



**Licenciatura – Engenharia Eletrotécnica e de
Computadores**

Programação Orientada a Objetos

Trabalho Prático

Sistema de Aluguer de Automóveis

Instituto Politécnico do Cávado e do Ave

Rui Ribeiro – 16419
Tiago Carvalho - 18601
João Sampaio – 18611
David Gonçalves – 18612

Resumo

No âmbito da unidade curricular de programação orientada a objetos foi proposta a realização de um trabalho prático com objetivo de consolidar os conteúdos abordados ao longo da cadeira.

No presente relatório abordamos todo o processo de desenvolvimento da aplicação utilizando o software QT Creator e a linguagem C++, utilizando o paradigma de programação orientada a objetos.

A aplicação consiste num sistema de aluguer de automóveis, permitindo a um cliente alugar o automóvel que quiser conforme a disponibilidade, e a um administrador fazer a gestão do sistema, atualizando a lista de veículos disponíveis.

A distinção entre cliente e administrador é realizada através de um sistema de login.

O programa possui também uma interface gráfica realizada através de widgets no software Qt Creator.

Índice

<i>Introdução</i>	6
<i>1. Métodos</i>	7
1.1 Arquitetura da solução	8
1.1.1 Base de dados:	8
1.1.2 Janelas.....	10
1.1.2.1 MainWindow (janela inicial/login):	10
1.1.2.2 SecondWindow(Menu):	11
1.1.2.3 ShowCarWindow (lista de veículos).....	12
1.1.2.4 AddCarWindow (adicionar um veículo):	13
1.1.2.5 RemoveCarWindow (adicionar um veículo):.....	14
1.1.2.6 RentCarWindow (alugar um veículo):	15
1.1.2.7 ClientSpaceWindow (alugar um veículo):	17
1.2 Estratégias seguidas	18
1.2.1 Comunicação entre janelas (cast):	18
1.2.2 Organização do programa (por janelas):	19
1.2.3 Organização dos ficheiros .txt:	20
<i>2. Resultados obtidos</i>	21
2.1. Exemplo de adição de um veículo:	21
2.2. Exemplo de aluguer com o cliente Rui:	22
<i>Conclusão</i>	24
<i>Bibliografia</i>	25

Índice de figuras

<i>Figura 1 – Ficheiro LoginData.txt.....</i>	<i>8</i>
<i>Figura 2 – Ficheiro CarrosData.txt.....</i>	<i>9</i>
<i>Figura 3 - Ficheiro RentData.txt.....</i>	<i>9</i>
<i>Figura 4 – Chamada á função membro verify().</i>	<i>10</i>
<i>Figura 5 – Realização de um cast para confirmar o nível de acesso</i>	<i>11</i>
<i>Figura 6 – Diferenciação entre os níveis de acesso</i>	<i>11</i>
<i>Figura 7 – Lista de veículos na base de dados.....</i>	<i>12</i>
<i>Figura 8 – Adição de um veículo</i>	<i>13</i>
<i>Figura 9 – Cast para aceder ao multimapa e adicionar o veículo</i>	<i>13</i>
<i>Figura 10 – Remoção de um veículo.....</i>	<i>14</i>
<i>Figura 11- Dois casts para encontrar o veículo e remover.....</i>	<i>14</i>
<i>Figura 12 – Aluguer de um veículo.....</i>	<i>15</i>
<i>Figura 13 – Cast a janelas que não são pai da janela atual.....</i>	<i>16</i>
<i>Figura 14 - Acesso á matrícula na MainWindow.....</i>	<i>16</i>
<i>Figura 15 – Espaço cliente modo cliente.....</i>	<i>17</i>
<i>Figura 16 – Espaço Cliente modo administrador.....</i>	<i>17</i>
<i>Figura 17 – Exemplo de um cast ao multimapa MapaCarros</i>	<i>18</i>
<i>Figura 18 – Mudança de janela</i>	<i>18</i>
<i>Figura 19 – Casts a janelas que não são pai da janela atual.....</i>	<i>18</i>
<i>Figura 20 – Diferenciação conforme níveis de acesso</i>	<i>20</i>
<i>Figura 21- Lista de veículos sem o novo carro</i>	<i>21</i>
<i>Figura 22 - Lista de veículos com o novo carro</i>	<i>21</i>
<i>Figura 23 – Espaço cliente do cliente Rui.....</i>	<i>22</i>
<i>Figura 24 – Sistema de aluguer.....</i>	<i>22</i>
<i>Figura 25 – Espaço cliente do cliente Rui com o novo veículo</i>	<i>23</i>

Introdução

O trabalho consiste num sistema de aluguer de automóveis, recorrendo a um sistema de Login de forma a realizar a distinção entre cliente e administrador.

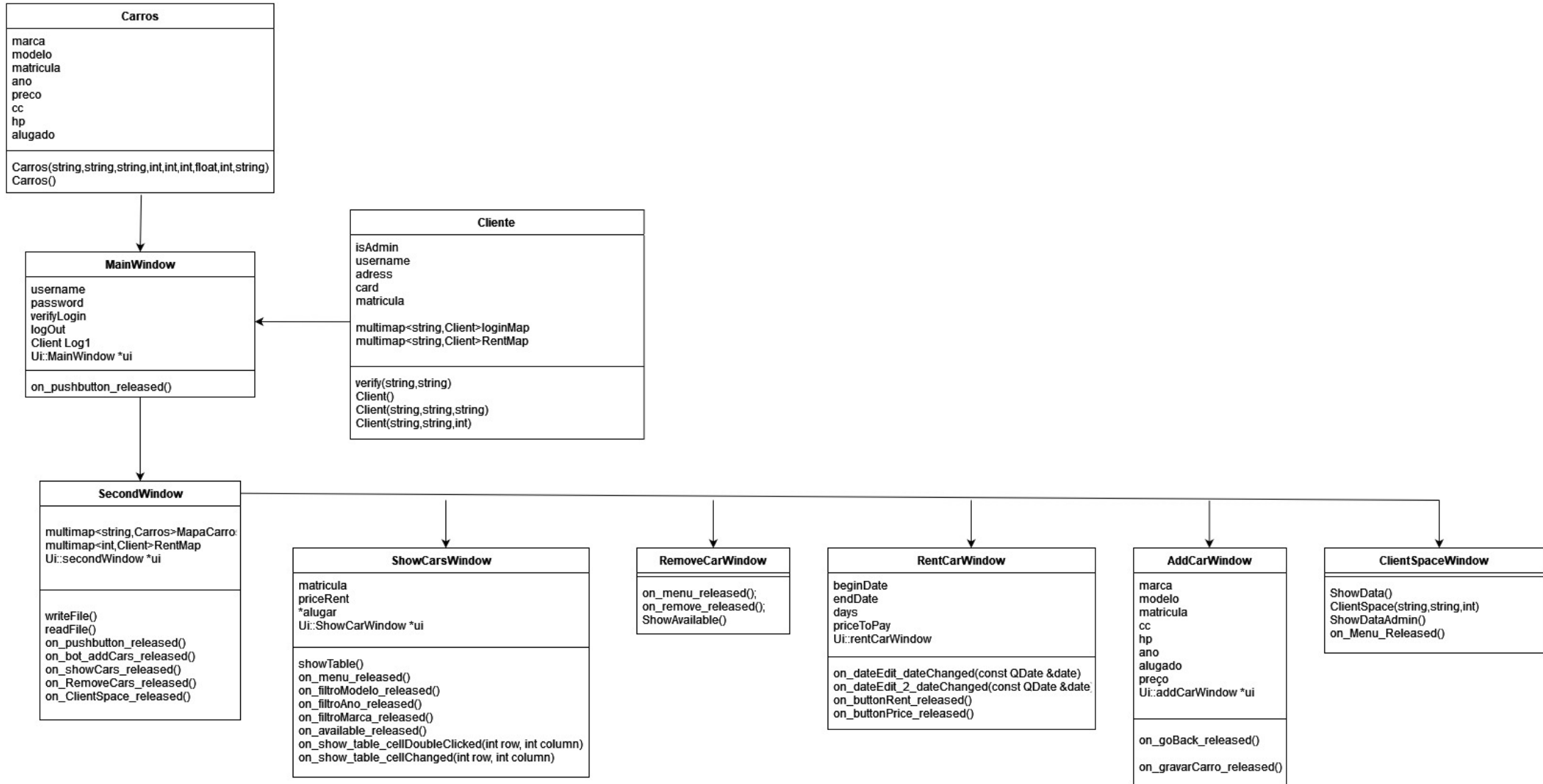
Essa distinção é um dos elementos cruciais do sistema, visto que apenas o administrador poderá realizar as seguintes operações:

- Adição de um veículo;
- Remoção de um veículo existente;
- Editar as informações de um veículo;

O cliente apenas terá acesso ao aluguer e ao seu espaço de cliente, onde consulta as suas informações e todos os veículos que tem alugados.

De forma a manter todos os veículos e os seus dados, é necessário criar uma base de dados, permitindo assim realizar todas as operações supracitadas.

1. Métodos



1.1 Arquitetura da solução

Inicialmente foi realizado um diagrama de classes, de forma a clarificar a estrutura do programa, possibilitando então a sua realização (página anterior).

Subsequentemente iniciamos a realização do programa, conforme o diagrama, criando duas classes distintas, Carros e Cliente, onde existem todas as informações de um automóvel e o sistema de login, respetivamente.

Estas classes servem de base a todo o programa.

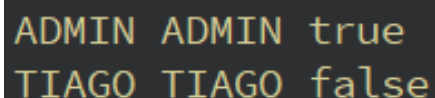
1.1.1 Base de dados:

O sistema contém uma base de dados, construída de forma simples, através de ficheiros de texto .txt.

Na classe Cliente existe um multimapa onde será carregado todo o conteúdo do ficheiro LoginData.txt. Neste ficheiro estão guardados todos os utilizadores que têm acesso a este programa, inclusive o seu nível de acesso, administrador ou cliente (true – administrador / false – cliente).

O ficheiro está organizado por nome do utilizador, palavra-passe e nível de acesso, respetivamente (como representado na figura seguinte).

O multimapa que recebe os dados do ficheiro, tem como chave o nome de utilizador.



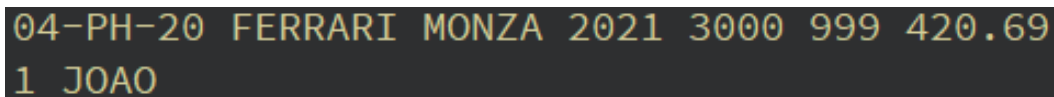
```
ADMIN ADMIN true
TIAGO TIAGO false
```

Figura 1 – Ficheiro LoginData.txt

Na segunda janela do programa, existem um multimapa onde será carregado todo o conteúdo do ficheiro CarrosData.txt. Nesse ficheiro estão guardados todos os veículos disponíveis e as informações deles, possibilitando depois realizar todas as operações necessárias.

O ficheiro está organizado pela matrícula, marca, modelo, ano, cilindrada, potência, preço do aluguer diário, o estado do veículo (alugado – 1 / disponível – 0) e a quem está alugado (nome do cliente ou “none” para não alugado), respetivamente.

O multimapa tem como chave a matrícula, visto que é única para cada veículo, permitindo uma fácil distinção aquando da realização das operações.



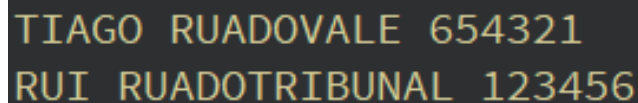
04-PH-20 FERRARI MONZA 2021 3000 999 420.69
1 JOAO

Figura 2 – Ficheiro CarrosData.txt

Juntamente com o ficheiro de login que é carregado para o multimapa na primeira janela, também as informações do cliente serão carregadas do ficheiro RentData.txt para um multimapa.

As informações consistem no nome do cliente, morada e número do cartão, sendo apresentadas no espaço do cliente.

Este ficheiro é essencial para permitir que os veículos alugados sejam associados aos respetivos clientes que realizaram o aluguer.



TIAGO RUADOVALE 654321
RUI RUADOTRIBUNAL 123456

Figura 3 - Ficheiro RentData.txt

1.1.2 Janelas

1.1.2.1 MainWindow (janela inicial/login):

Na seguinte janela é realizado o processo de login no sistema.

Através de dois widgets (QLineEdit), o utilizador colocará o seu nome de utilizador e palavra-passe, carregando de seguida no botão para realizar a confirmação do login. Um objeto criado como variável membro dessa janela, tipo Client, é usado para chamar a função membro verify() com os parâmetros de entrada correspondentes ao que foi inserido pelo utilizador.

O construtor da classe Client, inicialmente irá carregar todo o conteúdo de dois ficheiros diferentes para os respetivos multimaps, um que é correspondente ao processo de login e outro correspondente às informações do cliente (que não faz parte do processo do login).

Após chamar a função e enviar os parâmetros, esta função irá fazer a verificação do que foi inserido pelo utilizador, com o conteúdo da base de dados, e assim permitir ou não o acesso ao sistema.

Se o login for aceite, passa para a nova janela escondendo a MainWindow.

A verificação entre clientes e administradores é realizada nas janelas seguintes.

```
verifyLogin = Log1.Verify(username,password);
```

Figura 4 – Chamada á função membro verify().

1.1.2.2 SecondWindow(Menu):

A segunda janela trata-se do Menu do sistema, onde poderemos aceder a cada função do programa.

Inicialmente é realizada uma verificação do utilizador que está a utilizar o sistema, ou seja, se é administrador ou cliente. Essa verificação é feita através de uma operação chamada cast á janela pai desta segunda janela, ou seja, a janela que a originou (MainWindow).

```
//Verifica de o acesso é de cliente ou admin
if(((MainWindow*)(QWidget*)this->parent()))->verifyLogin == "client"){
    //caso seja cliente as janleas de adicionar e remover nao sao mostra
    status = "client"; //Variavel auxiliar para ser usada em outra janela
    ui->bot_addCars->hide(); //Esconder o botao para adicionar carros
    ui->bot_removeCars->hide(); //Esconder o botao para remover carros
}
```

Figura 5 – Realização de um cast para confirmar o nível de acesso

Se for administrador, terá acesso a todas as janelas correspondentes às operações no sistema, e se for cliente, apenas será capaz de visualizar a lista de clientes (sem permissões de modificação das informações dos veículos), podendo alugar qualquer dos veículos disponíveis, e também terá acesso ao espaço do cliente, onde constam as suas informações e os veículos que tem alugados.

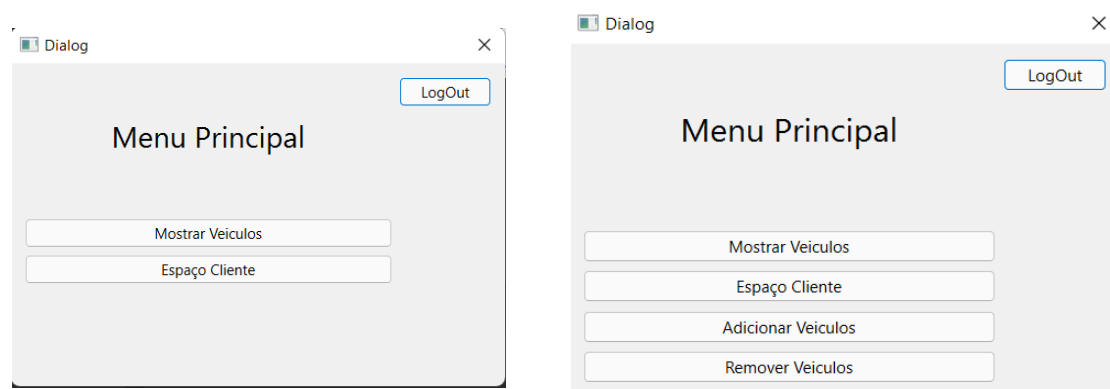


Figura 6 – Diferenciação entre os níveis de acesso

De forma a simplificar o programa, optamos por esconder certos botões, conforme as operações que um cliente não tem permissão de fazer, evitando assim a criação de diferentes janelas para os tipos de utilizadores diferentes.

1.1.2.3 ShowCarWindow (lista de veículos)

Ao pressionar o botão Mostrar Veiculos, vamos aceder a uma nova janela, filha da SecondWindow.

Esta janela apresenta todos os veículos disponíveis no sistema num widget (QTableWidget).

Sempre que uma operação é realizada, ou seja, se adicionarmos ou removermos um carro ou modificarmos as informações do veículo (modo administrador), a lista será atualizada durante a execução do programa.

É possível filtrar a lista de carros por marca, modelo, ano ou disponibilidade, escrevendo nos widgets (QLineEdit) á direita e carregando no botão para confirmar.

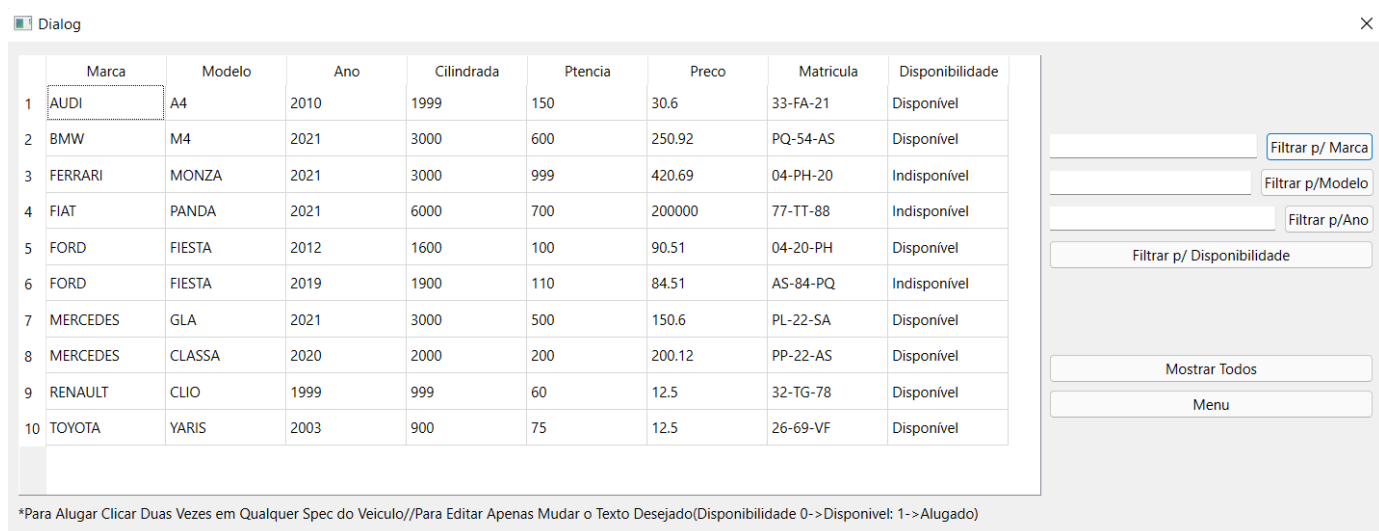


Figura 7 – Lista de veículos na base de dados

1.1.2.4 AddCarWindow (adicionar um veículo):

Ao pressionar o botão Adicionar Veículos, vamos aceder a uma nova janela, filha da SecondWindow.

Nesta janela o administrador é capaz de adicionar um novo veículo na base de dados, especificando as suas características.

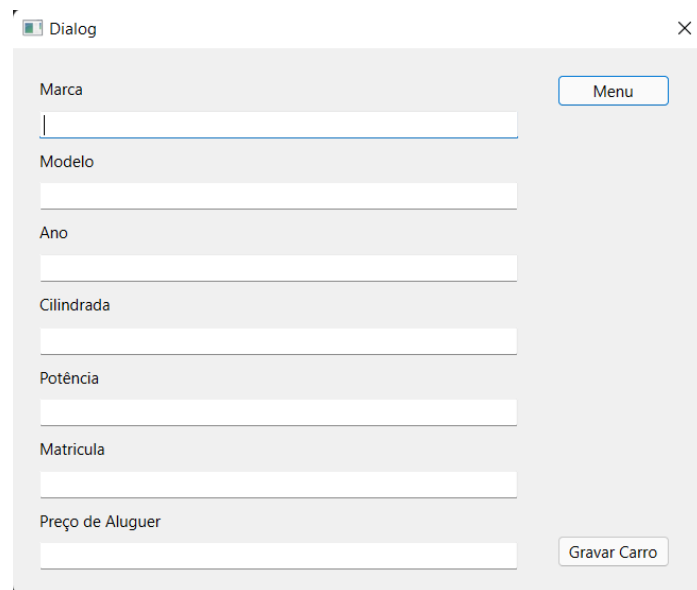
A screenshot of a Qt-style dialog box titled "Dialog". It features a light gray background and a standard window title bar with a close button. The dialog contains seven text input fields stacked vertically, each with a label to its left: "Marca", "Modelo", "Ano", "Cilindrada", "Potência", "Matrícula", and "Preço de Aluguer". To the right of the "Marca" field is a button labeled "Menu". To the right of the "Preço de Aluguer" field is a button labeled "Gravar Carro".

Figura 8 – Adição de um veículo

Quando terminar e pressionar o botão gravar carro, um cast será feito à janela pai (SecondWindow), de modo a aceder ao multimapa e adicionar o veículo agora criado.

Por predefinição, ao criar um veículo o carro será dado como disponível, não estando atribuído a nenhum cliente.

```
// Cast e Colocar os dados no Mapa
((secondWindow*)((QWidget*)this->parent()))->MapaCarros.insert(make_pair(matricula,Carros(matricula,
```

Figura 9 – Cast para aceder ao multimapa e adicionar o veículo

1.1.2.5 RemoveCarWindow (adicionar um veículo):

Ao pressionar o botão Remover Veiculos, vamos aceder a uma nova janela, filha da SecondWindow.

Nesta janela é apresentada um widget (QTableWidget) com todos os carros disponíveis, ou seja, não alugados a nenhum cliente.

A partir do widget abaixo da tabela (QLineEdit) o administrador é capaz de seleccionar o veículo a remover, digitando a matrícula correspondente.

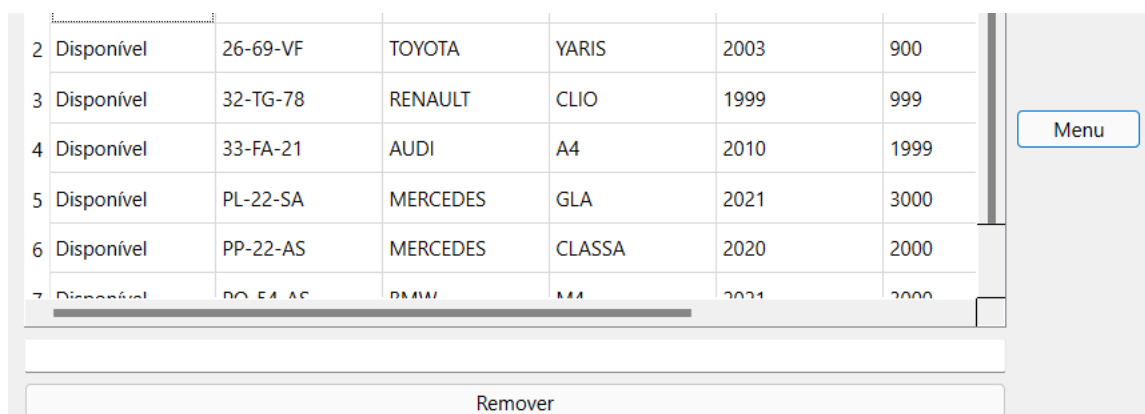


Figura 10 – Remoção de um veículo

Após pressionar o botão remover, o programa emitirá um sinal que vai aceder á respetiva função.

Para então poder remover o veículo, foi necessária a criação de um iterador, que irá iterar por todo o mapa que contém a lista de veículos.

Para aceder a essa lista, mais uma vez fazemos um cast ao mapa que se encontra na janela pai (SecondWindow), e através da função find(), o programa procura pela respetiva matrícula que foi dada como parâmetro.

Para apagar essa matrícula procedemos a outro cast usando a função erase() com o iterador como parâmetro, permitindo apagar o veículo desejado. As funções find() e erase () fazem parte das livrarias do C++.

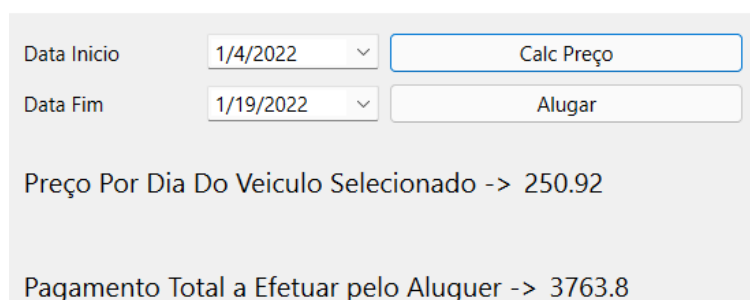
```
auto it = ((secondWindow*)((QWidget*)this->parent()))->MapaCarros.find(remover);  
((secondWindow*)((QWidget*)this->parent()))->MapaCarros.erase(it); //Apaga todos
```

Figura 11- Dois casts para encontrar o veículo e remover

1.1.2.6 RentCarWindow (alugar um veículo):

Através da janela ShowCarWindow, se o utilizador for um cliente, é possível aceder à janela RentCarWindow, carregando duas vezes em cima do veículo desejado.

Ao carregar, a janela é aberta e o cliente pode escolher até quando reservar, sendo possível fazer um cálculo do preço conforme os dias seleccionados nos widgets (QDateEdit) (selecção da data).



The screenshot displays a user interface for renting a car. It features two date selection widgets: 'Data Inicio' (Start Date) set to '1/4/2022' and 'Data Fim' (End Date) set to '1/19/2022'. To the right of these is a 'Calc Preço' (Calculate Price) button. Below the date fields is an 'Alugar' (Rent) button. The interface also shows the calculated price: 'Preço Por Dia Do Veiculo Seleccionado -> 250.92' and the total payment: 'Pagamento Total a Efetuar pelo Aluguer -> 3763.8'.

Figura 12 – Aluguer de um veículo

Ao carregar no botão alugar para confirmar o pedido, o programa dá a possibilidade de cancelar ou continuar para o aluguer.

Em caso afirmativo, o carro será dado como indisponível no sistema, e será atribuído ao veículo o respetivo dono do aluguer, conforme o utilizador no momento (o multimapa é modificado).

Esta janela tem uma particularidade, visto que para aceder ao estado do veículo e fazer a atribuição dele a um cliente, é necessário aceder a dados que não estão na janela pai (ShowCarsWindow).

Através de dois casts no início da imagem seguinte, o programa é capaz de aceder aos dados do avô (pai de ShowCarsWindow, ou seja, SecondWindow), e do bisavô (pai de SecondWindow, ou seja, MainWindow), de forma a obter as informações do cliente que está a alugar e do estado do veículo selecionado, atribuindo assim o estado indisponível e que cliente alugou, respetivamente.

Se o utilizador for um administrador, não tem acesso ao processo de aluguer, no entanto pode mudar o estado do veículo, modificando as informações.

```
QWidget *grandfather = ((QWidget*)this->parent()); //Cast para a secondWindow
QWidget *ggrandfather = ((QWidget*)grandfather->parent()); //Cast para a MainWindow
string password = ((MainWindow*)((QWidget*)ggrandfather->parent()))->password;
string matricula = ((ShowCarsWindow*)((QWidget*)this->parent()))->matricula;
```

Figura 13 – Cast a janelas que não são pai da janela atual

```
auto it = ((secondWindow*)((QWidget*)grandfather->parent()))->MapaCarros.find(matricula);
it->second.alugado = 1; //Estado do carro passa a alugado
//Buscar o carro que foi alugado para poder atribuir um cliente ao carro
it->second.rentedTo = password; //Atribuicao do cliente ao carro
```

Figura 14 - Acesso á matrícula na MainWindow

1.1.2.7 ClientSpaceWindow (alugar um veículo):

Ao pressionar o botão Espaço Cliente, vamos aceder a uma nova janela, filha da SecondWindow.

Esta janela está disponível para administradores ou clientes, apenas diferindo no que será apresentado conforme o nível de acesso.

Se o utilizador for um cliente, nesta janela constará uma QWidget com todos os veículos que foram alugados pelo respetivo cliente, incluindo também a sua informação.



Figura 15 – Espaço cliente modo cliente

Se o utilizador for um administrador, na janela constará uma QWidget com todos os carros alugados, indicando também o cliente que alugou.



Figura 16 – Espaço Cliente modo administrador

1.2 Estratégias seguidas

1.2.1 Comunicação entre janelas (cast):

O cast foi crucial para o desenvolvimento do trabalho, permitindo aceder a elementos das páginas principais, através das páginas secundárias. Este procedimento permitiu realizar todas as operações necessárias, sem necessidade de transferir “manualmente” a informação entre as páginas.

```
// Cast e Colocar os dados no Mapa
((secondWindow*)((QWidget*)this->parent()))->MapaCarros.insert(make_pair(matricula,Carros
```

Figura 17 – Exemplo de um cast ao multimapa MapaCarros

O cast permite aceder diretamente á página pai da janela atual. Para tal efeito, na ativação de um sinal (carregar num botão), através de um apontador, enviamos como parâmetro this, ou seja, a janela que irá desaparecer para dar lugar á nova janela. A janela que irá desaparecer será pai da janela que foi ativada, e desta forma, vai herdar todos os elementos constituintes da janela pai, podendo ter acesso aos seus dados.

```
addCarWindow* new_window = new addCarWindow(this);
new_window->show(); // como é apontador tenho de usar a seta
this->hide(); // Esconder a secondwindow.
```

Figura 18 – Mudança de janela

Através deste método, fomos capazes de implementar em poucas linhas de código, qualquer das operações desejadas.

Apenas na janela RentCarWindow foi necessário fazer cast mais complexos, de modo a aceder a outras janelas que não eram pais da janela atual, ou seja, não existia acesso ao conteúdo delas através dessa janela.

```
QWidget *grandfather = ((QWidget*)this->parent()); //Cast para a secondWindow
QWidget *ggrandfather = ((QWidget*)grandfather->parent()); //Cast para a Mainwindow
string password = ((MainWindow*)((QWidget*)ggrandfather->parent()))->password;
string matricula = ((ShowCarsWindow*)((QWidget*)this->parent()))->matricula;
```

Figura 19 – Casts a janelas que não são pai da janela atual

1.2.2 Organização do programa (por janelas):

O programa foi organizado inteiramente por janelas, tendo apenas duas classes base. Client, que tem as informações do cliente e onde é realizada a distinção entre Cliente e Admin e o sistema de login, e Carros, que contem toda a informação dos veículos.

Inicialmente existe a MainWindow, onde são carregados dois ficheiros para os respetivos mapas, contendo as informações de login e dos clientes.

Após essa janela temos a SecondWindow, que contém toda a base de dados dos veículos.

Realizando esta disposição, fomos capazes de simplificar muito o programa, visto que praticamente todas as janelas seguintes, herdaram como pai, a janela SecondWindow, facilitando a implementação das operações a serem realizadas.

De modo a aceder a qualquer valor, com o objetivo de modificar uma variável ou modificar o conteúdo de um mapa, facilmente o faríamos realizando uma operação cast.

1.2.3 Organização dos ficheiros .txt:

Através de uma organização inteligente da informação guardada nos ficheiros de texto, o grupo foi capaz de simplificar a implementação das operações.

Como por exemplo, a disposição do ficheiro LoginData de modo a poder realizar a distinção entre níveis de acesso ao programa.

Assim facilmente, utilizando um variável membro da classe Client, e uma condição, modificamos o programa de modo a restringir certas operações para quando o utilizador se trata de um cliente.

Para restringir essas operações, optamos por esconder os botões de modo ao cliente não ter acesso. E às janelas compartilhadas, tanto pelo cliente como pelo administrador, restringimos certas operações, como por exemplo, na lista de veículos o cliente não terá permissão para modificar os valores na base de dados, e um administrador não tem acesso á janela de aluguer, visto que não se trata de um cliente.

```
//Verifica de o acesso é de cliente ou admin
if(((MainWindow*)((QWidget*)this->parent()))->verifyLogin == "client"){
    //caso seja cliente as janleas de adicionar e remover nao sao mostradas ao cliente
    status = "client";//Variavel auxiliar para ser usada em outra janela
    ui->bot_addCars->hide(); //Esconder o botao para adicionar carros
    ui->bot_removeCars->hide();//Esconder o botao para remover carros
}
```

Figura 20 – Diferenciação conforme níveis de acesso

2.Resultados obtidos

2.1. Exemplo de adição de um veículo:

A seguinte lista de veículos apresentada na janela ShowCarWindow, é organizada por ordem crescente, conforme a marca.

9	RENAULT	CLIO	1999	999	60	12.5	32-TG-78	Indisponível
10	TOYOTA	YARIS	2003	900	75	12.5	26-69-VF	Indisponível

Figura 21- Lista de veículos sem o novo carro

Através da janela AddCarWindow, um administrador é capaz de adicionar um novo veículo á base de dados.

Após criado o veículo, podemos confirmar novamente a janela ShowCarWindow para verificar a lista de carros atualizada.

O novo veículo adicionado foi o Tesla, disponível para alugar.

9	RENAULT	CLIO	1999	999	60	12.5	32-TG-78	Indisponível
10	TESLA	MODELS	2021	1999	200	56.4	45-TY-89	Disponível
11	TOYOTA	YARIS	2003	900	75	12.5	26-69-VF	Indisponível

Figura 22 - Lista de veículos com o novo carro

2.2. Exemplo de aluguer com o cliente Rui:

O cliente Rui de momento apresenta um veículo alugado.

De forma a alugar outros veículos, deverá aceder à janela ShowCarWindow, através do Menu carregando em Mostrar Veículos.



The screenshot shows a 'Dialog' window with a table of client information and a summary on the right. The table has columns: Cliente, Marca, Modelo, Ano, Cilindrada, Potencia, and Matricula. The first row shows: 1, RUI, FIAT, PANDA, 2021, 6000, 700, 77-TT-88. To the right of the table, the following information is displayed: Nome : RUI, Cartão : 123456, and Morada : RUADOTRIBUNAL.

	Cliente	Marca	Modelo	Ano	Cilindrada	Potencia	Matricula
1	RUI	FIAT	PANDA	2021	6000	700	77-TT-88

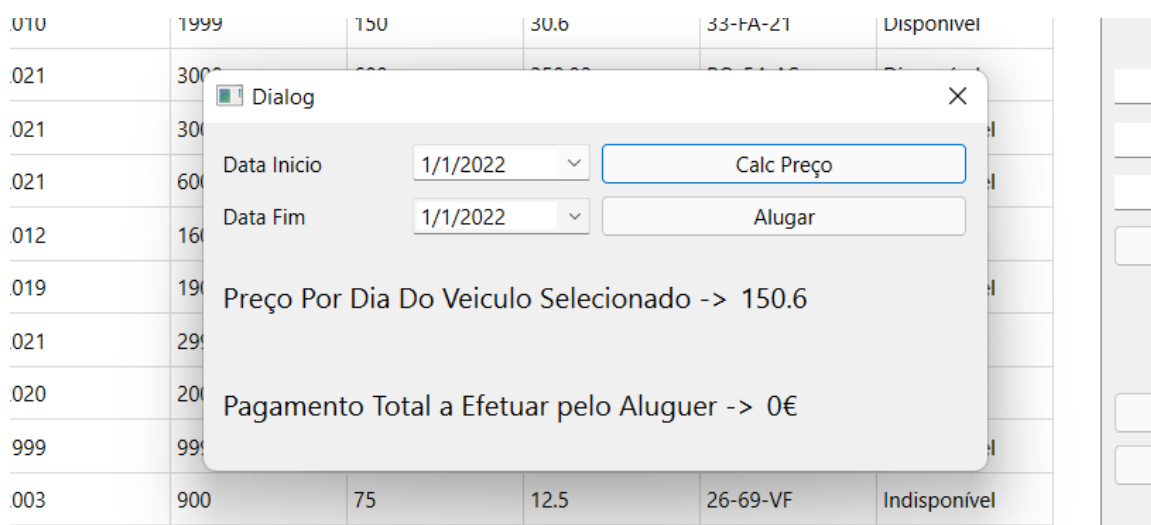
Nome : RUI

Cartão : 123456

Morada : RUADOTRIBUNAL

Figura 23 – Espaço cliente do cliente Rui

Em qualquer dos veículos, carregando duas vezes nele, a janela RentCarWindow vai aparecer no ecrã. Através dela o cliente vai seleccionar os dias que irá alugar o veículo, e poderá fazer uma simulação de preço através do botão calcular preço. Caso queira alugar, uma mensagem irá aparecer dando a opção de continuar ou cancelar a operação.



The screenshot shows a 'Dialog' window overlaid on a table of vehicles. The dialog contains fields for 'Data Inicio' (1/1/2022) and 'Data Fim' (1/1/2022), with buttons 'Calc Preço' and 'Alugar'. Below these, it displays: 'Preço Por Dia Do Veiculo Selecionado -> 150.6' and 'Pagamento Total a Efetuar pelo Aluguer -> 0€'. The background table shows vehicle details like ID, Year, Price, Power, License, and Availability.

ID	Ano	Preço	Potencia	Matricula	Disponivel
010	1999	150	30.6	33-FA-21	Disponivel
021	3000	200	35.0	33-FA-10	Disponivel
021	3000	200	35.0	33-FA-10	Disponivel
021	6000	200	35.0	33-FA-10	Disponivel
012	1600	150	30.6	33-FA-21	Disponivel
019	1900	150	30.6	33-FA-21	Disponivel
021	2900	150	30.6	33-FA-21	Disponivel
020	2000	150	30.6	33-FA-21	Disponivel
999	9900	150	30.6	33-FA-21	Disponivel
003	900	75	12.5	26-69-VF	Indisponível

Figura 24 – Sistema de aluguer

Em caso afirmativo, o carro será alugado pelo Rui, e ao aceder novamente ao espaço do cliente através do menu, constará o novo veículo alugado.

Dialog

	Cliente	Marca	Modelo	Ano	Cilindrada	Potencia	Matricula
1	RUI	FIAT	PANDA	2021	6000	700	77-TT-88
2	RUI	MERCEDES	GLA	2021	2999	500	PL-22-SA

Nome : RUI

Cartão : 123456

Morada : RUADOTRIBUNAL

Figura 25 – Espaço cliente do cliente Rui com o novo veículo

Conclusão

O trabalho prático foi enriquecedor, visto que nos possibilitou colocar em prática todos os conhecimentos obtidos ao longo da unidade curricular.

Possibilitou a evolução no processo de criação, mais propriamente os passos a seguir até ao programa final, demonstrando a importância de uma boa preparação e organização inicial, de modo a simplificar o processo de desenvolvimento, em prole da produção de um programa simples e altamente funcional.

O grupo foi capaz de cumprir todos os objetivos definidos para a criação da aplicação, adotando soluções praticas e inteligentes simplificando a implementação do programa.

Bibliografia

QT Creator:

<https://www.qt.io/product/development-tools> - consultado em janeiro de 2022.

<https://doc.qt.io/qt-5/qtablewidget.html> - consultado em janeiro de 2022.

<https://doc.qt.io/qt-5/qdateedit.html> - consultado em janeiro de 2022.

<https://doc.qt.io/qt-5/qlineedit.html> - consultado em janeiro de 2022.

Foi consultada também a documentação disponibilizada na unidade curricular.