**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 uma determinada província 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 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 serie 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), apoiando as tecnologias de informação e comunicação (Tic´s), utilizando os princípios que regem as boas práticas de desenvolvimento de software, tais como: aplicação do padrão arquitectural ASP.net MVC, aplicação do boostrap para termos um website responsivo, ajustável a qualquer ecrã e injecção de dependência de modo a impedir que haja no nosso projecto um forte acoplamento dos componentes.

**Palavras Chaves:** Gerenciamento; Munícipe; Malanje; Controlo Estatístico;

#### ABSTRACT

The control of citizens in a given province 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 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 indispensable because it provides greater control over the series of documents that are issued to each citizen.

In this way, the work presented here, describes an efficient and effective alternative that allows the Municipal Administrations to have more rigor and control of the citizens who live in that municipality and the history of their documentation. This alternative translates into developing a citizen management system (SiGMun), supporting information and communication technologies (Tic´s), using the principles that govern good software development practices, such as: application of the architectural standard ASP.net MVC, application of boostrap to have a responsive website, adjustable to any screen and dependency injection in order to prevent a strong component coupling in our project.

**Keys Words:** Managment; citizen; Malanje; Statistical Control;

#### LISTA DE SIGLAS E ABREVIATURAS

|  |  |
| --- | --- |
| AJAX | -Asynchronous Javascript and Xml |
| API | -Application Programming Interface |
| BACK-END | -Parte de Retaguarda, lado do Servidor |
| CASE | -Computer -Aided Software Engineering |
| CSS | -Folha de Estilo em Cascata |
| DADI | -Definition Architecture Design Implementation |
| DB | -Data Base |
| DCL | -Data Control Language |
| DDL | -Data Definition Language |
| DHCP | -Dinamic Host Configuration Protocol |
| DLL | -Biblioteca de Vínculo Dinâmico |
| DML | -Data Manipulation Language |
| DNS | -Domain Name System |
| 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 |
| IP | -Internet Protocol |
| ISS | -Internet Information Service |
| ISSO | -International Organization Standardization |
| 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 |
| 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 |
| DLL | -Dynamic Link Library |
| XP | -Extreme Programming |
| SiGMun | -Sistema de Gerenciamento de Munícipes |
| CIM | -Cartão de Identificação do Munícipe |

#### LISTA DE FIGURAS

[Figura 12.9.14: Distribuição dos pacotes Nuget 39](#_Toc50133961)

[Figura 2.8.2.10: Modelo de Objectos do ADO.Net 41](#_Toc50133962)

#### LISTA DE TABELAS

[Tabela 2.3.4: Regras de Negócios 21](#_Toc50000508)

#### ÍNDICE

[INTRODUÇÃO 13](#_Toc50420219)

[DEFINIÇÃO DO PROBLEMA 13](#_Toc50420220)

[JUSTIFICATIVA 14](#_Toc50420221)

[OBJECTIVOS 14](#_Toc50420222)

[OBJECTIVOS GERAIS 14](#_Toc50420223)

[OBJECTIVOS ESPECÍFICOS 14](#_Toc50420224)

[DELIMITAÇÃO 15](#_Toc50420225)

[ESTRUTURA DO TRABALHO 15](#_Toc50420226)

[CAPÍTULO: 1- FUNDAMENTAÇÃO TEÓRICA 15](#_Toc50420227)

[CAPÍTULO: 2- METODOLOGIA 16](#_Toc50420228)

[2.1. Tipo de Pesquisa 16](#_Toc50420229)

[2.2. Campo de Estudo 16](#_Toc50420230)

[2.3. Levantamento de Requisitos 17](#_Toc50420231)

[2.3.1. Analise de requisitos 17](#_Toc50420232)

[2.3.2. Requisitos Funcionais 17](#_Toc50420233)

[2.3.3. Requisitos Não Funcionais 18](#_Toc50420234)

[2.3.4. Regras de Negócio 19](#_Toc50420235)

[2.4. Análise do Sistema 21](#_Toc50420236)

[2.4.1. Diagramas do Sistema 21](#_Toc50420237)

[2.4.2. Diagrama de caso de Uso 21](#_Toc50420238)

[2.4.3. Descrição dos Principais Casos de Uso 22](#_Toc50420239)

[2.4.4. Diagrama Entidade-Relacionamento 28](#_Toc50420240)

[2.5. Diagrama Lógico 29](#_Toc50420241)

[2.5.1. Diagrama de Classe 30](#_Toc50420242)

[2.5.2. Diagrama de Sequência 31](#_Toc50420243)

[2.5.3. Diagrama de Instalação 34](#_Toc50420244)

[2.6. Ferramentas de Desenvolvimento 35](#_Toc50420245)

[2.7. Arquitectura lógica do Sistema 37](#_Toc50420246)

[2.7.1. Camada de Apresentação 37](#_Toc50420247)

[2.7.2. Camada de Serviço **Erro! Marcador não definido.**](#_Toc50420248)

[2.7.3. Camada de Aplicação **Erro! Marcador não definido.**](#_Toc50420249)

[2.7.4. Camada de Domínio **Erro! Marcador não definido.**](#_Toc50420250)

[2.7.5. Camada de Infra-estrutura 37](#_Toc50420251)

[2.8. Desenvolvimento de base de Dados 37](#_Toc50420252)

[2.8.1. Criação de base de dados 37](#_Toc50420253)

[2.9. Tecnologias Implementadas no Projecto 38](#_Toc50420254)

[2.9.1. Base de Dados 38](#_Toc50420255)

[2.9.2. Entity Framework 38](#_Toc50420256)

[2.9.3. Procedimentos armazenados 38](#_Toc50420257)

[2.9.4. Sequências 38](#_Toc50420258)

[2.9.5. Paralelismo 38](#_Toc50420259)

[2.9.6. Programação 38](#_Toc50420260)

[2.9.7. Asp.NET 4.8 38](#_Toc50420261)

[2.9.8. Booststrap 38](#_Toc50420262)

[2.9.9. Jquery 38](#_Toc50420263)

[2.9.10. Inversão de Controle 38](#_Toc50420264)

[2.9.11. Injecção de Dependência 38](#_Toc50420265)

[2.9.12. Padrão de repositório 38](#_Toc50420266)

[2.9.13. Scaffolding 38](#_Toc50420267)

[2.9.14. Nuget 39](#_Toc50420268)

[2.9.15. ASP.NET Identity 2.1 39](#_Toc50420269)

[2.9.16. Ado.NET 40](#_Toc50420270)

[2.9.17. Json 41](#_Toc50420271)

[2.10. Fase de Testes 42](#_Toc50420272)

[2.10.1. Teste de Unidade 42](#_Toc50420273)

[2.10.2. Teste de Integração 42](#_Toc50420274)

[2.10.3. Teste de Aceitação 42](#_Toc50420275)

[CAPÍTULO: 3- RESULTADOS 42](#_Toc50420276)

[3.1. Descrição do Projecto 42](#_Toc50420279)

[3.2. Interfaces do Sistema 42](#_Toc50420280)

[3.2.1. Tela do Administrador 42](#_Toc50420281)

[3.2.2. Tela do Atendente 42](#_Toc50420282)

[CONCLUSÃO E RECOMENDAÇÕES 43](#_Toc50420283)

[CONCLUSÃO 43](#_Toc50420284)

[RECOMENDAÇÕES 43](#_Toc50420285)

[REFERÊNCIAS BIBLIOGRAFICAS 44](#_Toc50420286)

[APÊNDICE 45](#_Toc50420287)

[ANEXOS 46](#_Toc50420288)

[GLOSSÁRIO 47](#_Toc50420289)

# 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 passou a ser divido 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, de tal modo que a finalidade deste software 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.

## DEFINIÇÃO DO PROBLEMA

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.

## 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 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 usadas 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.
* Desenvolver o Sistema utilizando as ferramentas Visual Studio e SQL Server.
* 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 é o capítulo reservado para abordagens mais teórica do projecto, apresentamos aqui as formas e procedimentos existentes actualmente para dar 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: 1- FUNDAMENTAÇÃO TEÓRICA

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.

# 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 se dá principalmente pro 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 empresa escolhida que teve a duração de 2 dias conforme consta no Apêndice A.

O autor entrou em contacto com a empresa e os respectivos funcionários

## Campo de Estudo

O campo de estudo desta pesquisa é a Administração Municipal de Malanje, que é um órgão desconcentrado da Administração do estado na província, que visa assegurar a realização das funções do estado a nível do município (MAT, 2007). A mesma está localizada no bairro azul, centro da cidade, rua …… e comporta a seguinte estrutura orgânica:

1-Administrador Municipal

1-Administrador Municipal Adjunto

3-Chefes de sessão

25-funcionários normais

Perfazendo um total de 30 funcionários. A mesma comporta uma estrutura física de oito gabinetes, um parque de estacionamento e uma pequena zona verde.

## 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. Analise 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 que os Administradores cadastrem e actualizem dados de perfis dos usuários. |
| RF2 | O sistema deve permitir que o Administrador cadastre e actualize dados dos usuários. |
| RF3 | O sistema deve permitir que o Administrador cadastre e actualize os dados das contas bancarias que serão efectuados os pagamentos. |
| RF4 | O sistema deve permitir que o Administrador cadastre e actualize os formas de pagamentos aceites. |
| RF5 | O sistema deve permitir que o Administrador cadastre e actualize os tipos de Documentos que serão emitidos. |
| RF6 | O sistema deve permitir que o Administrador cadastre e actualize os dados de finalidade dos Documentos que serão emitidos. |
| RF7 | O sistema deve permitir que o usuário cadastre e actualize dados dos Munícipes |
| RF8 | O sistema deve permitir que o usuário cadastre e actualize informações de familiares do Munícipe. |
| RF9 | O sistema deve permitir que o usuário cadastre e actualize os dados da morada dos Munícipes. |
| RF10 | O sistema deve permitir que o usuário cadastre e actualize os dados académicos dos Munícipes. |
| RF11 | O sistema deve permitir que o usuário faça emissão de documentos. |
| RF12 | O sistema deve permitir que o usuário efectue o pagamento. |
| RF13 | O sistema deve permitir que o usuário emita o CIM (Cartão de Identificação do Munícipe). |
| RF14 | O sistema deve permitir que o usuário efectue diversas consultas. |
| RF15 | O sistema deve permitir que o usuário emita, visualize ou imprima relatórios. |
| RF16 | O sistema deve permitir que o usuário reemita documentos, |
|  |  |

Tabela 2.3.2:Requisitos Funcionais

Fonte: Produzido pelo próprio autor.

* + 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 possuirá uma interface amigável com as seguintes características:  Idioma do sistema em Português, botões com ícones que representam a funcionalidade, ambiente de trabalho atractivo que permite que os usuários se adaptem facilmente.  Conta também com uma selecção de cores que não cansam a visão mesmo enxergando durante muito tempo. |
| RNF2 | Segurança | O sistema possuirá um mecanismo de autenticação por login e Autorização por permissões de acesso.  O sistema possuirá um algoritmo de criptografia AES de 128 bits. |
| RNF3 | Funcionalidade | O sistema executa todas as solicitações feitas pelos usuários desde que estas funcionalidades tenham sido implementadas. |
| RNF4 | Confiabilidade | O sistema é capaz de manter o seu normal funcionamento mesmo em situações de grandes requisições |
| RNF | Manutenibilidade |  |
| RNF | Escalabilidade |  |
| RNF | Portabilidade |  |

Tabela 2.3.3:Requisitos Não Funcionais

Fonte: Produzido pelo próprio autor.

* + 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 | -A entidade Perfil deve ser preenchido com o perfil de Administrador na implementação do sistema.  -O sistema não deve permitir o cadastro de perfis de usuário com campos obrigatório vazio. Campo obrigatório: Descrição do Perfil. |
| RN2 | -A entidade Usuários deve ser preenchido com um usuário que tenha o perfil de Administrador na implementação do sistema.  -O sistema não deve permitir o cadastro de usuário com campos obrigatórios vazio. Campos obrigatórios: Email, Senha, Perfil de Acesso. |
| RN3 | -O sistema não deve permitir o cadastro de Bancos com campos obrigatório vazio. Campos obrigatórios: Descrição e Sigla. |
| RN4 | -Para cadastrar os Tipos de Pagamentos, deve-se preencher os campos obrigatórios. Campos obrigatórios: Descrição do tipo de pagamento e seleccionar o banco. |
| RN5 | -Para cadastrar os Tipos de Documentos e Finalidade dos Documentos deve-se preencher os campos obrigatórios. Campos obrigatórios: Descrição do tipo de Documento, seleccionar o banco, descrição da finalidade e seleccionar o tipo de documento. |
| RN6 | -Para cadastrar munícipe deve-se preencher os campos obrigatórios. Campos obrigatórios: nome completo, data de nascimento, tipo de documento de identificação, número do documento de identificação, data de validade, município e província de nascimento, seleccionar o bairro.  -O sistema não deve permitir o cadastro com data de validade inferior ou igual a data actual. |
| RN7 | -Para cadastrar morada do munícipe deve-se preencher o campo obrigatório. Campo obrigatório: Bairro. |
| RN8 | -Para a emissão de documentos deve-se preencher os campos obrigatórios. Campos obrigatórios: buscar o número de Munícipe, seleccionar o tipo de documento, seleccionar a finalidade e a validade. |
| RN9 | -Para escolher a forma de pagamento deve-se emitir um documento e preencher o campo obrigatório. Campo obrigatório: seleccionar o banco e seleccionar uma das opções. Opções: Depósito, TPA ou Transferência. |
| RN10 | -Para efectuar o pagamento deve-se concluir o procedimento de escolha da forma de pagamento e a posterior, preencher os campos obrigatórios. Campos Obrigatórios: número de operação, valor e a data. |
| RN11 | -Para visualizar ou imprimir |
| RN12 | -Para emitir o cartão de identificação do munícipe, deve-se preencher o campo obrigatório. Campo obrigatório: Número de identificação do munícipe. |
| RN13 | -Para emitir relatório, visualizar ou imprimir deve-se preencher o campo obrigatório: escolher o período do relatório. |
| RN14 |  |

Tabela .3.4:Regras de Negócios

Fonte: Produzido pelo próprio autor.

## Análise do Sistema

Os A qualidade de um produto de software, aumenta à medida que se aplica a analise 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

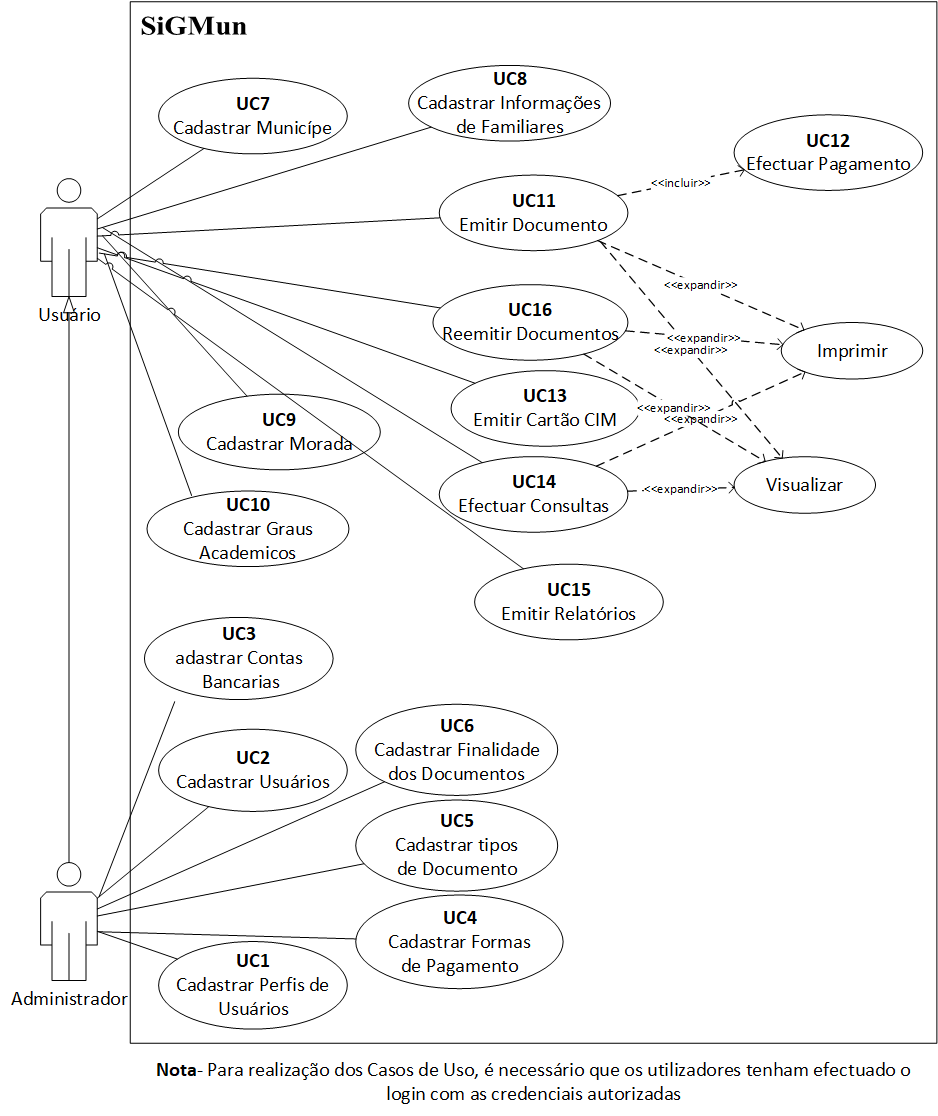
* + 1. Diagramas do Sistema

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).

* + 1. 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.



* + 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.

|  |  |
| --- | --- |
| Cadastrar Usuário | |
| Descrição: | Este caso de uso permite cadastrar 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: | Usuário cadastrado com Sucesso |
| Actores: | Administrador |
| Fluxo Principal: | 1. O usuário acede o menu Sistema. 2. Clicar no menu usuários. 3. O sistema apresenta a tela de usuários cadastrados. 4. O usuário deve clicar em 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: | O sistema apresenta uma mensagem de erro e retorna a mesma pagina para que o usuário possa corrigi-lo. |

Esta tabela, descreve o caso de uso cadastrar munícipe, pois este elemento é o centro da nossa aplicação.

|  |  |
| --- | --- |
| Cadastrar Munícipe | |
| Descrição: | Este caso de uso permite cadastrar os munícipes no sistema. |
| Pré-Condição: | 1. Estar logado no sistema. 2. Ter permissão para realização do caso de uso. |
| Pós-Condição: | Munícipe cadastrado com Sucesso |
| Actores: | Administrador e Usuário |
| Fluxo Principal: | 1. O usuário acede o menu Munícipe. 2. Clicar no menu Listar. 3. O sistema apresenta a tela de Munícipes cadastrados. 4. O usuário 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 usuário deve clicar no botão guardar. |
| Fluxo Alternativo: | O sistema apresenta uma mensagem de erro e retorna a mesma página para que o usuário possa corrigi-lo. |

|  |  |
| --- | --- |
| Cadastrar informações de familiares | |
| Descrição: | Este caso de uso permite cadastrar as informações de familiares dos munícipes. |
| Pré-Condição: | 1. Estar logado no sistema. 2. Ter permissão para realização do caso de uso. |
| Pós-Condição: | Informações cadastradas com sucesso |
| Actores: | Administrador e usuários. |
| Fluxo Principal: | 1. O usuário acede o menu munícipe. 2. Clicar no menu listar. 3. O sistema apresenta a tela de munícipes cadastrados. 4. O usuário deve clicar no ícone . 5. O sistema apresentará a tela de cadastro de informações de familiares. O usuário deve clicar no botão Adicionar e seleccionar o grau parentesco, a seguir deve preencher os campos. 6. Usuário deve clicar no botão guardar. |
| Fluxo Alternativo: | O sistema apresenta uma mensagem de erro e retorna a mesma página para que o usuário possa corrigi-lo. |

|  |  |
| --- | --- |
| Cadastrar Morada | |
| Descrição: | Este caso de uso permite que os usuários cadastrem a morada dos munícipes. |
| Pré-Condição: | 1. Estar logado no sistema. 2. Ter permissão para realização do caso de uso. |
| Pós-Condição: | Morada cadastrada com Sucesso |
| Actores: | Administrador e usuários |
| Fluxo Principal: | 1. O usuário acede o menu munícipe. 2. Clicar no menu listar. 3. O sistema apresenta a tela de munícipes cadastrados. 4. O usuário deve clicar no ícone . 5. O sistema apresentará a tela de cadastro de morada. O usuário seleccionar a o município, a comuna e preencher os campos rua, zona e número da casa. 6. Usuário deve clicar no botão guardar. |
| Fluxo Alternativo: | O sistema apresenta uma mensagem de erro e retorna a mesma página para que o usuário possa corrigi-lo. |

|  |  |
| --- | --- |
| Cadastrar Grau Académico | |
| Descrição: | Este caso de uso permite que os usuários cadastrem o nível académico dos munícipes. |
| Pré-Condição: | 1. Estar logado no sistema. 2. Ter permissão para realização do caso de uso. |
| Pós-Condição: | Grau académico cadastrado com sucesso |
| Actores: | Administrador e usuários |
| Fluxo Principal: | 1. O usuário acede o menu munícipe. 2. Clicar no menu listar. 3. O sistema apresenta a tela de munícipes cadastrados. 4. O usuário deve clicar no ícone . 5. O sistema apresentará a tela de cadastro de graus académicos. O usuário deve clicar em adicionar, seleccionar o nível académico e seleccionar o ano. 6. Usuário deve clicar no botão guardar. |
| Fluxo Alternativo: | O sistema apresenta uma mensagem de erro e retorna a mesma página para que o usuário possa corrigi-lo. |

|  |  |
| --- | --- |
| Emitir Documento | |
| Descrição: | Este caso de uso permite cadastrar 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: | Usuário cadastrado com Sucesso |
| Actores: | Administrador |
| Fluxo Principal: | 1. O usuário acede o menu Sistema. 2. Clicar no menu usuários. 3. O sistema apresenta a tela de usuários cadastrados. 4. O usuário deve clicar em 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: | O sistema apresenta uma mensagem de erro e retorna a mesma pagina para que o usuário possa corrigi-lo. |

|  |  |
| --- | --- |
| Cadastrar Usuário | |
| Descrição: | Este caso de uso permite cadastrar 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: | Usuário cadastrado com Sucesso |
| Actores: | Administrador |
| Fluxo Principal: | 1. O usuário acede o menu Sistema. 2. Clicar no menu usuários. 3. O sistema apresenta a tela de usuários cadastrados. 4. O usuário deve clicar em 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: | O sistema apresenta uma mensagem de erro e retorna a mesma pagina para que o usuário possa corrigi-lo. |

* + 1. Diagrama Entidade-Relacionamento

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).



Figura 2.4.4:Diagrama Entidade-Relacionamento

Fonte: Produzido pelo próprio autor.

## Diagrama Lógico

“Um diagrama logico é um 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)

* + 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:



* + 1. Diagrama de Sequência

Silva; Videira (2001), afirma que ilustra um 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.













* + 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.



## 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 3 camadas, aplicando alguns conceitos básicos e simplificado e na qual podemos implementar as camadas conforme mostra a figura abaixo:



* + 1. Camada de Apresentação

Os A qualidade de um produto de software, aumenta à medida que se aplica a analise 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

* + 1. Camada de Infra-estrutura

Esta camada actua como biblioteca de apoio a outra camada e implementa persistência dos dados em banco de dados escolhido para o desenvolvimento da aplição.

## Desenvolvimento de base de Dados

* + 1. Criação de base de dados

## Tecnologias Implementadas no Projecto

* + 1. Base de Dados
    2. Entity Framework
    3. Procedimentos armazenados
    4. Sequências
    5. Paralelismo
    6. Programação
    7. Asp.NET 4.8
    8. Booststrap
    9. Jquery
    10. Inversão de Controle
    11. Injecção de Dependência
    12. Padrão de repositório
    13. 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)[[1]](#footnote-1).

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

* + 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 2.9.14:** Distribuição dos pacotes Nuget

**Fonte:** <https://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 2.8..10: Modelo de Objectos do ADO.Net

Fonte: <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

* + 1. Teste de Unidade
    2. Teste de Integração
    3. Teste de Aceitação

# CAPÍTULO: 3- RESULTADOS



## Descrição do Projecto

## Interfaces do Sistema

* + 1. Tela do Administrador
    2. Tela do Atendente

# CONCLUSÃO E RECOMENDAÇÕES

# CONCLUSÃO

# RECOMENDAÇÕES

# REFERÊNCIAS BIBLIOGRAFICAS

# APÊNDICE

# ANEXOS

# GLOSSÁRIO

1. Maneiro O que é Scaffolding. Mensagem respondida por stackoverflow português em 23.Março.2016. [↑](#footnote-ref-1)