-

**Eduardo Lopes Fonseca Gonzales**

**Gabriel Machado Dos Santos**

**Guilherme Angelo Silva**

**João Pereira Neto**

**SISTEMA GERENCIADOR DE DOCUMENTOS**

**Projeto Integrador**

Disciplinas Envolvidas: Engenharia de Software, Técnicas Avançadas de Banco de Dados Relacional e não Relacional, Técnicas Avançadas de Programação, Técnicas Avançadas de Programação Web e Mobile, Gestão Ágil de Projetos de Software.

Jales

2023

SUMÁRIO

[1 INTRODUÇÃO 3](#_Toc144809989)

[2 LEVANTAMENTO DE REQUISITOS DE SOFTWARE 4](#_Toc144809990)

[2.1 Descrição dos objetivos do sistema 4](#_Toc144809991)

[2.2 Descrição do sistema atual 4](#_Toc144809992)

[2.3 Descrição dos principais problemas 4](#_Toc144809993)

[2.4 Descrição dos requisitos funcionais 4](#_Toc144809994)

[2.5 Descrição dos requisitos não funcionais 4](#_Toc144809995)

[3 VISÃO DE CASO DE USO – UML 5](#_Toc144809996)

[3.1 Diagrama de Classes 5](#_Toc144809997)

[3.2 Dicionário de classes 5](#_Toc144809998)

[3.3 Definição dos Atores 5](#_Toc144809999)

[3.4 Lista de Casos de Uso 5](#_Toc144810000)

[3.4. Diagrama de Casos de Uso 5](#_Toc144810001)

[3.5. Diagrama de Casos de uso individuais 6](#_Toc144810002)

[3.6. Diagrama de Sequência 7](#_Toc144810003)

[3.7. Diagrama de Comunicação 7](#_Toc144810004)

[3.8. Diagrama de Atividade 7](#_Toc144810005)

[4 DEFINIÇÃO DA INTERFACE COM O USUÁRIO (UX) (3º semestre) 8](#_Toc144810006)

[4.1 Descrição de cenário 8](#_Toc144810007)

[4.2 Descrição de personas 8](#_Toc144810008)

[4.3 Esboços de tela (wireframes) 8](#_Toc144810009)

[4.4 Protótipos de tela 8](#_Toc144810010)

[5 BANCO DE DADOS 9](#_Toc144810011)

[5.1 Modelo Entidade Relacionamento 9](#_Toc144810012)

[5.2 Script das tabelas 9](#_Toc144810013)

[6 ARQUITETURA DE SOFTWARE 10](#_Toc144810014)

[6.1 Arquitetura de desenvolvimento 10](#_Toc144810015)

[6.2 Telas do sistema 10](#_Toc144810016)

[7 CONCLUSÃO 11](#_Toc144810017)

[8 REFERÊNCIAS 12](#_Toc144810018)

# INTRODUÇÃO

A geração de documentos de obra se torna fundamental para qualquer construção, sendo registros oficiais das etapas, decisões, normas e regulamentos que estão relacionados a uma obra.

Segundo Costa (2020), "O licenciamento para obras é imprescindível ao construir um imóvel. Quando um projeto para construção de um imóvel é aprovado pela prefeitura, significa que o mesmo atendeu à legislação e a construção pode ser iniciada após a liberação do alvará, documento autorizando o início dos serviços".

Com isto, a gestão destes documentos e o atendimento de novas atualizações destes dados se tornam essenciais. Segundo MOBUSS (2018), “A gestão de documentos que conta com ferramentas inovadoras assegura a disseminação de dados rápida, eficiente e uniforme.”.

Atualmente, na Prefeitura de Jales, a gestão dos documentos de aprovação de construções é realizada por um sistema que necessita de atualizações, resultando em dificuldades e prolongamento no processo de busca por esses documentos.

Nesse cenário, este projeto propõe o desenvolvimento de um software destinado ao gerenciamento de obras que tenham documentos aprovados, incluindo aqueles que estejam relacionados a esses registros. Este sistema oferecerá uma pesquisa mais robusta, com o objetivo de otimizar o tempo de resposta e minimizar o esforço necessário para localizar os documentos, ao mesmo tempo em que assegura o controle de acesso e a segurança desses arquivos.

# LEVANTAMENTO DE REQUISITOS DE SOFTWARE

## Descrição dos objetivos do sistema

## Descrição do sistema atual

O sistema atual utilizado pela Secretaria de Obras revela-se inadequado para suprir as crescentes necessidades e demandas cotidianas do órgão, principalmente devido à sua limitação quanto à capacidade de armazenamento de informações. Esta escassez de espaço para registros se traduz em obstáculos consideráveis na tarefa de localizar e gerenciar os dados essenciais.

Essa deficiência prejudica significativamente a eficiência administrativa e a capacidade de busca e resposta da Secretaria de Obras aos arquivos e documentos, sendo assim urgente a busca por uma solução mais robusta e eficaz para gerenciamento de informações.

## Descrição dos principais problemas

## Descrição dos requisitos funcionais

### Login Administrador

O sistema deve permitir que o usuário consiga efetuar login como administrador, validando se os dados informados estão corretos.

### Alterar Administrador

O sistema deve permitir o administrador alterar seus dados, caso precise.

### Logout Administrador

O sistema deve permitir que o administrador efetue logout de sua sessão, retornando a página inicial do sistema.

### Cadastrar Atendente

O sistema deve permitir que o administrador cadastre um atendente no sistema.

### Listar Atendente

O sistema deve permitir que o administrador liste todos os atendentes cadastrados no sistema.

### Desativar Atendente

O sistema deve permitir que o administrador desative um atendente cadastrado no sistema, colocando seu status como “Desativado”, impossibilitando o usuário efetuar login com a conta afetada, existindo somente para que outros registros dependentes dele não sejam removidos ou corrompidos.

### Excluir Atendente

O sistema deve permitir que o administrador exclua um atendente cadastrado no sistema.

Caso o atendente tenha participado de algum registro no banco de dados, efetua Desativar Atendente.

Caso não, ele o removerá do banco de dados, efetuando o logout da sessão aberta para a conta afetada, caso exista.

### Login Atendente

O sistema deve permitir que o usuário consiga efetuar login como atendente, validando se os dados informados estão corretos.

### Alterar Atendente

O sistema deve permitir que o administrador e o atendente alterem seus dados, caso precise.

### Logout Atendente

O sistema deve permitir que o atendente efetue logout de sua sessão, retornando a página inicial do sistema.

### Cadastrar Usuário Público

O sistema deve permitir que o administrador e o atendente cadastrem um usuário público no sistema.

### Listar Usuário Público

O sistema deve permitir que o administrador e o atendente listem todos os usuários públicos cadastrados no sistema.

### Desativar Usuário Público

O sistema deve permitir que o administrador e o atendente desativem um usuário público cadastrado no sistema, colocando seu status como “Desativado”, impossibilitando o usuário efetuar login com a conta afetada, existindo somente para que outros registros dependentes dele não sejam removidos ou corrompidos.

### Excluir Usuário Público

O sistema deve permitir que o administrador e o atendente excluam um usuário público cadastrado no sistema.

Caso o Usuário público tenha participado de algum registro no banco de dados, efetua Desativar Usuário Público.

Caso não, ele o removerá do banco de dados, efetuando o logout da sessão aberta para a conta afetada, caso exista.

### Login Usuário Público

O sistema deve permitir que o usuário consiga efetuar login como usuário público, validando se os dados informados estão corretos.

### Alterar Usuário Público

O sistema deve permitir que o administrador, atendente e o usuário público alterem seus dados, caso precise.

### Logout Usuário Público

O sistema deve permitir que o usuário público efetue logout de sua sessão, retornando a página inicial do sistema.

### Cadastrar Munícipe

O sistema deve permitir que o administrador cadastre um munícipe no sistema.

### Listar Munícipe

O sistema deve permitir que o administrador, atendente e usuários públicos listem todos os munícipes cadastrados no sistema.

### Alterar Munícipe

O sistema deve permitir que o administrador e o atendente alterem os dados do munícipe, caso precise.

### Desativar Munícipe

O sistema deve permitir que o administrador e o atendente desativem um munícipe cadastrado no sistema, colocando seu status como “Desativado”, assim, o registro não interage mais no banco de dados, existindo somente para que outros registros dependentes dele não sejam removidos ou corrompidos.

### Excluir Munícipe

O sistema deve permitir que o administrador e o atendente excluam um munícipe cadastrado no sistema.

Caso o engenheiro tenha participado de algum registro no banco de dados, efetua Desativar Munícipe.

Caso não, ele o removerá do banco de dados.

### Cadastrar Engenheiro

O sistema deve permitir que o administrador cadastre um engenheiro no sistema.

### Listar Engenheiro

O sistema deve permitir que o administrador, atendente e usuários públicos listem todos os engenheiros cadastrados no sistema.

### Alterar Engenheiro

O sistema deve permitir que o administrador e o atendente alterem os dados do engenheiro, caso precise.

### Desativar Engenheiro

O sistema deve permitir que o administrador e o atendente desativem um engenheiro cadastrado no sistema, colocando seu status como “Desativado”, assim, o registro não interage mais no banco de dados, existindo somente para que outros registros dependentes dele não sejam removidos ou corrompidos.

### Excluir Engenheiro

O sistema deve permitir que o administrador e o atendente excluam um engenheiro cadastrado no sistema.

Caso o engenheiro tenha participado de algum registro no banco de dados, efetua Desativar Engenheiro.

Caso não, ele o removerá do banco de dados.

### Cadastrar Fiscal

O sistema deve permitir que o administrador cadastre um fiscal no sistema.

### Listar Fiscal

O sistema deve permitir que o administrador, atendente e usuários públicos listem todos os fiscais cadastrados no sistema.

### Alterar Fiscal

O sistema deve permitir que o administrador e o atendente alterem os dados do fiscal, caso precise.

### Desativar Fiscal

O sistema deve permitir que o administrador e o atendente desativem um fiscal cadastrado no sistema, colocando seu status como “Desativado”, assim, o registro não interage mais no banco de dados, existindo somente para que outros registros dependentes dele não sejam removidos ou corrompidos.

### Excluir Fiscal

O sistema deve permitir que o administrador e o atendente excluam um fiscal cadastrado no sistema.

Caso o fiscal tenha participado de algum registro no banco de dados, efetua Desativar Fiscal.

Caso não, ele o removerá do banco de dados.

## Descrição dos requisitos não funcionais

# VISÃO DE CASO DE USO – UML

## 3.1 Diagrama de Classes

*Fazer um texto explicativo sobre o diagrama e inserir abaixo. As imagens devem conter o nome da figura e abaixo a fonte*

*Colocar em outra cor as classes que foram implementadas nas aulas de programação*

## 3.2 Dicionário de classes

Descrever cada uma das classes do diagrama e seus atributos. Exemplo:

Administrador: tem como objetivo controlar o fluxo de acesso de cliente e produtor e medir o nível de satisfação do cliente/produtor.

**Quadro 1** – Descrição Classe Administrador

|  |  |  |
| --- | --- | --- |
| Atributo | Tipo | Descrição |
| idAdministrador | Integer | Código que irá identificar o administrador. |

Fonte: Elaborado pelos autores

## 3.3 Definição dos Atores

*Inserir a figura dos atores e abaixo fazer uma descrição sobre o ator (es).*

## 3.4 Lista de Casos de Uso

*(Descrever sobre a Lista de Casos de Uso)*

Quadro 1 – Lista de Casos de Uso

| **Nº** | **Descrição do Caso de Uso** | **Entrada** | **Caso de Uso** | **Resposta** |
| --- | --- | --- | --- | --- |
| 01 | Funcionário Cadastra Livro | dados livro | Cadastrar Livro | Msg01 |
| 02 | Funcionário Cadastra Autor | dados autor | Cadastrar Autor | Msg02 |

## 3.4. Diagrama de Casos de Uso

*Fazer um texto explicativo sobre o diagrama e inserir-lo abaixo. As imagens deve conter o nome da figura e abaixo a fonte.*

**Figura 01** — Diagrama de Contexto – Visão do Paciente.



Fonte: Silva, 2016.

## 3.5. Diagrama de Casos de uso individuais

*(2 principais: cadastro, consulta, relatório, agendamento, controle, etc..)*

*Inserir o diagrama e abaixo colocar a documentação do mesmo.*

3.5.1 – Caso de uso: Cadastrar Atendimento

**Figura 02** — Diagrama de Contexto – Visão do Paciente

**

Fonte: Silva, 2016.

**Fluxo Normal**

1-AtorAdm solicita exclusão do registro da pessoa

2-Sistema envia mensagem 06: "Deseja excluir o registro?"

3-AtorAdm confirma exclusão do registro

4-Sistema exclui registro

5-Sistema envia msg07: "Registro excluído com sucesso!"

6-Sistema atualiza banco de dados e retorna ao Caso de Uso "Alterar Pessoa"

**Fluxo Exceção**

2-Sistema envia mensagem 06: "Deseja excluir o registro?"

2.1-AtorAdm não confirma exclusão do registro

2.2-Sistema cancela ação

2.3-Sistema retorna ao caso de Uso "Alterar Pessoa"

Fluxo Alternativo

4-Sistema exclui registro

4.1-Sistema identifica erros de conexão com banco de dados ao excluir registro

4.2-Sistema envia msg07:"Erro s de conexão de BD"

4.3-Sistema retorna ao item 1

## 3.6. Diagrama de Sequência

*(2 principais: cadastro, consulta, relatório, agendamento, controle, etc..)*

*Inserir o diagrama e abaixo colocar a documentação do mesmo.*

## 3.7. Diagrama de Comunicação

*(1 diagrama relevante do projeto)*

## 3.8. Diagrama de Atividade

*(1 diagrama de uma ação específica do sistema)*

# DEFINIÇÃO DA INTERFACE COM O USUÁRIO (UX) (3º semeestre)

## Descrição de cenário

(citar referências sobre o assunto e descrever pelo menos 2 cenários de uso relevantes para o sistema)

## Descrição de personas

(citar referências sobre o assunto e descrever pelo menos 2 personas relevantes para o sistema)

## Esboços de tela (wireframes)

(fazer o wireframe da tela de login, tela principal e dois exemplos de cadastros relevantes para o sistema e descrever sobre a imagem que foi colocada)

## Protótipos de tela

(fazer o protótipo da tela de login, tela principal e dos dois exemplos de cadastros relevantes para o sistema, preferencialmente em Html e css ou em bootstrap, se não for possível fazer o protótipo em algum aplicativo de edição pertinente)

(descrever sobre as telas demonstradas)

# BANCO DE DADOS

## Modelo Entidade Relacionamento

*(descrever sobre o que se refere o Mapeamento do Objeto Relacional)*

Figura 9 – Mapeamento do Objeto Relacional



## Script das tabelas

# ARQUITETURA DE SOFTWARE

## 6.1 Arquitetura de desenvolvimento

*Explanar sobre como foi realizado o processo de desenvolvimento, arquitetura de desenvolvimento (Sugestão: fazer um diagrama de Componentes para demonstrar a estrutura do sistema e suas camadas)*

## 6.2 Telas do sistema

*Colocar as telas desenvolvidas (figuras) e explicar sua função e funcionamento*

# CONCLUSÃO

Fazer uma conclusão se é viável o desenvolvimento do software.

# REFERÊNCIAS

*Colocar as referências utilizadas no projeto conforme os modelos abaixo:*

ASFHAL, C. R. **Gestão de segurança do trabalho e de saúde ocupacional**. São Paulo: Reichmann & Autores, 2005.

BEAIRD, J. Princípios do Web Design Maravilhoso. Rio de Janeiro: Altabooks, 2008.

CHIAVENATO, I. **Gestão de pessoas**: o novo papel dos recursos humanos nas organizações. 3. ed. Rio de Janeiro: Elsevier, 2008.

COSTA, H. J. Acidentes do trabalho: teremos nova lei acidentária?. **Jus Navigandi**, Teresina, v. 9, n. 664, 1 maio 2005. Disponível em: <http://jus2.uol.com.br/doutrina/texto.asp?id=6662>. Acesso em: 1 jul. 2010.

DELIBERATO, P. C. P. **Fisioterapia preventiva**: fundamentos e aplicações. São Paulo: Manole, 2002.

DIAS, E. C.; MELO, E. M. de. Políticas públicas em saúde e segurança no trabalho. In: MENDES, R. (Org.). **Patologia do trabalho**. 2. ed. atual. e ampl. São Paulo: Atheneu, 2007. v. 2. p. 1683-1720.

GUEDES, Gilleanes T. A. **UML Uma Abordagem prática**, 3 ed. São Paulo: Novatec, 2008.

GUEDES, Gilleanes T. A. **UML 2 Uma Abordagem prática**, São Paulo: Novatec, 2009.

WATRALL, E; SIARTO, J. Use A Cabeça! **Web Design.** Alta Books, 2009.