

Índice

Índice	3
1. Introdução	4
2. Descrição do Sistema.....	5

1. Introdução

O objetivo do programa é simular um stand de automóveis com diversas opções, sendo possível utilizar uma oficina, uma loja de compra de automóveis e alugar um carro.

O programa consegue introduzir, eliminar, alterar e armazenar clientes, cada cliente pode alugar carros, arranjar o carro na oficina ou comprar um carro.

O programa contém 8 formulários e 8 partial classes.

2. Descrição do Sistema

Formulários

1. Form 1 (Homepage)

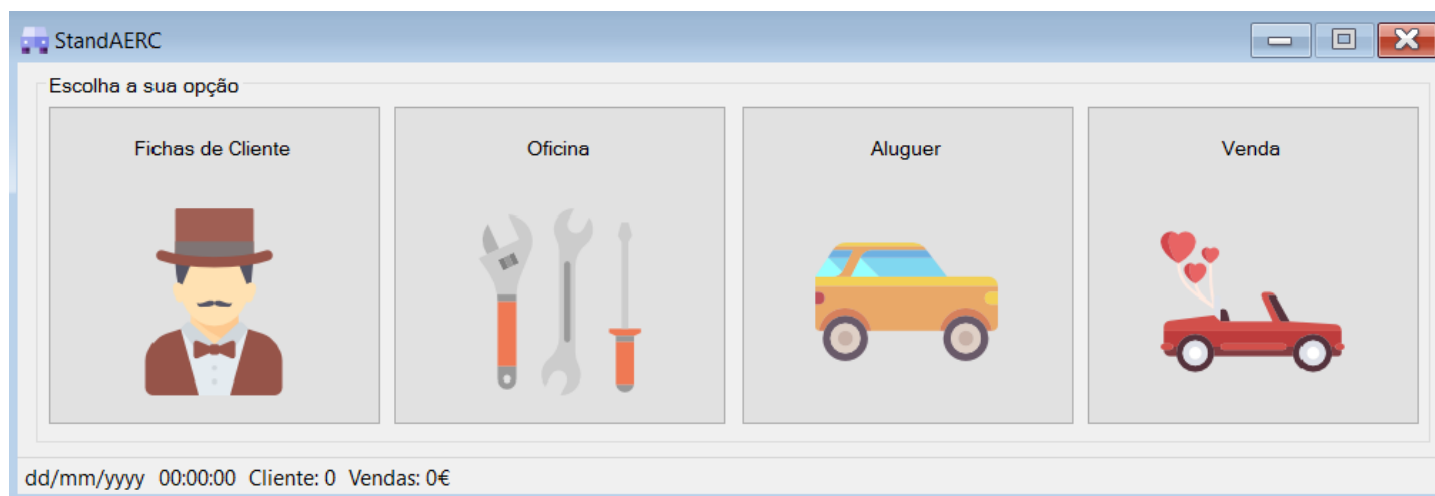


Figura 1 – Interface do Form1 (Homepage).

Na homepage existem 4 botões para abrir diferentes segmentos do programa, também mostra a data atual, o número de clientes e o total.

```
FormAluguer aluguer = new FormAluguer();  
  
if (aluguer.ShowDialog() != DialogResult.OK)  
    return;
```

Figura 2 – Código do botão para abrir o formulário Aluguer

Este exemplo de código mostra como a partir de um botão abre-se formulário Aluguer. A data, número de clientes e o total de vendas são atualizados em cada 1000ms.

1.2. FormFichasClientes

The screenshot shows a Windows application window titled "Fichas de Clientes". At the top, there's a toolbar with navigation icons (back, forward, etc.) and a status bar indicating "0 of {0}" items. The main content area is split into two parts. On the left, under the heading "Lista de Clientes", there is a search input field and a "Filtrar" button. Below this is a table with the following structure:

	Nome	NIF	Morada	Contacto
*				

The rest of the table area is greyed out. On the right side of the window, there are four text input fields labeled "Nome:", "NIF:", "Morada:", and "Contacto:" for data entry.

Figura 3 – Interface do formulário Fichas de Clientes.

O formulário Fichas de Clientes mostra a lista de Clientes armazenados na base de dados e permite introduzir, eliminar e filtrar clientes dentro da lista.

```
(from cliente in standAERC.Cliente orderby cliente.Nome select cliente).Load();
clienteBindingSource.DataSource = standAERC.Cliente.Local.ToBindingList();
```

Figura 4 – Código SQL permitindo ler os clientes dentro da base de dados

Este exemplo faz load dos clientes utilizando código SQL.

1.3. FormOficina

The screenshot shows a Windows application window titled "Oficina". The interface is organized into several functional areas. On the left, a vertical panel titled "Selecione um cliente" contains a "listBoxClientes". To the right of this panel, the "Cliente" section displays the selected client's details: "Nome: nenhum cliente selecionado" and "NIF: nenhum cliente selecionado". A "Total: 0.00€" is shown, along with an "Exportar Fatura" button. Below the client information, there are three main columns for adding items to the invoice. The "Carros" column features a "listBoxCarros" and buttons for "Adiciona Carro" and "Eliminar Carro". The "Serviços" column includes a "listBoxServicos", a dropdown menu, and buttons for "Criar Serviço" and "Eliminar Serviço". The "Parcelas" column has a "listBoxParcelas", input fields for "Descrição:" and "Valor:", and buttons for "Criar Parcela" and "Eliminar Parcela".

Figura 5 – Interface do formulário Oficina.

No formulário Oficina seleciona o Cliente mostrando o Nome, NIF e total da fatura e insere-se o veiculo a reparar (**FormAddCarro**), o serviço e a parcela.

Botão “Adicionar Carro” abre um form (**FormAddCarro**) que faz return dos valores que o utilizador introduziu. Esses valores são adicionados a um novo carro que por sua vez é adicionado a base de dados.

O botão “Criar Serviço” vai buscar o item selecionado da combobox.

“Criar Parcela” cria uma parcela com uma descrição e valor que é introduzido.

Os botões para eliminar, eliminam cada item selecionado respetivo ao seu botão.

1.4 FormAluguer

The screenshot shows a Windows application window titled "Aluguer". The interface is divided into several functional areas:

- Clientes:** A section on the top left containing a list box labeled "listBoxClientes" and a button labeled "Exportar Fatura".
- Carros alugados do cliente:** A section below "Clientes" containing a list box labeled "listBoxCarrosAlugados" and a button labeled "Eliminar Aluguer".
- Carros para Aluguer:** A large section on the right containing a list box labeled "listBoxCarrosAluguer" and two buttons at the bottom: "Adicionar Carro" and "Eliminar Carro".
- Alugar carro:** A bottom section containing input fields for "Data Inicial" (06/06/2016), "Data Final" (01/06/2017), "Preço", and "Kilometros", along with a button labeled "Alugar Carro".

Figura 6 – Interface do formulário Aluguer.

No formulário Venda seleciona o cliente, os carros para aluguer inseridos na base de dados e cria um registo de aluguer do cliente, cada cliente pode ter vários alugueres mas cada registo de aluguer só pode ter um carro.

Botão “Adicionar Carro” abre um form (**FormAddCarroAluguer**) que faz return dos valores que o utilizador introduziu. Esses valores são adicionados a um novo carro que por sua vez é adicionado a base de dados.

Quando clica no botão “Alugar Carro” o programa vai buscar o carro e o cliente selecionado e cria um aluguer relacionado com o cliente e guarda o data inicial, data final, o preço e os kilometros.

1.5. FormVenda

The image shows a Windows-style application window titled "Venda". Inside the window, there are several components: a "Clientes" section with a list box labeled "listBoxClientes" and an "Exportar Fatura" button below it; a "Carros para venda" section with a large list box labeled "listBoxCarroVendas" and two buttons, "Adicionar Carro" and "Eliminar Carro", at the bottom; a "Compras anteriores do cliente" section with a list box labeled "listBoxVendasCliente"; and a "Vender carro" section at the bottom with a "Data de Venda" date picker (showing 06/06/2016), a "Preço" text field, and a "Vender" button.

Figura 7 – Interface do formulário Vendas.

No formulário Venda seleciona o cliente pretendido, ao selecionar o cliente é mostrado as compras anteriores do mesmo, o carro para venda inserido, indica a data e o preço da venda.

Botão “Adicionar Carro” abre um form (**FormAddCarroVenda**) que faz return dos valores que o utilizador introduziu. Esses valores são adicionados a um novo carro que por sua vez é adicionado a base de dados.

O botão “Vender” faz o programa ir buscar o/s carro/s e o cliente selecionado e cria uma venda relacionado com o cliente e guarda a data de venda e o preço dessa venda.

Uma venda pode ter vários carros incluídos.

2. Entity Data Model

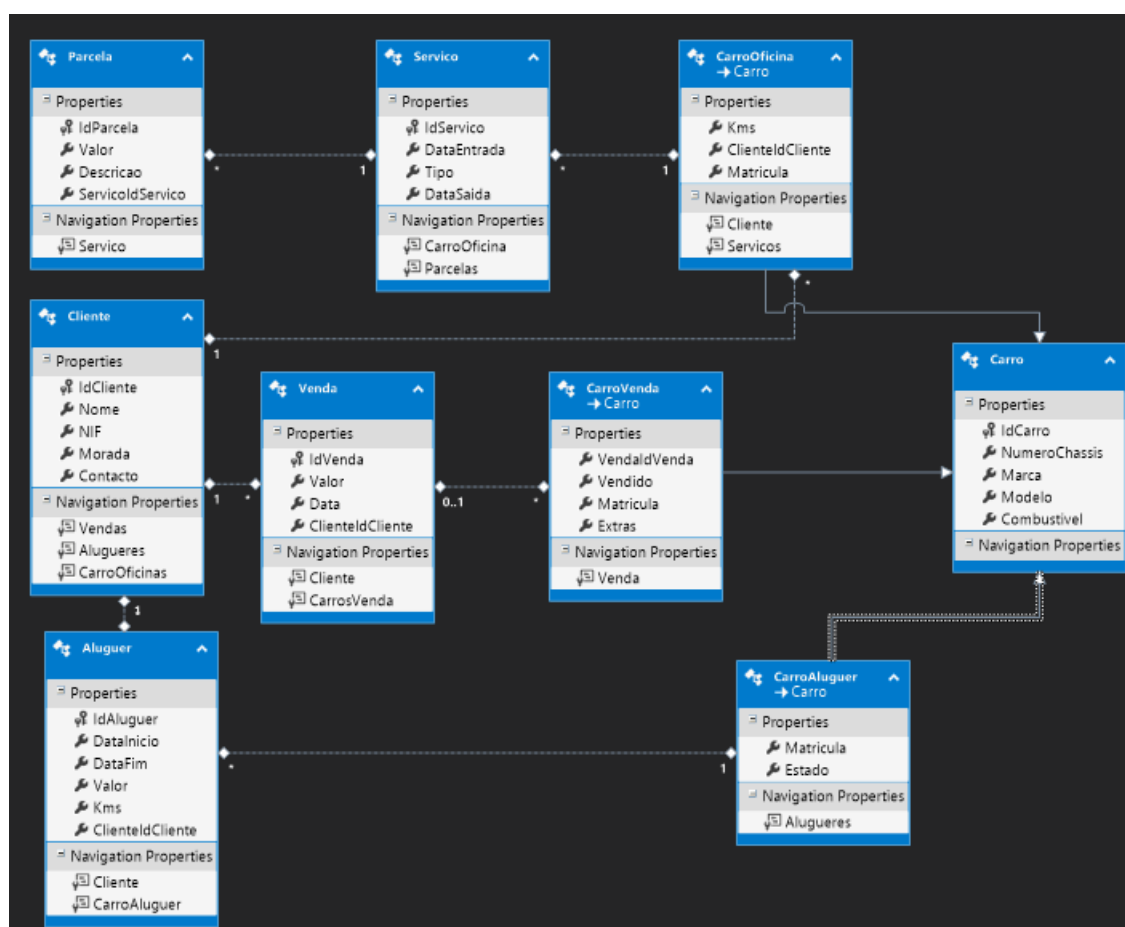


Figura 8 – Entity Data Model utilizado no projeto.