



Instituto Politécnico Industrial de Luanda

Área de Formação: Informática

Curso: Técnico de Gestão de Sistemas Informáticos

Programação web

**IMPLEMENTAÇÃO DE UMA PÁGINA DE:
DIVUGAÇÃO E RECUPERAÇÃO DE DOCUMENTOS PERDIDOS**

13º classe

Turma:IG13A

Luanda Setembro de 2020/2021

Instituto Politécnico Industrial de Luanda

Área de Formação: Informática

Curso: Técnico de Gestão de Sistemas Informáticos

Programação Web

**IMPLEMENTAÇÃO DE UMA PÁGINA:
RECUPERAÇÃO DE DOCUMENTOS PERDIDOS**

Trabalho de fim de curso apresentado no Instituto Médio Politécnico de Luanda Área de Formação de Sistemas Informáticos orientado pelo prof.*Eng: Nelson Mario Tito para obtenção de Técnico Médio

Prof.Eng.* Nelson Mario Tito

Orientador

FOLHA DE APROVAÇÃO

Câmia Francisco Miguel Diogo

Gabriela Augusto Camoço

IMPLEMENTAÇÃO DE WEB-SITE DE DIVUGAÇÃO E RECUPERAÇÃO DE DOCUMENTOS PERDIDOS

Data de Aprovação ____/____/____

Mesa de Jurados

Júri

1º Vogal

2º Vogal

3º Vogal

Avaliação final: _____

Orientadores

NELSON TITO MÁRIO

Orientador Regente

NDINGA MACAIA HERMES

Co-Orientador

INTEGRANTES

Nº 03	Câmia Francisco Miguel Diogo
Nº 14	Gabriela Augusto Camoço

Este trabalho é dedicado aos nossos pais,
professores, amigos e colegas que sempre nos
deram apoio e incentivo, permitindo assim,
a sua realização.

Agradecimento

É de agradecer a Deus todo poderoso que nós ajudou a chegar aqui visto que muita gente não conseguiu chegar aqui, agradecemos também pela saúde que nós proporcionou durante este tempo de elaboração do nosso projecto. Agradecemos aos nossos pais pelo apoio que nós deram durante a elaboração do nosso projecto, agracer os nossos amigos e colegas pela ajuda que nós deram.

Um agradecimento Especial

Ao nosso amigo Ndoma Domingos pela ajuda que nós deu durante a elaboração do nosso projecto. A ti agradecemos do fundo do nossos corações.

LISTA DE FIGURAS

Figura 1-HTML5 26

Figura 2-CSS 28

Figura 3-JS..... 29

Figura 4-PHP 30

Figura 5-Documentos 46

LISTA DE TABELAS

<i>Tabela 1-Plataforma . net</i>	33
<i>Tabela 2-Programação em camada</i>	36
<i>Tabela 3-Padrão em camada</i>	37
<i>Tabela 4-Requisitos funcionais</i>	48
<i>Tabela 5-Requisitos não funcionais</i>	49
<i>Tabela 6-Identidade dos atores</i>	50

LISTA DE TABELAS

<i>Tabela 1-Plataforma . net</i>	33
<i>Tabela 2-Programação em camada</i>	36
<i>Tabela 3-Padrão em camada</i>	37
<i>Tabela 4-Requisitos funcionais</i>	48
<i>Tabela 5-Requisitos não funcionais</i>	49
<i>Tabela 6-Identidade dos atores</i>	50

LISTA DE ABREVIATURAS, SIGLAS

HTML (Hypertext Markup Language)

PHP (Personal Home Page)

CSS (Cascading Style Sheets)

SGBD (Sistemas Gerenciadores de Banco de Dados)

DDL (Data Definition Language)

DML (Data Manipulation Language)

DCL (Data Control Language)

WWW (World Wide Web)

RDP (Recuperação de Documentos Perdidos)

CLR (Commom Language Runtime- ambiente de execução independente da linguagem)

FCL (Framework Class Library-conjunto de biblioteca unificadas)

JIT (Just In Time)

CIL(Commom Intermediate Language)

MVC (Model View Controller)

PDAS (Personal Digital Assistant)

ADO (Active Data Object)

DLL (Dynamic Link Library)

BCL (Base Class Library-Biblioteca de Classe Base)

Sumário

IMPLEMENTAÇÃO DE WEB-SITE DE RECUPERAÇÃO DE DOCUMENTOS PERDIDOS	3
Mesa de Jurados.....	3
LISTA DE FIGURAS.....	8
LISTA DE ABREVIATURAS, SIGLAS.....	11
INTRODUÇÃO	15
PROBLEMAS.....	16
OBJECTIVOS	17
OBJECTIVO GERAL	17
OBJECTIVOS ESPECIFICOS.....	17
METODOLOGIA.....	18
1-CONCEITO SOBRE PÁGINA WEB.....	21
1.1 Os 3 Pilares Básicos da Programação WEB.....	21
1.2 Como se cria uma página web?	21
1.3 OS NAVEGADORES.....	22
1.4 SERVIDOR WEB	22
1.1-CONCEITO SOBRE HTML.....	23
1.1.1 HTML Semântico	23
1.3-Conceito sobre Java Script	28
1.3.1 Característica da linguagem Java Script	28
1.3.2 Console do navegador	29
1.4 PHP	29
1.4.1 Interoperabilidade.....	30
1.5 Banco de Dados	30
1.5.1 Conceito de Sistema de Gerenciamento de Banco de Dados	31
1.5.2 SQL.....	31
2.1-Programação ou arquitetura em camada	36
Propósito de Cada Camada	36
UI	36

Negócio.....	36
Dados.....	36
Comunicação	36
Vantagens e Desvantagem	36
Vantagens.....	36
Desvantagem.....	37
2.2-Padrão Arquitetura em Camadas.....	37
2.3-Arquitetura ou modelo em 3 camadas	37
Camada de apresentação	37
Camada de negócio	37
Camada de Dados	38
Arquitetura em 3 camadas.....	38
Camada de Comportamento:.....	39
Model View Controller (Modelo-Visão-Controlé-MVC)	39
Os componentes do modelo MVC	40
Camada de modelo ou da lógica da aplicação (Model)	40
Camada de apresentação ou visualização (View)	40
Camada de controle ou controlador (Controller)	40
Interações dos componentes	41
Vantagens e Desvantagens do modelo MVC	41
Desvantagens do modelo MVC:	42
Justificativa	42
Plataformas de desenvolvimento.....	43
Java	43
PHP	43
Capitulo#3	44
Implementação do Projecto	44
3- Implementação de um sistema de Divulgação e Recuperação de Documentos Perdidos	45
3.1- Descrição do Sistema:	45

Problema	45
Proposta de solução	45
Como queremos que o sistema funcione?	45
3.2-O que é um documento?	46
Classificação dos documentos.....	46
3.3-Quanto ao seu valor	46
3.4- REQUISITOS	48
3.4.1- REQUISITOS FUNCIONAIS.....	48
3.4.2- REQUISITOS NÃO FUNCIONAIS	49
3.5-Identificação dos atores	49
Cidadão.....	49
3.5.1-Caso de Uso.....	50
3.6-IDENTIFICAÇÃO DOS CASOS DE USO POR ATORES	50
Modelo entidade relacionamento	52

INTRODUÇÃO

No nosso país notamos uma grande debilidades no que tem haver com a recuperação de documentos.

Um documento é qualquer registro de informação independentemente do seu formato ou suporte utilizado para registrá-las.

Quanto a sua origem os documentos podem ser divididos em publico ou privado:

Públicos: são emitidos e/ou recebido por um órgão governamental na gestão de suas atividades ou mediante procuração de autoridade pública. Por exemplo, escrituras públicas de compra e venda emitida por um notário.

Privado: são emitidos por um particular (pessoa física), ou por autoridade pública fora de suas funções, atribuição, ou competência.

O presente projecto visa implementar um sistema de recuperação de documento usando plataformas móveis para o melhoramento dos serviços de recuperar os seus documentos a nível do país.

No primeiro capítulo vamos abordar sobre os conceitos teóricos e falaremos sobre as tecnologias utilizadas para o desenvolvimento do nosso sistema.

No segundo capítulo vamos falar sobre a arquitetura em camada e sua funcionalidade, suas vantagens e desvantagens, sua arquitetura em 3 camadas, e o modelo em 4 camadas.

No último capítulo falaremos sobre a implementação do nosso sistema, sua descrição, problema ,proposta de solução para a resolução do problema, análise de requisitos, ilustrações de algumas tabelas e diagrama de entidade relacionamento e a descrição das tabelas.

PROBLEMAS

No País, nota-se claramente a existência de muitos documentos perdidos, o problema está no tempo em que as pessoas perdem , a espera da segunda via dos seus documentos.

Consoante os nossos estudos podemos notar uma grande debilidade no que tange a recuperação de documentos perdidos ,então surgiu a ideia de criarmos este site para ajudar o nosso povo ,na recuperação do mesmo.

JUSTIFICATIVA

Diariamente, várias pessoas perdem os seus documentos em diferentes pontos da cidade. Para a resolução desse problema, decidimos criar um sistema eficiente, e seguro, para fazer a recuperação de documento. Isso eliminará aglomeração nas identificações, facilitará a recuperação do documento.

Ele será simples a nível visual, e complexo a nível funcional, facilitará ao máximo o trabalho de todos os cidadão.

OBJECTIVOS

OBJECTIVO GERAL

É a divulgação e repuração de documentos perdidos.

OBJECTIVOS ESPECIFICOS

- Gerir documentos;
- Eliminar usuário/cidadão;
- Gerir usuário/cidadão;
- Listar documentos;
- Postar documentos.

METODOLOGIA

Neste subtema serão discutidos todos aspectos concernentes a metodologia científicas, a criação da proposta de solução, bem como os recursos utilizados para o desenvolvimento do mesmo.

Modelo de Pesquisa

Foi feita uma pesquisa baseada nas possibilidades e analisada pelo referencial teórico, nomeadamente sobre RDP (Recuperação de Documentos Perdidos) e sistema de gestão de RDP, conseguimos chegar a conclusão de que á uma necessidade de ter um sistema de recuperação de documentos perdidos para o nosso país.

Tipo de Pesquisa

Quanto ao tipo de pesquisa escolhemos uma pesquisa do tipo qualitativa porque os estudos feitos no desenvolvimento do projecto nós permitiu colocar o raciocino lógico, chegar a conclusão da utilização da ferramenta descrita.

Pesquisa Bibliográfica

Foi usada o tipo de pesquisa bibliográfica, porque o estado da arte de pesquisa foi sustentada com base em livros, tutorias, pdf, relacionada ao assunto de RDP.

Pesquisa Documental

Nesta pesquisa documental foi útil na aquisição de informação que estão relacionada as características RDP e de normais para o uso da implementação do sistema. Foram utilizados na maior parte documentos electrónicos.

Técnicas de extração de dados

Foi utilizada um método empírico para extração de dados relevantes para implementação do nosso projecto, uma vez que os dados foram passados por informações advinhas de docentes.

ESTRUTURA DO RELATÓRIO

O relatório está composto pelas seguintes partes:

a) Elementos pré-textuais:

- Capa;
- Contra-capas;
- Dedicatória;
- Agradecimento;
- Sumário;
- Lista de figuras;
- Lista de abreviaturas;
- Lista de tabelas;

b) Elementos textuais:

- Introdução;
- Problemática;
- Justificativa;
- Objectivos geral e específicos;
- Metodologias;
- Estrutura do relatório;
- Capítulo 1
- Capítulo 2
- Capítulo 3
- Conclusão

c) Elementos pós-textuais:

- Referências;
- Anexos;
- Apêndice;

Capítulo#1

FUNDAMENTAÇÃO TEORICA

1-CONCEITO SOBRE PÁGINA WEB

Uma página web é um elemento específico de uma apresentação da web que está contida em uma estrutura.

A web é um conjunto de documentos acessíveis por meio da internet. Esses documentos (ou páginas Web) contêm uma tecnologia chamada hipertexto. Onde você poderá percorrer partes do documento e outros documentos através de pontes chamadas de links.

Poderia descrever a World Wide Web (www) como um sistema de informação em hipertexto, gráfico, distribuído, independente de plataforma, dinâmico, interativo e global, utilizado na internet.

O hipertexto permite que você leia um texto e navegue por ele e por informações visuais de forma não linear, com base nas informações que deseja obter em seguida. A Web fornece recursos de imagens e animação fantásticos.

Desenvolvimento Web, é um termo utilizado para descrever o desenvolvimento de Sites, na internet ou numa intranet. O profissional que trabalha desenvolvendo Web sites, pode ser um Web designer (desenvolvedor do layout), ou um Web developer (desenvolvedor de sistemas).

1.1 Os 3 Pilares Básicos da Programação WEB

A WWW tem 3 Linguagens padrão que se complementam, e acabam sendo os 3 pilares básicos da programação WEB. O HTML em conjunto com o CSS e JAVASCRIPT, formam as pedras principais para a World Wide Web. Todos os navegadores atuais recebem documentos em HTML que são processados, renderizados, e apresentados como conteúdo online.

1.2 Como se cria uma página web?

Uma página Web é composta de textos e comandos especiais (tags) de HTML, um acrônimo de Hypertext Markup Language. Essa linguagem é bastante simples e tem como finalidade básica formatar o texto exibido e criar ligações entre as páginas da Web, criando assim documentos com o conceito de hipertexto.

Para que o conteúdo de um documento HTML possa ser formatado, exibido na Internet temos que usar um programa chamado browser, o navegador. Ele lê o conteúdo do arquivo, interpreta os comandos e exibe sua página. Como já foi citado acima, existem diversos tipos de browsers disponíveis no mercado.

1.3 OS NAVEGADORES

Para visualizar toda informação disponível na Internet é necessário um Browser, navegador. Há vários disponíveis no mercado. Os mais conhecidos são o Internet Explorer e Netscape Navegador.

Os navegadores podem acessar vários tipos de dados, não só do protocolo FFTP mas, também FTP, Golpher, Usenet News, Telnet, etc.

1.4 SERVIDOR WEB

Para exibir páginas na Web e navegar por elas, você precisará apenas de um navegador da Web. Para divulgar páginas na Web, você precisará, na maioria dos casos, de um servidor Web.

Servidor Web é o programa que é executado em um site da Web e que é responsável por atender solicitações de arquivos feitos pelo navegador da Web. Você precisará de um servidor Web para divulgar documentos na Web.

1.1-CONCEITO SOBRE HTML

A HTML é uma linguagem de marcação de texto. Criar um documento em uma linguagem de marcação significa que você começa com o texto da sua página e inclui tags especiais no início e no final de determinadas palavras ou parágrafos.

A sigla HTML significa Hypertext Markup Language e que em português quer dizer Linguagem de Marcação de hipertexto.

O HTML não é uma linguagem de programação, mas, sim de marcação de texto e é usada para criar páginas webs. HTML faz parte dos três pilares básicos da programação web.

O HTML foi criado pelo físico cientista Timoth John Berners-Lee em 1990. Actualmente o HTML se encontra na sua versão5 que trouxe muitas novidades.

As tags normalmente especificadas em pares, delimitado em texto que sofrerá algum tipo de formatação. As tags são identificadas por estarem entre os sinais < > e < / >.

1.1.1 HTML Semântico

A linguagem HTML tem o papel de estruturar documentos para web, mas, também descrever o significado do conteúdo presente nesses documentos por meio de tags semânticas.

O HTML Semântico tem com objecto descrever o significado do conteúdo presente em documentos HTML tornando-o, mais claro tanto para programadores quanto para browser e outros engines que processam essa informação.

Os elementos que fazem parte do HTML Semânticos são:

`<header>`

É utilizado para representar o cabeçalho de um documento ou seção declarado no HTML. Nele podemos inserir elementos para representar imagem, parágrafos ou mesmo listas de navegação.

`<section>`

Representa uma secção dentro de um documento e geralmente contém um título, o qual é definido por meio de um dos elementos.

<footer>

Representa um rodapé de um documento, como a área presente no final de uma página web. Normalmente é utilizado para descrever informações de autoria como nome e contacto do autor e data de criação do conteúdo.

<nav>

É utilizada quando precisamos representar um agrupamento de links de navegação, que por sua vez são criados com os elementos , e <a>.

<article>

Utilizamos quando precisamos declarar um conteúdo que não precisa de outros para fazer sentido em um documento HTML.

<aside>

É utilizado quando precisamos criar um conteúdo de apoio adicional ao conteúdo principal.

1.1.2 Tipos de lista em HTML

As listas são muito importantes quando queremos listar alguns itens no site e também para a criação do menu de navegação. Em HTML existem três tipos de listas:

1-Lista Ordenadas

2- Listas Desordenadas

3-Lista Definições

Listas Ordenadas

Nas listas ordenadas, os itens serão ordenados, como o próprio nome já diz e listado de forma que seja por números, letras ou algarismo romanos.

Lista Desordenada

Lista Desordenada iniciadas com a tag e são representadas por pequenas bullets.

Lista Definição ou Descrição

Listas de definição são constituídas de duas partes: um termo e uma descrição. Para codificar uma lista de definição são necessários três elementos HTML.

- 1- Um container <dl>
- 2- Um termo de definição
- 3- Uma descrição

1.1.3 Tabelas em HTML

As tabelas são definidas pela tag <table>. Uma tabela é dividida em linhas que são representados pela tag <tr> e cada linha é dividida células de dados representado pela tag <td>.

As letras td significa “table data” que é o conteúdo de uma célula de dados.

As letra tr significa “table row”.

Inserção de links em HTML

Links são pontas clicáveis de uma página que apontam para outra página.

Link, em Inglês, significa ligação, elo ou conexão. E é isso que os links fazem em HTML, eles conectam uma página da internet até outra, eles levam o usuário de um local a outro.

Um link pode ser um texto, uma imagem, um e-mail, um endereço FTP. Por isso são chamado endereço ou referencias de um local para outra.

Links internos e externos

Internal links (ou links internas) apontam para páginas dentro do seu próprio site, ou seja, e possuem como fonte o mesmo domínio.

São considerados externos tantos aqueles links que apontam para sua página quanto os que você aponta para páginas de outros sites.



Figura 1-HTML5

1.2-Conceito Sobre CSS

CSS é a sigla para Cascading Style Sheets que em português significa folha de estilo em cascatas. A verdade é que o HTML é uma linguagem que não serve para formatações, ela apenas serve para definir o conteúdo da página e descrever o significado de cada elemento. Não cabe ao HTML fornecer informações sobre apresentação dos elementos, cores, tamanho dos textos e posicionamentos. Isto é trabalho das CSS.

Usa-se muito a seguinte frase que resume muito bem a função de cada linguagem: HTML para estruturar e CSS para apresentar. Por isto em 1996 Tim Berners Lee e a w3c lançaram a primeira versão do CSS. 1- 0 CSS1.

1.2.1 As regras de CSS: unidade básica de uma folha de estilo

A unidade básica de CSS, a menor porção do código que capaz de produzir efeitos de estilização. Uma regra CSS é composta por duas partes: um selector e uma declaração. Dentro de uma declaração encontramos também dois elementos: uma propriedade e um valor.

Seletor: é o alvo da regra css. Geralmente é uma tag HTML.

Declaração: determina os parâmetros de estilização.

Propriedade: define qual a característica do seletor a ser usada.

Valor: é a qualificação ou quantificação da propriedade.

1.2.2 Como ligar HTML com CSS

Existem três formas de aplicar CSS num documento HTML são eles:

- 1- Estilo inline
- 2- Estilo incorporado

3- Estilo externo

Estilo inline

No estilo inline nós aplicamos a estilização diretamente dentro da tag da abertura do elemento a estilizar usado o atributo style isso quer dizer que toda a tag HTML tem o seu atributo style.

Este método dificulta a manutenção e retira um dos maiores poderes das folhas de estilo que é o controle centralizado da apresentação.

Estilo incorporado

Outro método de escrever a folha de estilo no próprio documento HTML é com emprego do elemento style.

A vantagem deste método em relação ao método inline é que, localizamos com, mas, facilidade a folha de estilo, mas a desvantagem é que a folha é colocada no mesmo lugar do que o documento. Não seria sensato vincular uma mesma folha de estilo a vários documentos empregando este método.

O elemento style deve estar contido na secção head do documento.

Estilo externo

Folhas de estilo externo são aquelas que não estão escritas no documento HTML. Trata-se de um arquivo de texto contendo as regras de estilo e os comentários CSS. Um arquivo de folha de estilo deve ser guardado com a extensão e pode ser vinculado ou ligado ao documento HTML de duas maneiras:

1-Folha de estilo linkada

2-Folha de estilo importada.



Figura 2-CSS

1.3-Conceito sobre Java Script

Java Script é a linguagem de programação, mas, popular no desenvolvimento web.

O termo Java Script nada tem haver com a linguagem de programação Java, está é uma grande confusão feita pela comunidade de desenvolvimento web.

A linguagem Java Script é uma linguagem do tipo cleint side, ou seja, ela é executada no computador do usuário. Está é uma das definições mais importantes desta linguagem, pois, e isto explica muita coisa.

O Java Script foi inventado por Brendon Eich, confundador do projecto Mozilla, da fundação Mozilla e da corporação Mozilla.

A linguagem Java Script nasceu de uma linguagem chamada de Livescript no acordo com a SUN para alavancar o uso das duas.

1.3.1 Característica da linguagem Java Script

O Java Script, como o próprio nome sugere, é uma linguagem de scripting. Uma linguagem de scripting é comumente definida como uma linguagem de programação que permite ao programador controlar uma ou mais aplicações de terceiros. No caso do Java Script, podemos controlar alguns comportamentos dos navegadores através de trechos de código que são enviados na página HTML.

Outra característica comum nas linguagens de scripting é que normalmente elas são linguagens interpretadas, ou seja, não dependem de compilação para serem executadas.

Essa característica é presente no Java Script: o código é interpretado e executado conforme é lido pelo navegador, linha a linha, assim como o HTML.

1.3.2 Console do navegador

Existem várias formas de executar códigos Java Script em uma página. Uma delas é executar códigos no que chamamos de Console.

O console faz parte de uma série de ferramentas embutidas nos navegadores especificamente para nós que estamos desenvolvendo um site. Essa série de ferramentas é o que chamamos de Developer Tools.



Figura 3-JS

1.4 PHP

O PHP(Hypertext Preprocessor) é uma linguagem interpretada livre que actua do lado do servidor, tem seu melhor uso no desenvolvimento de aplicações web dinâmicas.

Figura entre as primeiras linguagens passíveis de inserção em documentos HTML e é uma linguagem extremamente modularizada. O que a torna ideal para servidores web.

Em 1994/1995 foi criado o PHP ½ por Rasmus que iria ser usado no desenvolvimento da sua página pessoal que era um conjunto de script que foi chamado de Personal Home Page Tools.

Em 1997/1998 foi criado o PHP 3 por Adi e Zeev eles entram para reescrever a base do PHP e dizem que é uma linguagem completa de programação.

Em 2000/2003 foi criado o PHP 4 neste tempo a versão 3 do php foi abandonada e surgiu o php 4 com recursos de OO, o que atraiu mas desenvolvedores para linguagem.

Em 2004/2014 foi criado o PHP 5 na qual ocorre a ascensão do php com a versão 2 do Zend engine, agora totalmente orientado a objecto banco de dados com PD,JSON nativo, namespaces, interfaces, traits e diversos outros recursos que solidificaram o PHP.

Em 2015/20xx foi criado o PHP 7 com performance supreendente e fortificação em recursos de OO, o PHP é visto como uma linguagem madura presente em mas de 83.5% dos sites na internet sendo a mas utilizada do mundo.

1.4.1 Interoperabilidade

Interoperabilidade é a capacidade de um sistema de se comunicar de forma transparente ou mais próximo disso com outro(sistema, componente, API).Por meio de Open Standard ou Ontologies.

Open Standard são padrões abertos, livres e disponíveis para acesso e implementação que independem de royalties, outras taxas ou discriminação de uso.

Ontologies são ontologias com modelos de dados que representam um conjunto de conceitos possuindo o domínio da aplicação e os relacionamentos entre ela.



Figura 4-PHP

1.5 Banco de Dados

Banco de dados é um conjunto de dados gravados em média apropriada organizado em coleções que se relacionam entre si e representam algo que tenha sentido.

A estrutura onde estão armazenado os dados podendo ser representada por um ou mais arquivo.

Segundo Korth, um banco de dados “é uma coleção de dados inter-relacionados, representando informações sobre um domínio específico”, ou seja, sempre que for possível agrupar informações que se relacionam e tratam de um mesmo assunto, posso dizer que tenho um banco de dados.

Para criar um banco de dados em um computador, é necessário um servidor local.

1.5.1 Conceito de Sistema de Gerenciamento de Banco de Dados

Um SGBD (do Inglês DBMS – Data Base Management System) é uma ferramenta construída para gerir a informação que se encontra armazenada numa base de dados.

Um SGBD ajuda a adicionar nova informação à medida que fica disponível, a actualizar a informação sempre que necessário, a catalogar rapidamente a informação, a realizar cálculos com os dados e a imprimir a informação numa grande variedade de formatos. Mais importante que tudo isto, permite-lhe encontrar exactamente a informação pretendida.

1.5.2 SQL

A linguagem SQL é usada para comunicar com bases de dados. É uma linguagem padrão para base de dados, declarativa, ao contrário das linguagens habituais que são do tipo procedimental. O SQL é uma linguagem de programação que serve criar, actualizar e responder a questões apartir da informação armazenada num Sistema Gerenciador de Base de Dados (SGBD).

A sigla SQL significa S+Structured Query Language que em português significa Linguagem Estruturada para Consultas.

1.5.2.1 A linguagem SQL é constituída por três sublinguagens:

DDL (Data Defemetron Language)

- Create
- Alter
- Drop

DML (Data Manipulation Language)

- Select

- Insert
- Update
- Delete

DCL (Data Control Language)

- Grant
- Revok

1.6-Plataforma.Net

A Plataforma .Net é um framework de desenvolvimento web que fornece facilidades de reutilização e reaproveitamento de código, dentre outras facilidades de uso.

A Plataforma .Net fornece diversos recursos relacionados a interface, acesso a dados, conectividade a banco de dados, criptografia, comunicações de redes e ambiente de desenvolvimento web.

A Plataforma .Net oferece aos desenvolvedores um ambiente virtual controlado: em vez de escrever código para uma aplicação específica, ele passa a escrever para a plataforma é um pacote de classes com soluções codificadas para problemas comuns de programação, suporta varias linguagens de programação no ambiente.

A Plataforma . Net é uma iniciativa da empresa Microsoft, que visa uma plataforma única para desenvolvimento e execução de sistemas e aplicações.

Todo e qualquer código gerado para . Net pode ser executado em qualquer dispositivo desde que este tenha uma framework.

Desenvolvidor	Microsoft
Plataforma	Windows

Lançamento	13 de Fevereiro de 2002
Versão estável	4.8 (18 de Abril de 2019)
Idiomas	Variados
Sistemas operacionais	Windows98 ou posterior, Windows NT 4.0 ou posterior
Gênero	Framework
Licença	Mista
Estado de desenvolvimento	Activo
Página oficial	Dot net.microsoft.com

Tabela 1-Plataforma . net

1.6.1-Principais linguagens de programação utilizada atualmente na plataforma .Net:

C#;

VB.Net;

F#;

Delphi prisma.

1.6.2-Algumas vantagens da plataforma .Net

Maior eficiência: na detecção de erros, na correção de erros e solução de problemas;

Eficiência e facilidade de acesso á informação;

Flexibilidade suporta varias linguagens de programação;

Melhor performance: já que o gerenciamento de memória é realizado pelo sistema operacional e não pelo aplicativo.

1.6.3- Características da plataforma .Net

Disponibiliza um hall de dispositivos que podem ser utilizados juntos em um mesmo ambiente de desenvolvimento web;

Sincronização de informações entre usuários e dispositivos;

Faz maior uso de XML em vez de HTML resultando em mais capacidade interativa para sites;

Armazenamento centralizados de dados;

A capacidade de integrar vários meios de comunicação, como e-mail, fax e telefones;

A capacidade de criar módulos reutilizáveis. O framework é executado em Common Language Runtime- CLR(Ambiente de execução independente de linguagem).

Capitulo-#2

Programação ou arquitetura em camada

2.1-Programação ou arquitetura em camada

Estimula a organização da arquitetura do sistema em um conjunto de camadas coesas com fraco acoplamento entre elas. Cada camada possui um propósito bem definido. A camada superior conhece apenas a camada imediatamente inferior (que fornece seus serviços através de uma interface).

Cada camada é formada por um conjunto de classes com um determinado propósito.

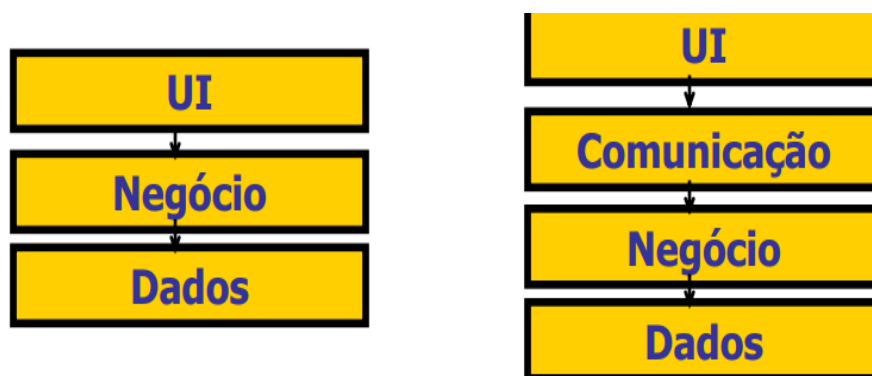


Tabela 2-Programação em camada

Propósito de Cada Camada

UI: agrega as classes do sistema com as quais os usuários interagem.

Negócio: mantém as classes do sistema responsáveis pelos serviços e regras do negócio.

Dados: camada responsável pelo armazenamento e recuperação dos dados persistentes do sistema.

Comunicação: responsável pela distribuição do sistema em várias máquinas.

Vantagens e Desvantagem

Vantagens: Separação de código relativo a interface com o usuário (UI), comunicação, negócio e dados. Permite a mudança de implementação de uma camada sem afetar a outra, desde que a interface entre as mesmas seja mantida. Possibilita que uma camada trabalhe com diferentes versões de outra camada.

Desvantagem: Aumento no número de classes existentes no sistema.

2.2-Padrão Arquitetura em Camadas

Exemplos de diferentes configurações do padrão arquitetura em camadas usando tecnologia java.

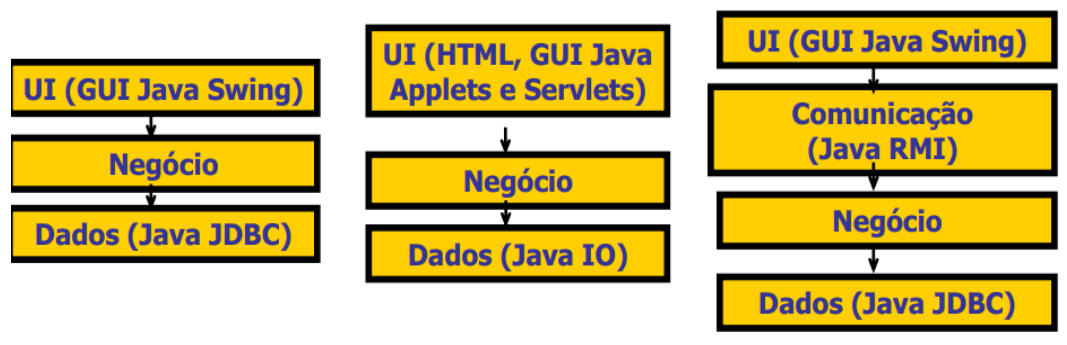


Tabela 3-Padrão em camada

2.3-Arquitetura ou modelo em 3 camadas

No modelo 3 camadas, a lógica de apresentação esta separada em sua própria camada lógica e física. A separação em camadas lógicas torna os sistemas mais flexíveis, permitindo que as partes possam ser alteradas de forma independente.

As funcionalidades da camada de negócio podem ser divididas em classes e essas classes podem ser agrupadas em pacotes ou componentes, reduzindo as dependências entre as classes e pacotes, podem ser reutilizadas por diferentes partes do aplicativo e até por aplicativos diferentes. O modelo de 3 camadas tornou-se a arquitetura padrão para sistemas corporativos com base na Web.

As três partes de um ambiente modelo três camadas são: camada de apresentação, camada de negócio e camada de dados.

Camada de apresentação

É a chamada GUI (Graphical User Interface), ou simplesmente interface. Esta camada interage diretamente com o usuário, é através dela que são feitas as requisições como consultas.

Camada de negócio

Também chamada de lógica empresarial, regras de negócio ou funcionalidade. É nela que ficam as funções e regras de todo o negócio. Não existe uma interface para o usuário e seus dados são voláteis, ou seja, para que algum dado seja mantido deve ser utilizada a camada de dados.

Camada de Dados

É composta pelo repositório das informações e as classes que as manipulam. Esta camada recebe as requisições da camada de negócios e seus métodos executam essas requisições em um banco de dados. Uma alteração no banco de dados alteraria apenas as classes da camada de dados, mas o restante da arquitetura não seria afetado por essa alteração.

Arquitetura em 3 camadas

Entre as camadas UI e Negócio haverá sempre uma interface Java que uma classe Fachada do sistema implementará. A classe Fachada é utilizada para oferecer um caminho único para acesso aos serviços da camada de regras de negócio. As classes da UI, portanto, comunicam-se apenas com a classe Fachada, que por sua vez colabora com as outras classes internas da camada de regras de negócio para oferecer os serviços.

O **Controlador** pode conter regras de controle do sistema e delega ações da fachada para a camada de acesso a dados. v Entre as camadas Negócio e Dados haverá sempre uma interface Java que uma classe Repositório implementará.

O **Repositório** armazena os objetos persistentes do sistema em algum meio de armazenamento físico (banco de dados, arquivo, etc.).

2.6-Modelo em quatro Camadas

Em camadas distintas podem ser delineadas da seguinte forma:

Camada de armazenamento dos dados (DS - Data Store): Camada onde os dados são armazenados. Pode ser um banco de dados relacional (RDBMS), um arquivo XML, texto ou qualquer outro sistema de armazenamento de informações;

Camada de acesso aos dados (DAL-Data Access Layer): Camada onde reside o código que trata da obtenção, tratamento e retorno dos dados salvos na camada de armazenamento de dados. É composta pelo repositório das informações e as classes que as manipulam.

Camada da lógica de negócio (BLL-BusinessLogic Layer): Camada onde está o código que obtém os dados retornados pela **DAL** e os repassa ao cliente de uma forma mais intuitiva, efetuando a validação lógica conforme as regras do negócio e verificando se os dados

estão no formato adequado. Também chamada de lógica empresarial, regras_de negócio ou funcionalidade. É nela que ficam as funções e regras de todo o negócio;

Camada de apresentação (UI - User Interface) - Camada onde reside o código que define o que será efetivamente apresentado ao usuário, efetuando a formatação dos dados, navegação e ajuda ao usuário final. Esta camada interage diretamente com o usuário, é através dela que são feitas as requisições como consultas;

Camada de Conteúdo: para essa camada o objetivo é fazer a demarcação dos blocos de conteúdo propostos pelo panejamento e pelo layout do website. Para isso a principal linguagem utilizada é o HTML, que está em sua versão 5.

Camada de Formatação: nesta camada o objetivo é formatar o conteúdo do modo como ele foi formatado na criação do layout. Para isso são utilizadas imagens e a linguagem CSS, que está em sua versão 3.

Camada de Comportamento: Nesta camada são incluídos movimentos e recursos de interação do usuário com o conteúdo, para melhorar sua experiência com o site. Para isso é utilizada a linguagem JAVASCRIPT e alguns framework JQUERY e seus plugins.

Camada de Automação: esta camada envolve a automação do gerenciamento das informações do website, tanto por parte de usuários administradores quanto pelos visitantes do site. Para isso são utilizadas linguagens avançadas de programação, como o C#, Java ou PHP e banco de dados, como MsSQL ou MySQL.

Model View Controller (Modelo-Visão-Controle-MVC)

Modelo Visão Controle é um padrão de projecto de software ou padrão de arquitectura de software formulado na década de 1970, focada no reuso de código e a separação de conceitos em três camadas, onde a apresentação dos dados e interação dos usuários (front-end) são separados dos métodos que interagem com o banco de dados (back-end).

Normalmente é usado para o desenvolvimento de interfaces de usuário que divide uma aplicação em partes (camadas/componentes) interconectadas. Isto é feito para separar representações de informação internas dos modos como a informação é apresentada para e aceita pelo usuário, levando ao desenvolvimento paralelo de maneira eficiente.

Os componentes do modelo MVC

Tradicionalmente usado para interfaces gráficas de usuário (GUIs), esta arquitetura tornou-se popular para projetar aplicações web e até mesmo para aplicações móveis, para desktop e para outros clientes.^[6] Linguagens de programação populares como Java, C#, Object Pascal/Delphi, Ruby, PHP, JavaScript e outras possuem frameworks MVC populares que são atualmente usados no desenvolvimentos de aplicações web.

Camada de modelo ou da lógica da aplicação (Model)

Modelo é a ponte entre as camadas Visão (*View*) e Controle (*Controller*),^[2] consiste na parte lógica da aplicação, que gerencia o comportamento dos dados através de regras de negócios, lógica e funções.^[7] Esta fica apenas esperando a chamada das funções,^[7] que permite o acesso para os dados serem coletados, gravados e, exibidos.^[2]

É o coração da execução, responsável por tudo que a aplicação vai fazer a partir dos comandos da camada de controle em um ou mais elementos de dados, respondendo a perguntas sobre o sua condição e a instruções para mudá-las. O modelo sabe o que o aplicativo quer fazer e é a principal estrutura computacional da arquitetura, pois é ele quem modela o problema que está se tentando resolver. Modela os dados e o comportamento por trás do processo de negócios. Se preocupa apenas com o armazenamento, manipulação e geração de dados. É um encapsulamento de dados e de comportamento independente da apresentação.

Camada de apresentação ou visualização (View)

Visão pode ser qualquer saída de representação dos dados, como uma tabela ou um diagrama. É onde os dados solicitados do Modelo (*Model*) são exibidos.^[2] É possível ter várias visões do mesmo dado, como um gráfico de barras para gerenciamento e uma visão tabular para contadores. A Visão também provoca interações com o usuário, que interage com o Controle (*Controller*). O exemplo básico disso é um botão gerado por uma Visão, no qual um usuário clica e aciona uma ação no Controle.^[2]

Não se dedica em saber como o conhecimento foi retirado ou de onde ela foi obtida, apenas mostra a referência. Segundo Gamma et al (2006), "A abordagem MVC separa a View e Model por meio de um protocolo inserção/notificação (subscribe/notify). Uma View deve garantir que sua expressão reflita o estado do Model. Sempre que os dados do Model mudam, o Model altera as Views que dependem dele. Em resposta, cada View tem a oportunidade de modificar-se". Adiciona os elementos de exibição ao usuário : HTML, ASP, XML, Applets. É a camada de interface com o usuário. É utilizada para receber a entrada de dados e apresentar visualmente o resultado.

Camada de controle ou controlador (Controller)

Controle é o componente final da tríade, faz a mediação da entrada e saída, comandando a visão e o modelo para serem alterados de forma apropriada conforme o usuário solicitou através do mouse e teclado.^[7] O foco do Controle é a ação do usuário, onde são manipulados os dados que o usuário insere ou atualiza, chamando em seguida o Modelo.^[2]

O Controle (*Controller*) envia essas ações para o Modelo (*Model*) e para a janela de visualização (*View*) onde serão realizadas as operações necessárias.

Interações dos componentes

Além de dividir a aplicação em três tipos de componentes, o desenho MVC define as interações entre eles.

- O **Controlador (controller)** envia comandos para o modelo para atualizar o seu estado (por exemplo, editando um documento). O controlador também pode enviar comandos para a visão associada para alterar a apresentação da visão do modelo (por exemplo, percorrendo um documento).
- Um **modelo (model)** armazena dados e notifica suas visões e controladores associados quando há uma mudança em seu estado. Estas notificações permitem que as visões produzam saídas atualizadas e que os controladores alterem o conjunto de comandos disponíveis. Uma implementação *passiva* do MVC monta estas notificações, devido a aplicação não necessitar delas ou a plataforma de software não suportá-las.
- A **visão (view)** gera uma representação (Visão) dos dados presentes no modelo solicitado, fazendo a exibição dos dados, sendo ela por meio de um html ou xml.

Uso em aplicação web

Mesmo tendo sido desenvolvida inicialmente para computação pessoal, o MVC foi drasticamente adaptado como uma arquitetura para as aplicações web em todas as maiores linguagens de programação. Muitos frameworks de aplicação comerciais e não comerciais foram desenvolvidos tendo como base esse modelo. Tais frameworks variam em suas interpretações, principalmente no modo que as responsabilidades MVC são separadas entre o cliente e servidor.

Os frameworks web MVC mais recentes levam uma abordagem de thin client que quase colocou o modelo, a visão e a lógica do controlador inteiros no servidor. Nesta abordagem, o cliente envia requisições de hiperlink ou entrada de formulário ao controlador e então recebe uma página web completa e atualizada (ou outro documento) da visão. O modelo existe inteiramente no servidor. Como as tecnologias de cliente amadureceram, frameworks como JavaScriptMVC e Backbone foram criados o que permite que os componentes MVC executem parcialmente no cliente (ver também AJAX).

Um caso prático é uma aplicação web em que a visão é um documento HTML (ou derivado) gerado pela aplicação. O controlador recebe uma entrada `GET` ou `POST` após um estímulo do utilizador e decide como processá-la, invocando objetos do domínio para tratar a lógica de negócio, e por fim invocando uma visão para apresentar a saída.

Vantagens e Desvantagens do modelo MVC

Vantagens do modelo MVC

1. Como o modelo MVC gerencia múltiplos views usando o mesmo modelo é fácil manter, testar e atualizar sistemas compostos;
2. É muito simples adicionar novos clientes apenas incluindo seus views e controles;
3. Torna a aplicação escalável;
4. É possível ter desenvolvimento em paralelo para o modelo, visualizador e controle pois são independentes;
5. Facilita o reuso do código;
6. Melhor nível de sustentabilidade, pois facilita a manutenção da aplicação;
7. Fácil transformação da interface, sem que haja necessidade de modificar a camada de negócio;
8. Melhor desempenho e produtividade, graças a estrutura de pacotes modulares;
9. A arquitetura modular permite aos desenvolvedores e designers desenvolverem em paralelo;
10. Partes da aplicação podem ser alteradas sem a necessidade de alterar outras.

Desvantagens do modelo MVC:

1. Necessita de um tempo maior para explorar e modelar o sistema;
2. Requer mão-de-obra especializada;
3. À medida que o tamanho e a complexidade do projeto crescem, a quantidade de arquivos e pastas continuará aumentando também. Os interesses de UI (interface do usuário) (modelos, exibições, controladores) se localizam em várias pastas, que não são formadas em grupos por ordem alfabética.

Justificativa

A popularidade dessa arquitetura pode ser vista principalmente no desenvolvimento de aplicações WEB. Isso é uma consequência do ambiente ter um alcance ilimitado e com constante crescimento, onde a escalabilidade é um dos fatores mais importantes.

Com o aumento da complexidade das aplicações desenvolvidas, sempre visando a programação orientada a objeto, torna-se relevante a separação entre os dados e a apresentação das aplicações. Desta forma, alterações feitas no *layout* não afetam a manipulação de dados, e estes poderão ser reorganizados sem alterar o *layout*.

Esse padrão resolve tal problema através da separação das tarefas de acesso aos dados e lógica de negócio, lógica de apresentação e de interação com o utilizador, introduzindo um componente entre os dois: o controlador.

A separação de propósitos que o paradigma MVC propõem ajuda na escalabilidade e padronização de um produto. Manter um código que cresce constantemente em grandes cor-

porações é uma tarefa árdua, e pequenas equipes também podem sofrer para aumentar a escalabilidade de um produto se não houver um preparo.

Essa divisão de preocupações, ajuda na separação de tarefas e torna-as mais específicas, uma consequência disso é que os códigos possuem um propósito melhor definido, o que facilita no teste e no contexto de seu desenvolvimento e na reutilização do mesmo.

Plataformas de desenvolvimento

Java

- Apache Struts
- Brutos Framework
- Click Framework
- JSF
- Mentawai
- MVC 1.0 (Ozark)
- Neo Framework
- PlayFramework
- Spring MVC
- Tapestry
- VRaptor
- WebWork

PHP

- Akelos
- CakePHP - para as versões 4 e 5
- CodeIgniter - para as versões 4 e 5
- FuelPHP - para versões 5.3+
- iGrape
- Kohana Framework - para a versão 5
- LightVC - para a versão 5.
- Laravel - para a versão 5.3+
- Megiddo - para a versão 5
- Oraculum PHP Framework - para a versão 5
- PageletBox - IDE para Celular WAP com framework MVC embutido em PHP 5
- Phalcon - Extensão em C visando alta performance e baixo consumo de recursos - para a versão 5
- PHPBurn
- PHPonTrax - para a versão 5
- PRADO - para a versão 5
- Seagull
- Spaghettsymfony - para a versã

Capitulo#3

Implementação do Projecto

3- Implementação de um sistema de Divulgação e Recuperação de Documentos Perdidos

3.1- Descrição do Sistema:

A RDP é um sistema de serviço que ajuda a recuperar todo tipo de documento a nível nacional. O processo de recuperação de documentos vai ocorrer da seguinte forma:

Sabendo que o nosso sistema será implementado em algumas identificadora quando um cidadão pela primeira vez tratar o seu documento em uma identificadora, lhe-é atribuído uma conta no nosso sistema, onde será armazenado todos os seus documentos para facilitar no ato da recuperação dos seus dados. Caso o cidadão tiver uma conta e quiser tratra um outro documento o sistema reconhecerá e simplesmente lhe adicionará um novo documento.

Problema

No País, nota-se claramente a existência de muitos documentos perdidos, o problema está no tempo em que as pessoas perdem , a espera da segunda via dos seus documentos.

Devido a pandemia da covid-19, o tempo de espera no atendimento para aquisição dos seus documentos, poderá ser maior que a normal.

Proposta de solução

Consoante os nossos estudos podemos notar uma grande debilidade no que tange a recuperação de documentos perdidos ,então surgiu a ideia de criarmos este sistema para ajudar o nosso povo ,na recuperação dos seus documentos.

Como queremos que o sistema funciona?

Uma vez que será muito complicado por parte dos cidadãos colocar net ao dia de 100kz, para visualizar os seus documentos ou postar um documento que ele achou perdido, sem ter nenhuma recompensa por parte do sistema, por isso queremos que o nosso sistema seja um software livre que nem o facebook onde podemos acessar apenas com prazo permitido que o sistema funciona de uma forma correta para que os cidadão colaborem ao fazer outros encontrarem os seus documentos.

3.2-O que é um documento?



Figura 5-Documentos

Um documento é qualquer registro de informação. Independentemente do formato ou suporte utilizado para registrá-las.

Os órgãos típicos de documentação são os **museus**, os **arquivos**, as **bibliotecas**, e os **centros de documentação**, cada um com as suas características peculiares.

Classificação dos documentos.

Os documentos classificam-se em públicos e privados.

- **Públicos:** são emitidos e/ou recebidos por um órgão governamental na **gestão** de suas atividades ou mediante procuração de autoridade pública. Por exemplo, **escritura pública** de compra e venda emitida por um **notário**.
- **Privados:** são emitidos por um particular (**pessoa física**), ou por autoridade pública fora de suas funções, **atribuições**, ou **competência**.

3.3-Quanto ao seu valor

O valor dos documentos devem ser claramente definidos, sustentáveis e coexistentes ao período que foram gerados. Desta forma tem-se os seguintes valores:

- **Administrativo** - Valor que um documento possui para a administração produtora do arquivo, na medida em que informa, fundamenta ou aprova seus **atos** presentes ou futuros.
- **Fiscal** - Valor atribuído a documentos ou arquivos para comprovação de operações financeiras ou fiscais.
- **Informativo** - Valor que um documento possui pelas informações nele contidas, independente de seu valor probatório.

- **Legal** - Valor **processual** que um documento possui perante a lei para comprovar um fato ou constituir um direito.
- **Permanente** - Valor probatório ou valor informativo que justifica a guarda permanente de um documento em um arquivo. Também referido como valor histórico.
- **Primário** - Valor atribuído aos documentos em função do interesse que possam ter para o gerador do arquivo, levando-se em conta a sua utilidade para fins administrativos, legais e fiscais.
- **Probatório** - Valor intrínseco de um documento de arquivo que lhe permite servir de prova legal.
- **Secundário** - Valor atribuído aos documentos em função do interesse que possam ter para o gerador do arquivo, e para outros usuários, tendo em vista a sua utilidade para fins diferentes daqueles para os quais foram originalmente produzidos.

3.4- REQUISITOS

Os requisitos num sistema, “são funcionalidades ou características consideradas relevantes na ótica do utilizador”. Normalmente, representam o comportamento esperado do sistema, que na prática consiste num serviço que deve ser disponibilizado a um utilizador.

3.4.1- REQUISITOS FUNCIONAIS

Requisitos funcionais, “descrevem o que um sistema faz”. São os requisitos que inicialmente serão levantados, abrangendo a descrição de processamento a efetuar pelo sistema, entradas (inputs) e saídas (outputs) de informação em papel ou no ecrã que derivam da interação com pessoas e outros sistemas. A seguir, é apresentado uma tabela com os requisitos funcionais do sistema.

Tabela 4-Requisitos funcionais

RF01	Login do administrador
RF02	Login do cidadão/usuário
RF03	Fornece uma conta
RF04	Recebe notificação
RF05	Envia mensagens
RF06	Conversa com o cidadão
RF07	Envia Notificação
RF08	Permite o usuário consultar e editar o seu perfil
RF09	Indica a localização do utilizador
RF10	Permite a postagens de documentos

3.4.2- REQUISITOS NÃO FUNCIONAIS

Requisitos não funcionais “está relacionados com as características qualitativas do sistema, descrevendo a qualidade com que o sistema deverá fornecer os requisitos funcionais”. Abrange medidas de desempenho como, por exemplo, tempos de resposta, volume de dados ou considerações de segurança.

Tabela 5-Requisitos não funcionais

RNF01	Deverá verificar a autenticação do usuário	Segurança
RNF02	Deverá encriptar os dados a circularem bem com os dados de acesso o sistema.	Segurança
RNF03	Interação facilitada entre o usuário e o administrador	Confiabilidade
RNF04	Restrição controlada	Segurança
RNF05	Facíl uso no sistema para o Usuário quanto ao administrador	Flexibilidade

3.5-Identificação dos atores

A primeira tarefa a desenvolver para contruir um digrama de casos de uso é a identificação dos atores do sistema. Um Actor representa uma entidade externa que interage com o sistema.

Exemplo:

Cidadão: pessoa que divulga o documento;

Administrador: pessoa que ajuda a recuperar documentos perdidos.

3.5.1-Caso de Uso

Os casos de usos podem ser definidos numa perspectiva de negócio ou de sistema. Na primeira perspectiva, procura-se identificar a forma como se responde a um cidadão ou evento em termos de processo de negócio. Na perspectiva do sistema, procura-se caracterizar as funcionalidades que a aplicação a desenvolver deve disponibilizar aos usuários. Um caso de uso corresponde a um conjunto de ações executadas durante a realização de uma funcionalidade do sistema. Casos de uso concentram-se nas relações entre as funções do sistema e os usuários que delas participam de alguma forma. Um caso de uso de sistema tem as seguintes características

- Captura as ações para a realização de uma função do sistema, enfocando as interações entre os usuários e o sistema;
- É uma unidade coerente de passos, expressa como uma transação entre os atores o sistema, compondo-se tipicamente de várias ações dos atores e respostas do sistema;
- É uma sequência de ações que produzem resultados observáveis de valor para os usuários;
- Expressa o que acontece quando um caso de uso é executado, incluindo suas possíveis variações.

3.6-IDENTIFICAÇÃO DOS CASOS DE USO POR ATORES

Tabela 6-Identidade dos atores

Actor	Caso de uso
	Cadastra documento
	Elimina Usuário
	Cadastra ADM

Administrador	Cadastra documento
	Recebe notificação
	Elimina notificação
	Elimina Usuário
	Lista cidadão
	Recebe mensagem

Cidadão	Recebe mensagens
	Envia Notificação
	Informa localização
	Fecha Mensagens
	Cancela Notificação

Modelo entidade relacionamento

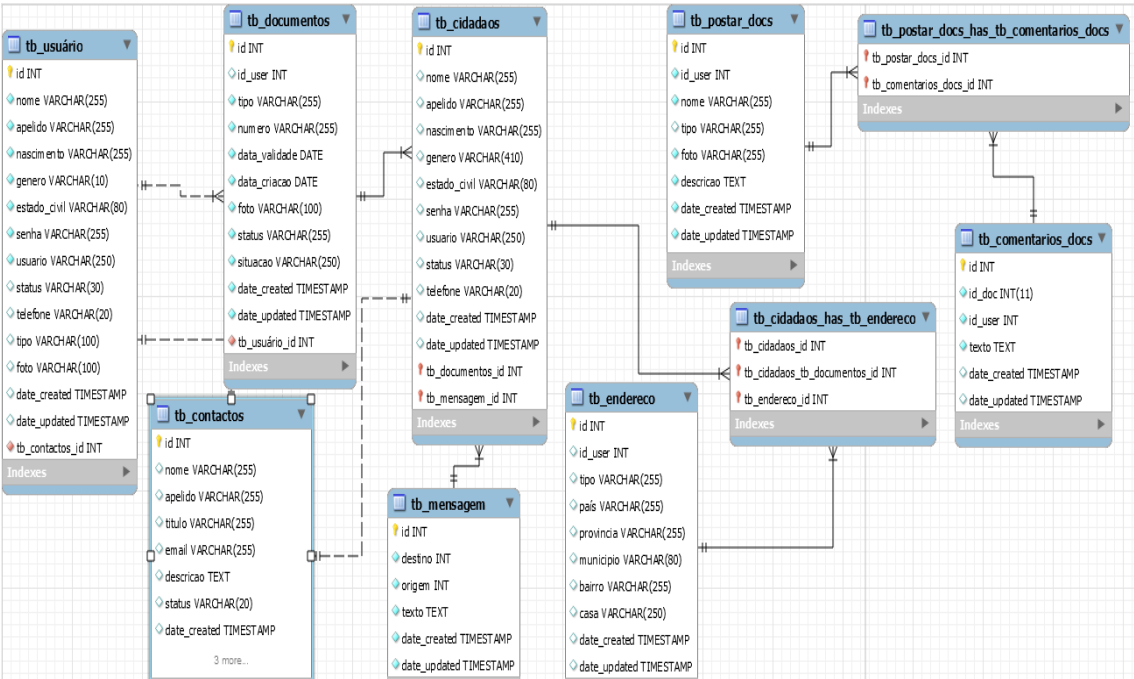


Figura 6-Modelo entidade relacionamento