**1 Introdução**

O Sistema sistema de gestão condominial será desenvolvido para um condomínio composto por um prédio de 10 (dez) andares com 4 (quatro) apartamentos em cada andar, com os seguintes elementos:

1. Usuários do Sistema:

* Síndico
* Funcionários
* Condôminos

1. Permissões e Funcionalidades:

* Tipo A – Cadastro de moradores, visitantes, entrada de visitantes, reserva de espaços
* Tipo B – Reserva de espaços e alteração de cadastro

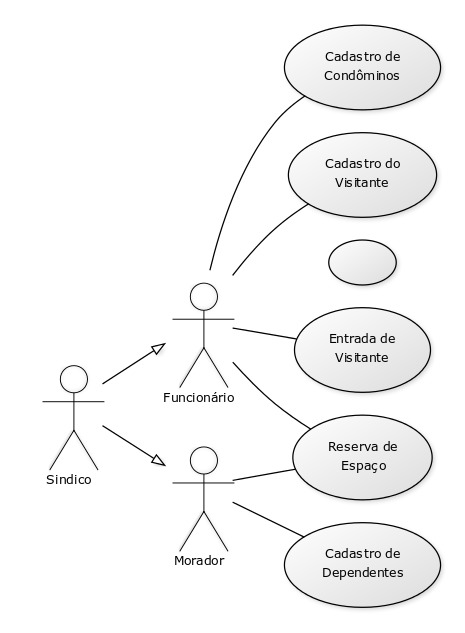
1. O acesso a essas funcionalidades será delimitado de acordo com o nível de permissão de cada usuário:

* Síndico e Funcionários – Tipo A
* Condôminos – Tipo B

Especificação de Caso de Uso

**Caso de Uso**

Diagrama de Caso de Uso – Visão geral dos atores e seus respectivos casos de uso



|  |  |
| --- | --- |
| Nome do caso de uso | Inclusão de condôminos |
| Tipo | Caso de uso geral |
| Atores Principais | Funcionário |
| Ator Secundário | Síndico |
| Resumo | Este caso de uso visa descrever as etapas de uma inclusão de condôminos |
| Pré-Condições | Ter login e senha do sistema |
| **Fluxo Principal** | |
| Ações do Ator | Ações do Sistema |
| 1. O usuário abre a página do sistema e escolhe a aba de login conforme o seu tipo cadastrado: morador, síndico ou funcionário |  |
|  | 2. Requisita o login e senha do usuário |
| 3. Informa o login e senha |  |
|  | 4. Libera o acesso |
| 5. Abre a opção condôminos e clica em inserir |  |
|  | 6. Mostra lista de condôminos e apartamentos cadastrados |
|  | 7. Grava no banco de dados os dados do condômino |
|  | 8. Envia um aviso em tela ao usuário que o cadastro foi realizado com sucesso |
| **Fluxo Alternativo** | |
| Ações do Ator | Ações do Sistema |
|  | 1. Se já existir um condômino cadastrado para aquele apartamento. |
| 2. Clica em alterar e faz a alteração dos dados do condômino. |  |

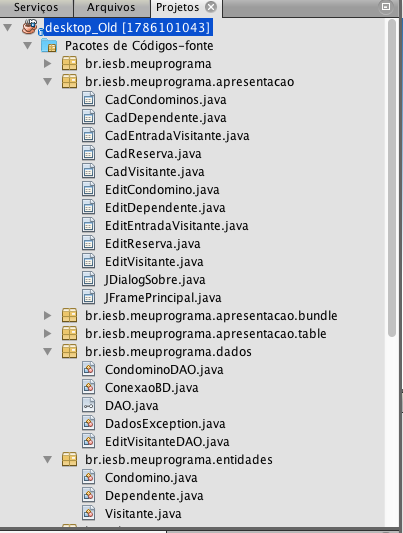
**2 Principais Funções do Sistema**

|  |  |  |
| --- | --- | --- |
| **Requisito** | **Tipo** | **Justificativa** |
| 001 – Efetuar Login | Funcional | Permitir ao usuário acessar o software com suas próprias configurações. |
| 002 – Manter Backup | Não Funcional | Realizar cópia de segurança dos dados do software e restaurar a mesma. |
| 003 – Atualizar estrutura de dados | Não Funcional | Executar atualização na estrutura do banco de dados referente ao software. |
| 004 – Manter os Dados Básicos dos Condôminos | Funcional | Armazenar os dados comuns e obrigatórios dos condôminos. |
| 005 – Manter Dados Básicos dos Usuários | Funcional | Armazena os dados básicos dos usuários do sistema |
| 019 – Manter Apartamento | Funcional | Armazenar os dados do apartamento. |

**3 Planejamento da Arquitetura e do Desenvolvimento do Sistema**

Com base na necessidade do cliente e utilizando os casos de uso para levantamento de requisitos, o sistema será desenvolvido em linguagem Java com framework Swing.  A IDE Java é referência em desempenho, segurança e principalmente grande portabilidade, podendo ser instalada em qualquer ambiente desktop, por exemplo, em Windows e Linux.

Utilizaremos o pattern DAO para configurar a camada de persistência com o banco de dados PostgresSQL 12.

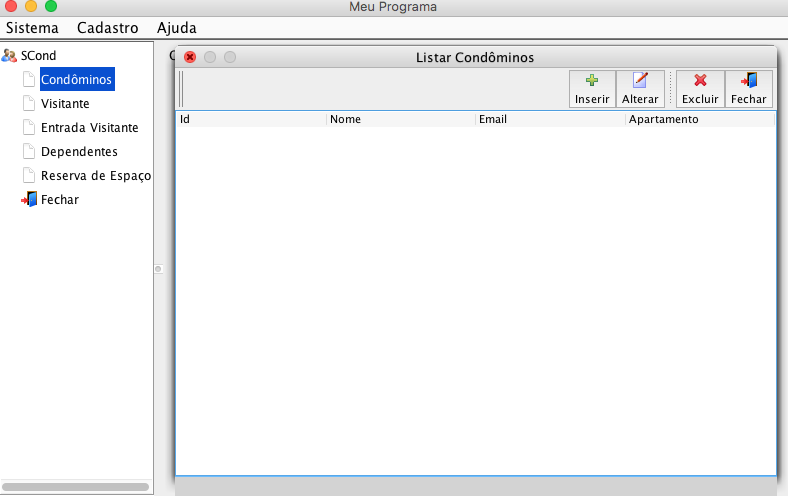


O sistema será desenvolvido para controlar apenas as rotinas administrativas do condomínio, não estando inclusas rotinas de outras naturezas, como por exemplo, geração de boletos etc.

A técnica de desenvolvimento será para Desktops, não serão incrementadas no projeto outras plataformas, como por exemplo, o ambiente Web.

**4 Planejamento dos Testes do Sistema**

**4.1 Requisitos a Testar**

****

**4.1.2 Teste do Banco de Dados**

* Verificar se as informações sobre condôminos, dependentes e visitantes podem ser inseridas ou modificadas do Banco de Dados
* Verificar se as informações obtidas no Banco de Dados consistem com as informações reais sobre condôminos, dependentes e visitantes.
* Verificar se as informações cadastradas podem ser consultadas.

**Técnica:**

Invocar cada método de acesso ao banco de dados, alimentando cada um com dados válidos e inválidos.

Inspecionar o banco de dados e verificar se os dados nas tabelas estão de acordo com as ações realizadas

**4.1.3 Teste Funcional**

* Verificar se o nível de acesso as funcionalidades do sistema a cada tipo de usuário estão corretas.

**Técnica:**

Executar cada caso de uso funcional através de seu fluxo principal e secundário, usando dados válidos e inválidos, para verificar o seguinte:

1. Os resultados esperados ocorrem quando dados válidos são usados.
2. As mensagens de erro ou aviso apropriadas são exibidas quando dados inválidos são usados.
3. Cada regra de negócio é aplicada apropriadamente.

**4.1.4 Teste do Ciclo de Negócios**

* Verificar se o tratamento de exceções está correto
* Verificar se os campos obrigatórios estão sendo preenchidos.
* Verificar se os campos estão sendo preenchidos com informações no formato correto.

**4.1.5 Teste de Interface do Usuário**

* Verificar se cada tela de interface gráfica pode ser facilmente entendida e utilizada.
* Verificar se os formulários de cadastro e edição estão pegando os dados inseridos pelo usuário corretamente.

**Técnica:**

Criar ou modificar os testes para cada janela para verificar a navegação e os estados de objeto apropriados para cada janela e objetos da aplicação.

Observar grupos de usuários usando a interface, analisando a taxa de aprendizado dos mesmos com o sistema e a aceitação da interface pelos usuários.

**4.1.6 Teste de Performance**

* Verificar o tempo de resposta de consultar/inserção/alteração no banco de dados.

**Técnica:**

Usar Procedimentos de Teste desenvolvidos para Teste da Função e Ciclo de Negócio.

**4.1.7 Teste de Instalação**

* Verificar se a instalação do sistema ocorre normalmente em todas as máquinas que possuam os requisitos mínimos.
* Verificar se a ferramenta possa ser instalada em diferentes ambientes (ex: Windows XP ou Windows Vista)
* Verificar se a atualização dos dados no servidor se reflete em todos os terminais.

**Técnica:**

Começar ou executar a instalação.

**4.1.8 Cronograma**

|  |  |  |
| --- | --- | --- |
| **Atividade** | **Inicio** | **Fim** |
| Planejamento de Testes | **27/5** | **1/6** |
| Projetar Testes | **27/5** | **1/6** |
| Implementar Testes | **2/6** | **10/6** |
| Execução de Testes | **11/6** | **14/6** |
| Avaliação de Testes | **15/6** | **23/6** |

**5 Planejamento da Implantação**

O sistema será instalado na máquina de cada usuário para que possam se familiarizar com a interface.

Todos os usuários receberão manuais do sistema e treinamentos para conhecer as funcionalidades

**6 Plano de Projeto**

Histórico da Revisão

|  |  |  |  |
| --- | --- | --- | --- |
| **Data** | **Versão** | **Descrição** | **Autor** |
| 23/6 | Release 1.0.0 | 1. Caso de uso Manter Condôminos 2. Caso de uso Manter Dependentes 3. Caso de uso Manter Visitante 4. Caso de uso Manter Entrada de Visitante 5. Caso de uso Manter Reserva | Grupo 4 |

|  |  |
| --- | --- |
| **Função** | **Profissionais** |
| Gerente do Projeto | * Carlos Rodrigues e Francisco Martins |
| Analista do Projeto | * Carlos Rodrigues e Túlio |
| Analista de Requisitos | * Anna Glória e Giovanna |
| Programador | * Anna Glória, Túlio, Francisco Martins |
| Teste | * Giovanna e Carlos Rodrigues |

**Cronograma**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ITENS** | **ATIVIDADE** | **INÍCIO** | **FIM** | **RESPONSAVEIS** |
| 1 | Definição de Requisito | **25/2** | **8/3** | Analista de Requisitos |
| 2 | Análise de Requisito | **11/3** | **13/3** | Analista de Requisitos |
| 3 | Projeto | **25/2** | **3/7** | Analista de Projeto |
| 4 | Protótipo | **14/3** | **15/3** | Programador |
| 5 | Codificação | **25/3** | **24/5** | Programador |
| 6 | Testes | **27/5** | **23/6** | Analista de Projeto / Analista de Requisitos |
| 7 | Integração de Testes | **25/3** | **28/6** | Analista de Projeto / Analista de Requisitos |