** | **Nulo** | **PK** | **FK** |
| Identificador | BancoId | Int |  | Não | Sim | Não |
| Descrição | BancoNome | Varchar | 100 | Não | Não | Não |
| Sigla | BancoSigla | Varchar | 20 | Não | Não | Não |
| Estado | BancoEstado | Bit |  | Não | Não | Não |
| Identificação do Cadastrador | BancoUsuarioid | Int |  | Não | Não | Não |

Tabela 27:Bancos

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Nome Físico:** | Tabela de Bairros | | | | | |
| **Nome Lógico:** | TB\_Bairros | | | | | |
| **Descrição:** | Tabela que armazena os dados dos bairros | | | | | |
|  | | | | | | |
| **Descrição** | **Nome Campo** | **Tipo dados** | **Tamanho** | **Nulo** | **PK** | **FK** |
| Identificador | BairroId | Int |  | Não | Sim | Não |
| Descrição | BairroNome | Varchar | 50 | Não | Não | Não |
| Identificador da Comuna | BairroComunaId | Int |  | Não | Não | Não |

Tabela 28:Bairros

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Nome Físico:** | Tabela de Permissões dos usuários | | | | | |
| **Nome Lógico:** | TB\_Permissoes | | | | | |
| **Descrição:** | Tabela que armazena as permissões dos usuários | | | | | |
|  | | | | | | |
| **Descrição** | **Nome Campo** | **Tipo dados** | **Tamanho** | **Nulo** | **PK** | **FK** |
| Identificador do usuário | UserId | Int |  | Não | Sim | sim |
| Identificador do Perfil | RoleId | Int |  | Não | Sim | sim |

Tabela 29:Permissões

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Nome Físico:** | Tabela de Operações | | | | | |
| **Nome Lógico:** | .TB\_Operacao | | | | | |
| **Descrição:** | Tabela que armazena os dados das Operações | | | | | |
|  | | | | | | |
| **Descrição** | **Nome Campo** | **Tipo dados** | **Tamanho** | **Nulo** | **PK** | **FK** |
| Identificador | OperacaoId | Int |  | Não | Sim | Não |
| Identificador do munícipe | OperacaoMunicipeNM | Bigint |  | Não | Não | Sim |
| Identificador do tipo de documento | OperacaoTipoDocumentoId | Int |  | Não | Não | Sim |
| Identificador da finalidade | OperacaoDocsFinalidadeId | Int |  | Não | Não | Sim |
| Identificador da forma de pagamento | OperacaoFormaPagamentoId | Int |  | Não | Não | Sim |
| Valor total | OperacaoValorTotal | Decimal | 18,2 | Não | Não | Não |
| Data da operação | OperacaoData | Date |  | Não | Não | Não |
| Identificação do usuário | OperacaoUsuarioId | Int |  | Não | Não | Sim |

Tabela 30:Operações

# APÊNDICE II – ROTEIRO DE ENTREVISTA

Por Preencher

# APÊNDICE III- EXEMPLO DE ENTRVISTA

Por Preencher

# ANEXOS

# Anexo I

# Anexo II

# Anexo III

# Anexo IV

# Anexo V

# GLOSSÁRIO

Por Preencher

1. Kassequele, M. J. Procedimentos Armazenados. Informação passada em aula dia 29/05/2017. [↑](#footnote-ref-1)
2. Maneiro O que é Scaffolding. Mensagem respondida por stackoverflow português em 23/Março/2016. [↑](#footnote-ref-2)
3. Informação fornecida por Bengui na aula de Qualidade S. Informação, p.8 em Luanda, em 2019. [↑](#footnote-ref-3)
4. Ibid., p.35 [↑](#footnote-ref-4)