

Relatório do trabalho da disciplina de Programação Orientada a Objetos

Relatório de Gestão de Rendas/Imóveis

Pedro Carneiro - 26027

Carlos Brandão – 26025

Licenciatura em Engenharia de Sistemas Informáticos Pós-Laboral

Novembro de 2023

Relatório de Modelo de Negócio de um Site Imobiliário

Introdução

Este documento descreve o modelo operacional de um Gestor de Contratos de Arrendamento e de habitações arrendadas ou por arrendar, planeada por nós para apoiar este modelo de negócio. A nossa Aplicação será uma plataforma digital que irá facilitar a gestão de imóveis, para o Senhorio e a quem arrenda. A aplicação será construída em C#.

Modelo de Negócio

O objetivo da aplicação online de imóveis consiste em fornecer uma plataforma abrangente que simplifica o procedimento de aluguer de imóveis. Ele garantirá uma experiência prática e eficaz para todas as partes envolvidas no setor imobiliário, sobressaindo-se pelos seguintes aspetos:

- Do lado do utilizador: Existe para estes uma plataforma que permite ao mesmo criar uma conta na aplicação permitindo-o a alugar imóveis na plataforma.
- Do lado dos Senhorios: Existe para estes um sistema que lhes dá acesso à gerência dos seus imóveis, contratos, recibos, documentos, pagamentos e reviews acerca dos clientes que alugam as suas habitações.

Aplicação

A aplicação será desenvolvida em C#, uma linguagem de programação amplamente utilizada para o desenvolvimento de aplicações, devido à sua facilidade de integração com bases de dados. A base de dados será essencial para armazenar informações sobre propriedades, utilizadores e contratos.

Os principais objetivos do desenvolvimento da aplicação incluem:

- Usabilidade: Criar uma interface para o utilizador que seja intuitiva e amigável para os mesmos.
- Confiabilidade: Garantir que a base de dados do sistema seja de fácil utilização para os administradores, não criando redundância nos dados e evitando outras más práticas, fazendo assim com que os dados sejam confiáveis.
- Segurança: Implementar medidas de segurança rigorosas para proteger informações sensíveis e dados dos utilizadores.
- Escalabilidade: Projetar o sistema para ser escalável, de modo que possa lidar com um aumento dos dados na base de dados.

Desenvolvimento

1. Home.cs:

- **Descrição Principal:**
 - Classe Home que encapsula propriedades como Id, Type, Address e State.
 - Responsável por instâncias de casas e gestão de propriedades.

2. HomeOperations.cs:

- **Descrição Principal:**
 - Classe HomeOperations com métodos para manipular objetos da classe Home.
 - Inclui métodos para criação, adição, verificação de existência e exclusão de casas.

3. Landlord.cs:

- **Descrição Principal:**
 - Classe Landlord que herda de Person com propriedades, incluindo LandlordId.
 - Cria instâncias de senhorios e gere suas propriedades.

4. LandlordOperations.cs:

- **Descrição Principal:**
 - Classe LandlordOperations com métodos para manipular objetos da classe Landlord.
 - Métodos para criar, adicionar, verificar existência e excluir senhorios.

5. Person.cs:

- **Descrição Principal:**
 - Classe base Person com propriedades comuns como Name(Nome) e Nif.
 - Usada como base para outras classes.

6. PersonOperations.cs:

- **Descrição Principal:**
 - Contém métodos para validar o Número de Identificação Fiscal (NIF) de uma pessoa.

7. Contract.cs:

- **Descrição Principal:**
 - Classe Contract com propriedades relacionadas a contratos de aluguel.
 - Cria instâncias de contratos e gere suas propriedades.

8. ContractOperations.cs:

- **Descrição Principal:**
 - Classe ContractOperations com métodos para manipular objetos da classe Contract.
 - Inclui métodos para criação, adição, verificação de existência e listagem de contratos.

9. Receipt.cs:

- **Descrição Principal:**
 - Classe Receipt com propriedades relacionadas a recibos de pagamento.
 - Cria instâncias de recibos e gere suas propriedades.

10. ReceiptOperations.cs:

- **Descrição Principal:**
 - Classe ReceiptOperations com métodos para manipular objetos da classe Receipt.
 - Inclui métodos para criação, adição, verificação de existência e exclusão de recibos.

11. Review.cs (Fase2DLL):

- **Descrição Principal:**
 - Classe Review com propriedades relacionadas a avaliações de propriedades.
 - Cria instâncias de avaliações e gere suas propriedades.

12. ReviewOperations.cs (Fase2DLL):

- **Descrição Principal:**
 - Classe ReviewOperations com métodos para manipular objetos da classe Review.
 - Inclui métodos para criação, adição, verificação de existência, edição e exclusão de avaliações.

Criação de Classes

Em C#, classes são estruturas fundamentais para organizar e encapsular dados e funcionalidades relacionadas. Elas servem como modelos para criar objetos, promovendo a modularidade e reutilização de código. Classes desempenham papel vital na orientação a objetos, permitindo a criação de sistemas eficientes, escaláveis e de fácil manutenção.

Aqui estão alguns pontos chave em C# e a sua importância:

1. Modelagem de Objetos
2. Encapsulamento
3. Polimorfismo
4. Hierarquia

Requisitos

- Hierarquia de Classes:

O projeto utiliza herança para criar uma hierarquia de classes, como pode ser observado na classe Landlord que herda da classe Person. Isso exemplifica o conceito de herança, um dos pilares da orientação a objetos.

- Encapsulamento:

Todas as classes do projeto, como Home, Landlord, Contract, Receipt e Review, possuem propriedades privadas com métodos públicos para acessar e modificar essas propriedades. Isso reflete o encapsulamento, garantindo o controle sobre o acesso aos dados internos das classes.

- Polimorfismo:

Embora o exemplo fornecido não demonstre polimorfismo de forma explícita, a aplicação desse conceito pode ser explorada futuramente com a introdução de métodos sobrescritos ou interfaces, permitindo flexibilidade no uso de classes derivadas.

- Modularidade e Reutilização de Código:

O projeto é modular, com classes independentes para diferentes entidades, como Landlord, Contract, Receipt, etc. Cada classe possui métodos específicos para realizar operações relacionadas à sua responsabilidade. Isso promove a modularidade e a reutilização de código.

- Validação de Dados:

O código inclui um método na classe PersonOperations para validar o Número de Identificação Fiscal (NIF). Essa validação demonstra uma abordagem para garantir dados consistentes e válidos, evidenciando uma prática sólida de programação.

- Persistência de Dados:

Embora não seja explicitamente abordado no código fornecido, a estrutura do projeto permite a introdução de persistência de dados. Pode-se integrar um sistema de gerenciamento de banco de dados para armazenar e recuperar informações, atendendo assim à necessidade de persistência de dados

- Tratamento de Exceções:

O código inclui blocos de tratamento de exceções em algumas partes, como ao salvar os dados em arquivo na classe `ReviewOperations.cs`. Isso demonstra uma consideração para lidar com possíveis erros durante a execução do programa, contribuindo para a robustez do sistema.

Funcionalidades a implementar

1. Página Inicial:
 - Imóveis para Arrendamento, Acesso ao Login
2. Login:
 - Permite o acesso a clientes e aos senhorios.
3. Processo de Pagamentos:
 - Permite ao Clientes fazer o pagamento ao Senhorio.

Conclusão

Esta aplicação facilita o uso e gerenciamento de dados, proporciona praticidade para os senhorios e os residentes e facilita o aluguel e a gestão do imóvel ideal para os seus clientes.

Os pontos fundamentais são o processamento, organização e segurança de dados através de uma base de dados. Oferecemos uma interface que permite a apresentação, análise e interpretação dos dados de forma intuitiva.

Em resumo, nossa meta é construir uma base de dados segura e confiável para os senhorios e os residentes, tornando não apenas a busca, mas também as transações de Arrendamento de propriedades mais acessíveis e simplificadas, criando um ambiente onde se consegue gerir e arrendar propriedades.