

Índice

Índice	3
1. Introdução	4
2. Descrição do Sistema	5
6	Erro! Marcador não definido.

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)

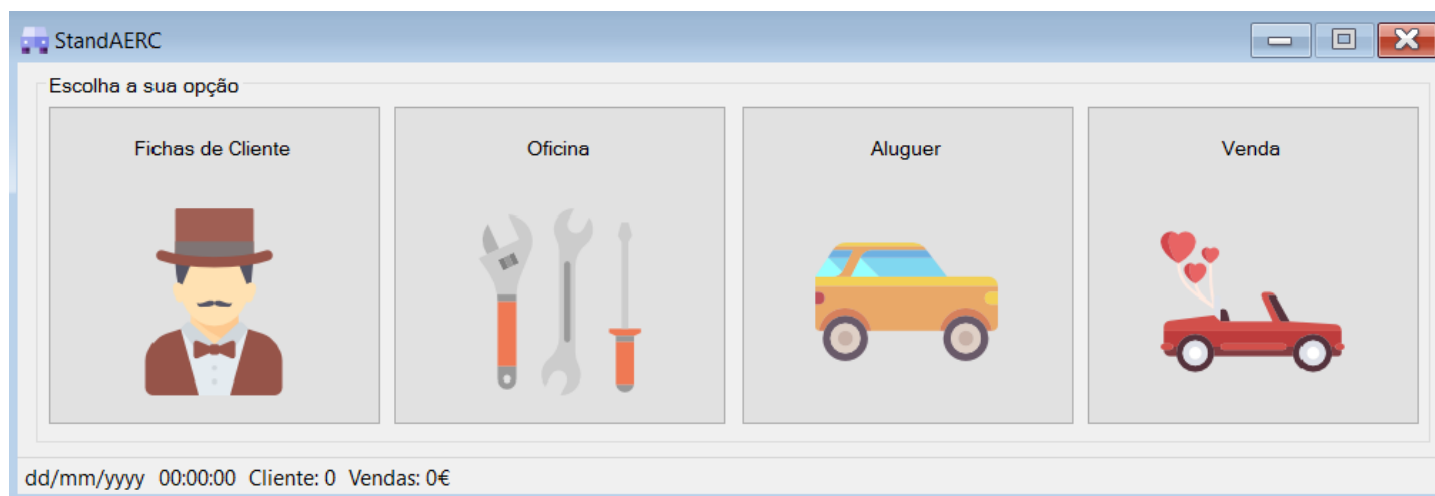


Figura 1 – Interface do Form1 (Homepage).

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, search, etc.) and a status bar indicating "0 of {0}". 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:".

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

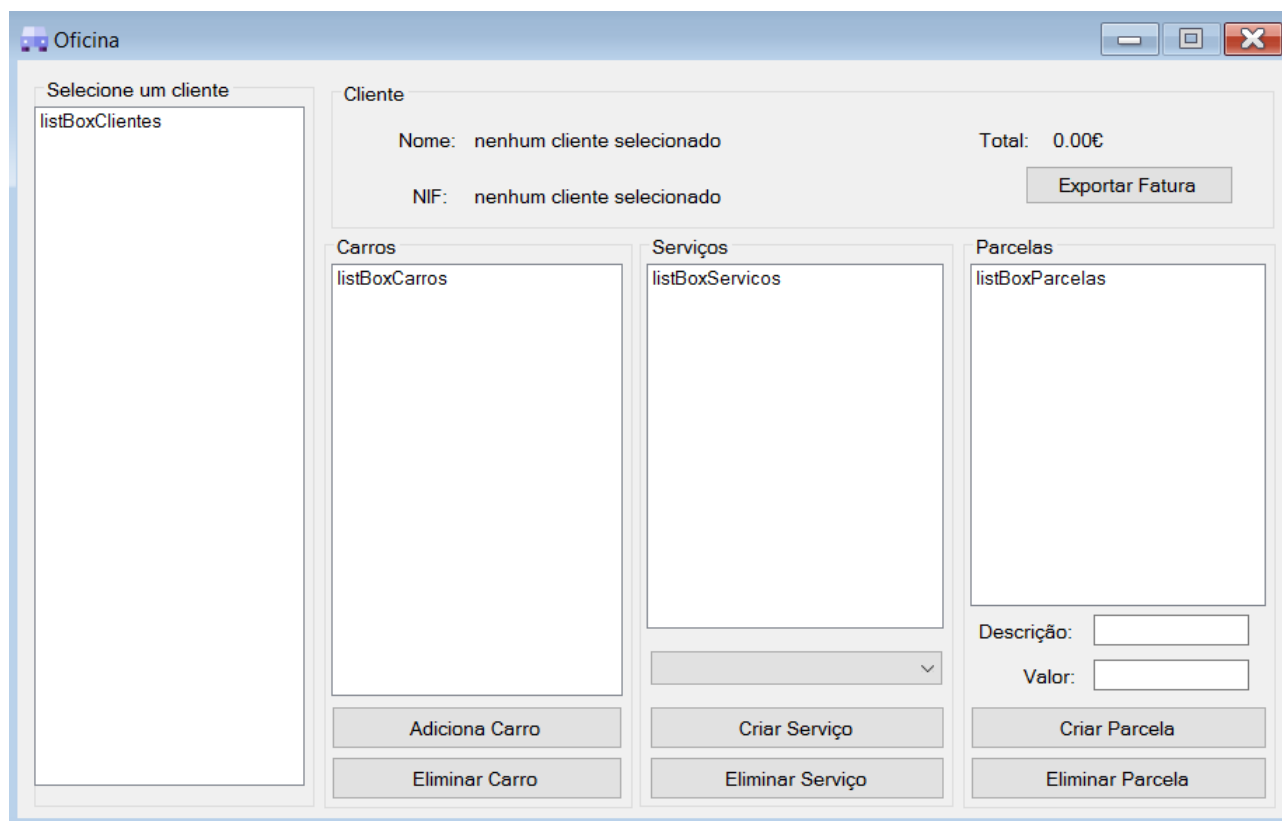


Figura 5 – Interface do formulário Oficina.

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 image shows a software window titled "Aluguer". It contains several functional areas: a "Clientes" area with a list box and an "Exportar Fatura" button; a "Carros alugados do cliente" area with a list box and an "Eliminar Aluguer" button; a "Carros para Aluguer" area with a large list box and "Adicionar Carro" and "Eliminar Carro" buttons; and a bottom section for entering rental details with fields for start/end dates, price, and kilometers, plus an "Alugar Carro" button.

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". It contains several UI elements: a "Clientes" section with a "listBoxClientes" and an "Exportar Fatura" button; a "Carros para venda" section with a "listBoxCarroVendas" and "Adicionar Carro" and "Eliminar Carro" buttons; a "Compras anteriores do cliente" section with a "listBoxVendasCliente"; and a "Vender carro" section at the bottom with a "Data de Venda" field (showing 06/06/2016), a "Preço" 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

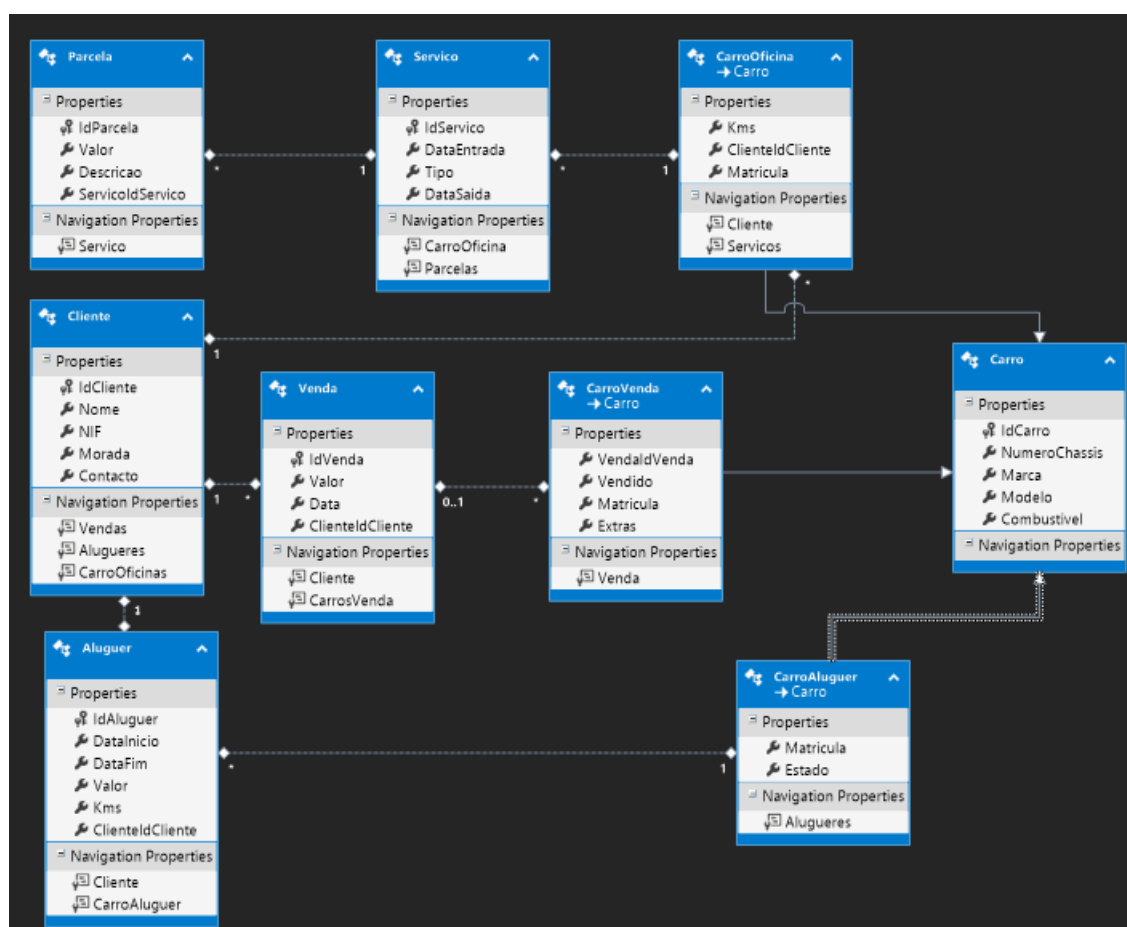


Figura 8 – Entity Data Model utilizado no projeto.