



FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

Gestão de Informação de Condomínios

Turma 6

- **Nuno Filipe Sousa e Silva** (up201404380)
up201404380@fe.up.pt
- **Paulo Sérgio Silva Babo** (up201404022)
up201404022@fe.up.pt
- **Pedro Miguel de Serpa Pinto Pereira Gomes** (up201303271)
up201303271@fe.up.pt

Índice

Descrição do Trabalho.....	3
Descrição da solução implementada.....	3
Diagrama UML.....	4
Lista de casos de utilização identificados para a aplicação.....	5
Principais dificuldades na realização do trabalho.....	6
Esforço dedicado por cada elemento do grupo.....	6

Descrição do Trabalho

Neste trabalho, o objetivo era criar uma aplicação em C++ que pudesse gerir condomínios.

Os condomínios, que temos de gerir, tem serviços aos quais a adesão é facultativa, habitações (vivendas e apartamentos) sendo que as vivendas podem ter piscina e têm área exterior, os apartamentos, não tem área exterior e piscina, mas são classificados por tipologia e piso. Ambos possuem área habitacional, mensalidade, morada e um condómino.

Tudo isto deve ser possível gerir usando a aplicação.

Descrição da solução implementada

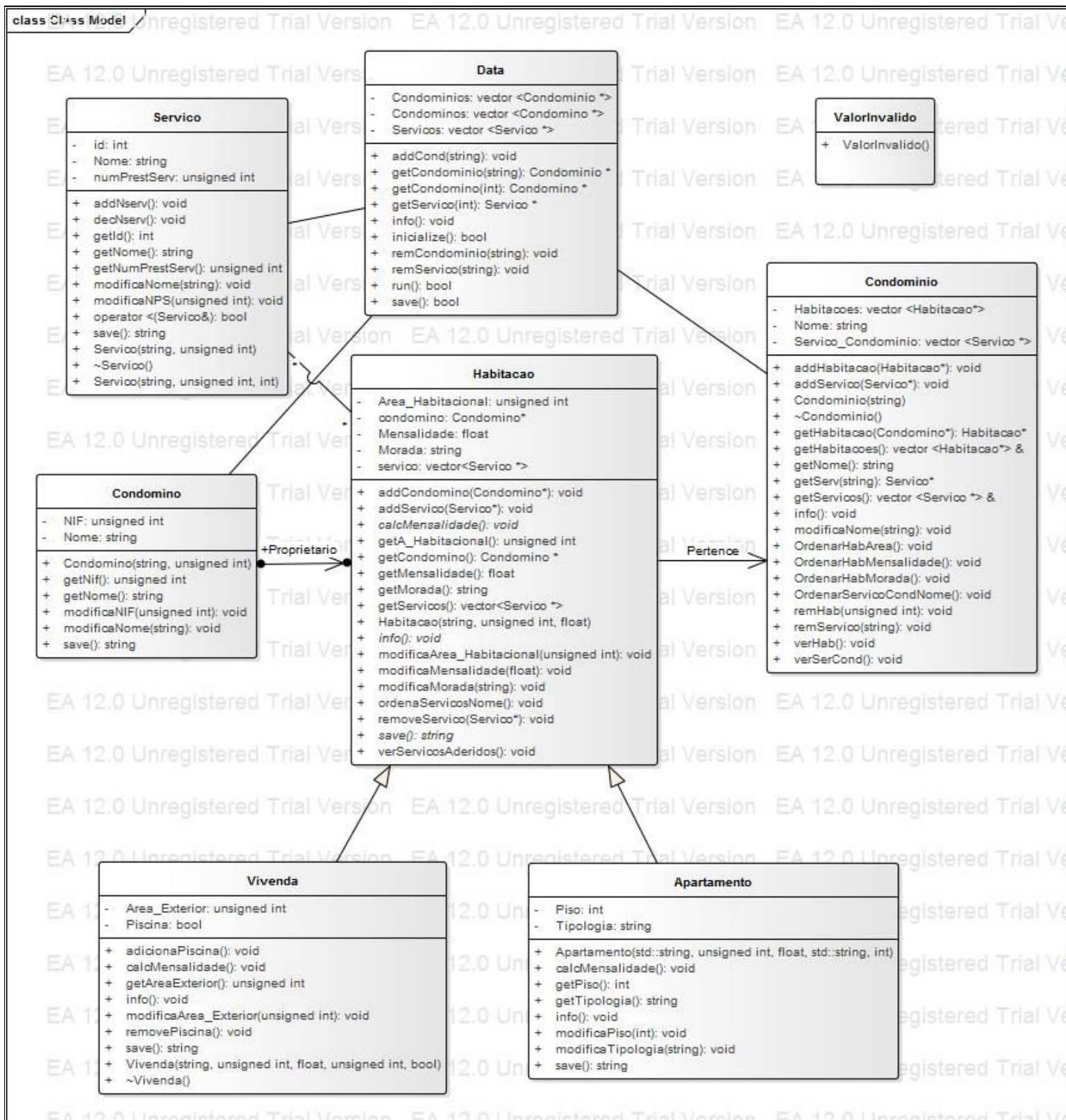
Um condomínio é formado por um conjunto de habitações e dispõe de serviços facultativos, como referido acima habitações podem ser Apartamentos ou Vivendas e cada uma das habitações tem um condómino.

Existe uma relação entre as diferentes partes que forma o condomínio como um todo, então usando classes (Herança de classes), vetores, ficheiros de texto (para guardar informações para uso futuro), conseguimos assim, criar um sistema onde a informação estivesse interligada e facilmente acessível.

Também usamos excepções para assinalar ações que não deviam de acontecer e todo o tipo de métodos, que permitem manipular a informação e também a visualizar, organizadas de várias maneiras.

O IDE usado para a realização da aplicação foi o Visual Studio 2013.

Diagrama UML



Lista de casos de utilização identificados para a aplicação

Para a demonstração de funcionalidades da aplicação foi criado um menu com várias opções.

No menu inicial o utilizador tem 5 opções.

Opção 0: Sair;

Opção 1: Mostra informação sobre o número de condomínios, serviços e condóminos e os seus nomes;

Opção 2: Gerir os condomínios, onde escolhe o condomínio que quer gerir e depois é lhe apresentado mais 9 opções:

- opção 0: Sair

- opção 1: Ver informação sobre Condomínio, Habitações, possibilita a visualização ordenada por três formas, por Área, Morada e Mensalidade;

- opção 2: Ver informação sobre os Serviços do condomínio (ordenados por nome);

- opção 3: Adicionar habitação;

- opção 4: Remover habitação;

- opção 5: Modificar habitação (pode modificar todos os atributos de uma habitação e fazer a habitação aderir ou terminar serviços);

- opção 6: Adicionar serviço ao condomínio;

- opção 7: Remover serviço do condomínio;

- opção 8: Modificar serviço (nome e numero de prestadores de serviço).

Opção 3: Adicionar condomínios.

Opção 4: Remover condomínios.

Principais dificuldades na realização do trabalho

Horários, residências diferentes e alguns acontecimentos que se passaram ao longo das semanas, impossibilitou o encontro dos três elementos, mas fomos nos mantendo organizados e em contato com a ajuda de várias ferramentas como o google drive, repositório SVN.

O Pedro só fez a função `save()` perto da data de entrega e não tinha testado, depois do Nuno Silva corrigir os pequenos erros (como a falta do `return`, da declaração de algumas variáveis usadas na função, etc) o mesmo testou a função e reparou que a função não funcionava corretamente (Várias vezes foi dito para ir fazendo com tempo e testando o que fazemos). Sendo que publicação da função `save()` por parte de Pedro não foi com antecedência e não tinha comentários, os outros dois membros, Paulo e Nuno não conseguiram corrigir a função a tempo, o que significa que a gravação da informação nos ficheiros de texto só funciona parcialmente.

Esforço dedicado por cada elemento do grupo

O Nuno Silva organizou grande parte do trabalho e teve o maior contributo na realização do mesmo, pois contribui-a semanalmente para o desenvolvimento do trabalho, fez o relatório, à excepção do diagrama UML, e todo o código à excepção do código desenvolvido pelos outros membros que é referido de seguida.

O Paulo Babo fez todas as funções de organizar os vetores, do código que trata das excepções e a documentação usando o Doxygen.

O Pedro Gomes fez o diagrama UML e a classe “data” à excepção das funções:

“`run()`”, “`getCondominio(string nome)`”, “`info()`”, “`remCondominio(string nome)`”, “`remServico(string nome)`” e “`addCond(string nome)`”, não terminou corretamente a função “`save()`”