

Gestão de um Condomínio

Relatório



Mestrado Integrado em Engenharia Informática e
Computação

Algoritmos e Estruturas de Dados

Grupo:

Adelaide Silva - 201406986@fe.up.pt

Beatriz Martins - up201502858@fe.up.pt

Maria Teresa Chaves - up201306842@fe.up.pt

Faculdade de Engenharia da Universidade do Porto
Rua Roberto Frias, sn, 4200-465 Porto, Portugal

8 de Novembro de 2015

Conteúdo

1. Gestão de um Condomínio	3
2. Solução Implementada	4
3. Diagrama UML	6
4. Casos de Utilização	7
5. Principais dificuldades encontradas	8

Gestão de um Condomínio

Uma empresa de condomínios gere um conjunto de habitações.

Os clientes são proprietários das habitações, que têm ao seu serviço uma empresa de condomínios. Este possui um leque de serviços que estão ao dispor de cada cliente (desde que estejam disponíveis).

A empresa tem um conjunto de empregados que trata de todos os serviços (pintura, canalização, limpeza) que cada cliente pede. Esta gestão também passa pela remoção e adição de clientes que passam e/ou deixam de fazer parte deste grupo de habitações.

A administração deste condomínio gere também o pagamento de mensalidades referente a cada habitação, de acordo com a sua área habitacional e algumas áreas de lazer que as ditas habitações possam ter ou não.

Desta forma o desenvolvimento deste trabalho envolverá uma serie de funções, que dará resposta as mais diversas questões que o gerenciamento das habitações obriga a fazê-lo.

Solução Implementada

Para a implementação desta aplicação, recorreremos à linguagem de programação c++, que sendo uma linguagem orientada a objetos, permite a criação de várias classes. Sendo um dos pontos essenciais para a resolução do problema proposto.

Deste modo, decidimos criar uma classe para cada uma das entidades constituintes da aplicação. No decorrer do trabalho, podemos concluir que precisaríamos de recorrer ao polimorfismo e herança de classes. Uma vez que várias classes partilhavam informações cruciais.

Decidimos que a aplicação teria duas classes principais, *Condominio* e *Cliente*. Visto que uma empresa de gestão de condomínios não existe sem clientes.

Para a classe *Cliente* criamos subclasse *Habitacao*. Deste modo, facilitaria a caracterização das habitações do cliente. Sendo que as habitações podem ser de dois tipos distintos, *Apartamento* e *Vivenda*.

A classe *Condominio* dispõe de uma subclasse *Servico* para o gerenciamento dos serviços prestados pelo condomínio aos clientes. Esta necessita de empregados (classe *Empregado*) divididos em mais 3 subclasses para diferenciar o tipo de empregados e serviços prestados (*Pintura*, *Canalização* e *Limpeza*).

Para a classe *Condominio*, criamos os seguintes membros-dado:

- Nome do condomínio
- NIF do condomínio
- Lista de clientes
- Serviços prestados

Para a classe *Cliente*, criamos os seguintes membros-dado:

- Nome do cliente
- N.º do bilhete de identidade do cliente
- Lista de habitações do cliente

Para a classe *Habitacao*, criamos os seguintes membros-dado:

- Morada da habitação
- Área da habitação

Para a classe *Apartamento*, acrescentamos ainda os seguintes membros-dado:

- Tipologia do apartamento
- Piso onde se encontra o apartamento

Para a classe *Vivenda*, acrescentamos ainda os seguintes membros-dado:

- Área exterior da vivenda
- Se possui piscina ou não

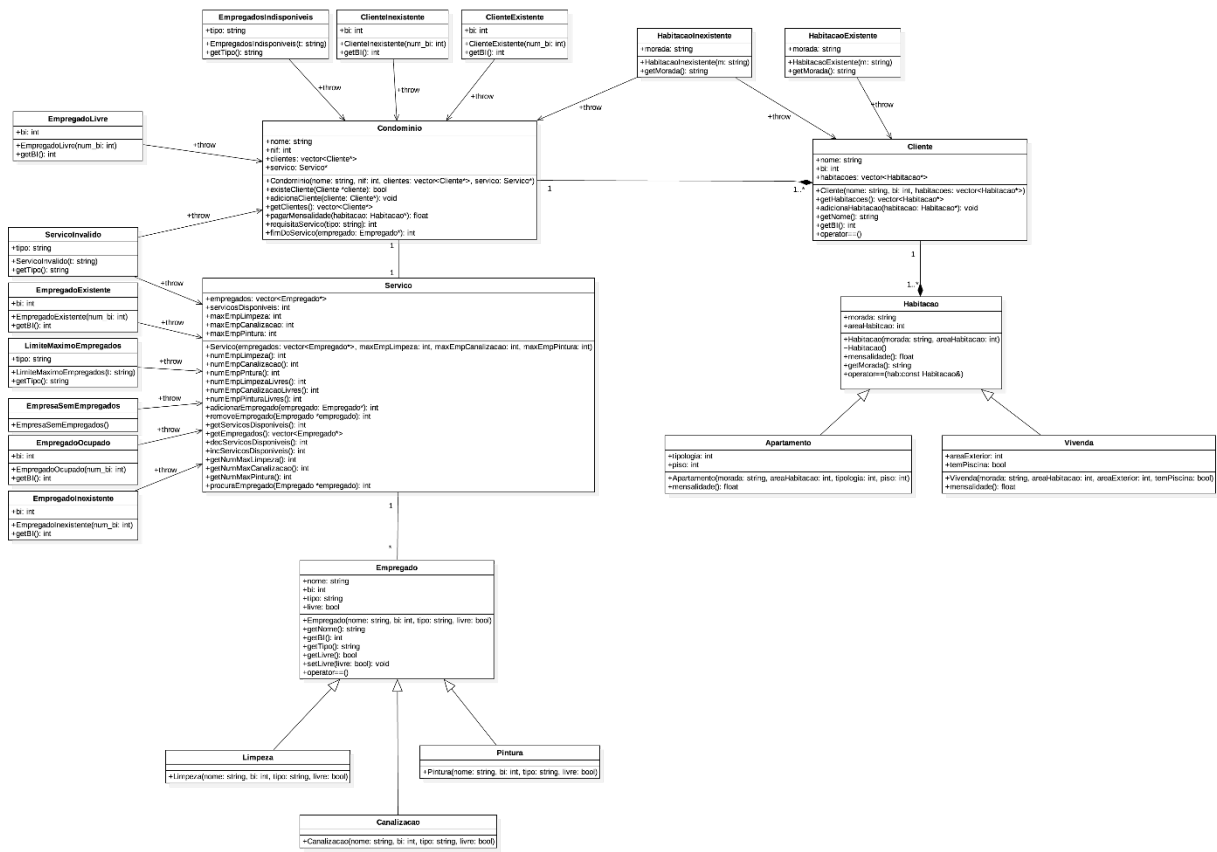
Para a classe *Servico*, criamos os seguintes membros-dado:

- Lista de empregados
- Serviços disponíveis
- Todos os empregados de limpeza, canalização e pintura

Para a classe *Empregado*, criamos os seguintes membros-dado:

- Nome do empregado
- N.º bilhete de identidade
- Tipo de serviço
- Se está livre ou não

Diagrama UML



Casos de Utilização

A aplicação tem essencialmente 3 casos de utilização:

1. Condomínio - é utilizado por funcionários da empresa de Gestão do Condomínio, que permite:
 - Adicionar cliente, caso não exista aquele cliente na lista;
 - Remover cliente, quando este já não faz possui nenhuma habitação que usufrui dos serviços deste condomínio;
 - Consultar a lista de Clientes;
 - Consultar dados dos clientes.

2. Cliente – é usado por clientes, que permite:
 - Pagar mensalidade, das suas habitações que possuem;
 - Adicionar habitação, no caso de não existir;
 - Remover habitação;
 - Requisitar serviço, ao condomínio para as suas habitações;
 - Listar habitações;
 - Consultar dados da habitação;
 - Listar histórico de serviço de habitação.

3. Empregado – é utilizado pelos empregado, que permite:
 - Terminar serviço;
 - Consultar histórico de serviços.

Principais dificuldades encontradas

A maior dificuldade residiu na implementação dos diversos serviços que o condomínio dispõe, uma vez que a aplicação apenas permite que um serviço seja requisitado se existem empregados livres desse tipo, atualizando os empregados disponíveis para prestar serviços.

Sentimos também alguma dificuldade na implementação de um sistema que permitisse ao utilizador aceder a um histórico de serviços prestados para uma dada habitação.

A utilização do Doxygen não foi possível dentro do IDE Eclipse. Sempre que tentávamos criar um ficheiro Doxygen aparecia uma mensagem de erro ("An error occurs while running doxygen. eclox.core.doxygen.RunException:"). Como solução a este problema, pela linha de comandos, executamos o Doxygen criando um ficheiro Doxygen dentro da pasta onde estávamos a trabalhar no Eclipse. Após execução destes comandos, já era permitido gerar a documentação no Eclipse.

Outra questão que também dificultou-nos, foi a relação entre as classes.