

Sistema de Gestão de uma Ótica

Projeto de Base de Dados

Universidade de Aveiro
LECI

Sara Gonçalves (98376), Pedro Santos (98158)



2022/2023

Conteúdo

1	Introdução	1
2	Análise de Requisitos	2
2.0.1	Entidades e Atributos	3
2.1	SQL Scripts	6
2.2	Interface	9
2.2.1	Linguagem de Desenvolvimento	9
2.2.2	Conexão com a Base de Dados	9
2.2.3	Formulários	10
2.3	Conclusão	14

Lista de Figuras

2.1	Diagrama Entidade-Relacionamento	4
2.2	Esquema Relacional	5
2.3	Remover Cliente	7
2.4	Transaction	8
2.5	Change Data to connect to other database	9
2.6	Login e Registo	10
2.7	Registo e Informação do Cliente	11
2.8	Receitas e ver consultas por doutor	12
2.9	Funcionalidades do Fornecedor e Exemplo	13

Capítulo 1

Introdução

No âmbito da cadeira de Base de Dados, o projeto que iremos desenvolver é um Sistema de Gestão de uma Ótica. A gestão de uma ótica é bastante importante para o seu sucesso, tanto de forma lucrativa como de organização, pois isso envolve, por exemplo, a seleção cuidadosa dos produtos necessários aos clientes, a manutenção de uma loja "agradável", com funcionários profissionais, entre outras coisas. A base de dados que irá ser desenvolvida permitirá guardar todos os dados necessários de um cliente, dos produtos disponíveis em loja, produtos que foram pedidos ao fabricante, receitas, pagamentos, consultas, etc.

Capítulo 2

Análise de Requisitos

Os utilizadores deste sistema (funcionários da ótica que tiverem autorização) deverão conseguir:

- Inserir/Remover clientes no registo de clientes da ótica;
- Inserir/Remover informação de determinado cliente;
- Agendar consultas com o data e hora;
- Atribuir a consulta a um Doutor;
- Verificar se determinado produto encontra-se em loja;
- Verificar pedidos ao fabricante;
- Registar produtos novos no sistema e adicionar informações como tipo, marca, preço e quantidade em stock;
- Atualizar informações de produtos existentes, como preço, marca ou quantidade em stock;
- Verificar o stock de um produto específico;
- Gerir as encomendas a fornecedores, incluindo fazer pedidos de novos produtos e registar entregas de produtos;
- Registar informações de pagamento de clientes, incluindo consultas e produtos comprados;
- Gerir o calendário dos Doutores, permitindo que os recepcionistas agendem consultas em horários disponíveis;

2.0.1 Entidades e Atributos

- **Pessoa:** caracterizada por número CC, nome, telemóvel, morada e email. Pode ser um Funcionário ou um Cliente.
- **Funcionário:** é uma pessoa que trabalha na ótica e é caracterizado por um número de CC (FK), nome, telemóvel, morada e email e NIF. Podem ser Recepcionista ou Doutor.
- **Recepcionistas:** caracterizados pelo número de funcionário e são responsáveis por marcar consultas e por vender os produtos aos clientes.
- **Doutores:** são caracterizados pelo número de SNS e são responsáveis por realizar as consultas.
- **Cliente:** é uma pessoa que compra produtos ou "vai" às consultas e é caracterizado por um número de CC(FK), nome, morada, telemóvel, email e NIF.
- **Fornecedor:** é uma pessoa que fornece os produtos pedidos e é caracterizado por NIF (PK), um nome, morada, telemóvel, email.
- **Produto:** é o produto que está em loja ou que poderá ser pedido ao fornecedor e é caracterizado pelo tipo de produto, marca, preço, quantidade, nome e ID do produto (PK).
- **Consulta:** é uma consulta marcada pelo Recepcionista, realizada pelo Doutor para o Cliente. É caracterizada pelo número da consulta (PK), data, hora, preço, CC do cliente (FK) e CC do doutor (FK).
- **Receita:** é uma receita passada pelo Doutor depois da consulta. É caracterizada pelo ID da receita(PK), descrição, CC do cliente (FK), CC do doutor (FK) e NIF do fornecedor (FK).
- **Pagamento:** é o pagamento do produto ou da consulta. É caracterizado pelo Valor, método de pagamento, data de pagamento, comprovativo (PK), CC do cliente (FK) e ID do produto (FK).
- **Fatura:** é o documento fiscal que é emitido para o cliente após uma compra. É caracterizado pelo ID da fatura(PK), pela descrição do produto, impostos, preço total e comprovativo (FK).
- **Stock:** representa a quantidade de cada produto disponível em loja num determinado momento. é caracterizado pelo ID do stock (PK), pelo ID do produto(FK), quantidade, data de entrada, data de saída, e o estado (se está em stock ou não).

Diagrama Entidade-Relacionamento

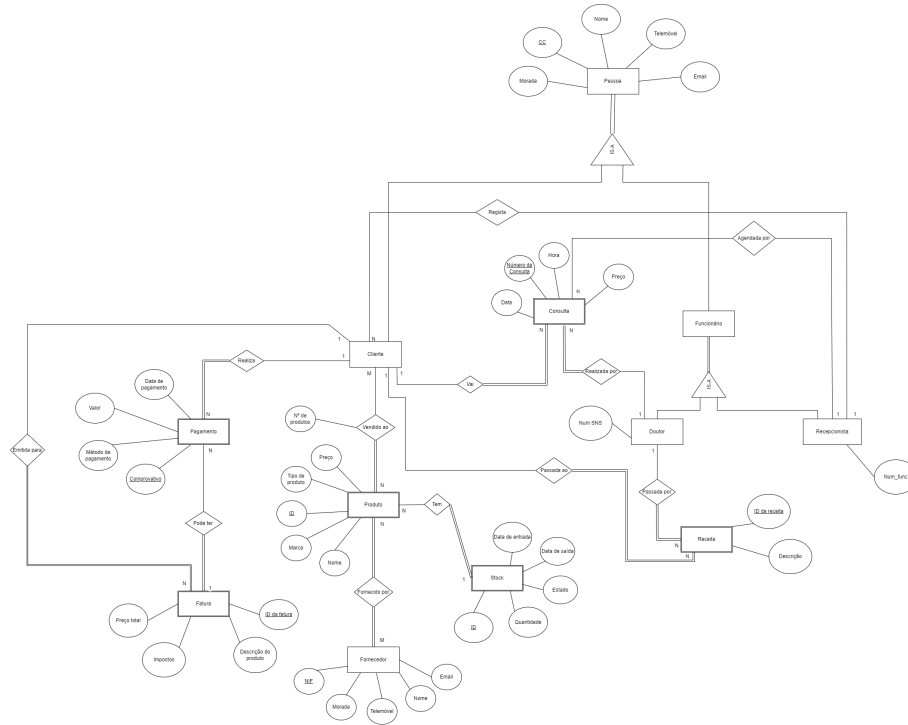


Figura 2.1: Diagrama Entidade-Relacionamento

Esquema Relacional

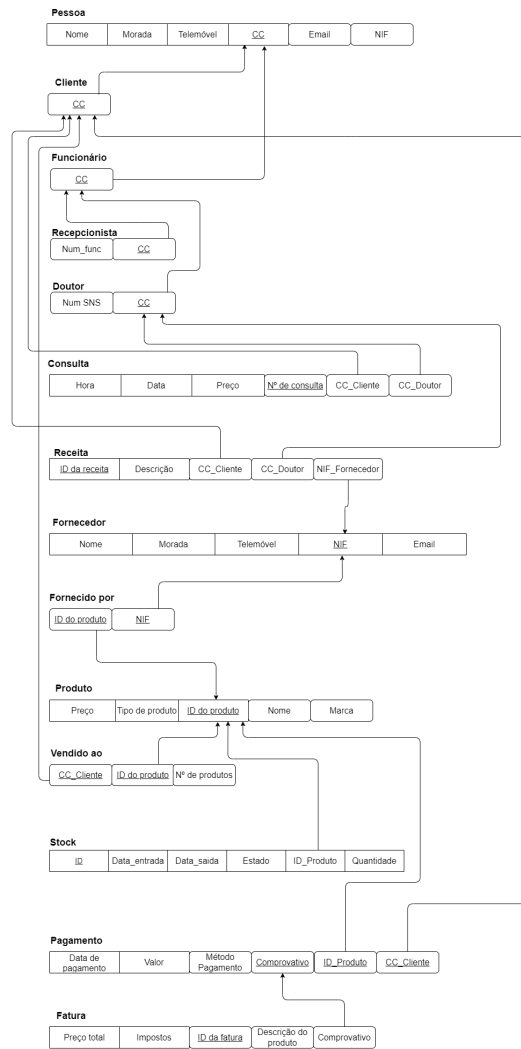


Figura 2.2: Esquema Relacional

2.1 SQL Scripts

Todo o código desenvolvido está inserido na pasta `SQL_scripts`. Esta pasta contém os seguintes ficheiros: `dropTables.sql`, `queries.sql`, `indexes.sql`, `Tables.sql`, `insertValues.sql`, `sp.sql`, `udfs.sql`, `triggers.sql`, `views.sql` e `transactions.sql`.

Tables: Criação das tabelas da base de dados.

Drop: Eliminação das tabelas da base de dados.

InsertValues: Inserir dados nas tabelas.

Queries: Queries de teste, para visualizar os dados das tabelas.

Views: Retornam o número total de consultas (por doutor e por cliente) e a informação sobre o stock.

Triggers: O trigger `CheckStockQuantity` foi criado para garantir que a quantidade de produtos em stock na tabela `Stock` nunca seja negativa.

Indexes

- `Idx_cliente_cc` sobre a tabela `Cliente` e atributo `cc`;
- `Idx_produto_id` sobre a tabela `produto` e atributo `id`;
- entre outros;

UDFs

- `IsProductInStock (@produto_id VARCHAR(20))`: Verifica se um produto está em stock com base no seu ID, retornando um valor booleano indicando a disponibilidade.
- `CalculateTotalSales (@cliente_cc INT)`: Calcula o total de vendas de um cliente com base nos registos da tabela `Otica_Vendido_Ao`, retornando o valor total das vendas.
- `GetTotalProductsSuppliedBySupplier (@fornecedor_nif VARCHAR(20))`: Retorna o total de produtos fornecidos por um fornecedor com base no seu NIF, utilizando informações das tabelas `Otica_Stock` e `Otica_Fornecido_Por`.

Stored Procedures

- InserirCliente: Inserir um novo cliente na base de dados, simplificando a inserção de informações pessoais na tabela Otica_Pessoa e informações específicas do cliente na tabela Otica_Cliente.
- RemoverCliente: Realiza a remoção completa de um cliente e os dados associados, garantindo a consistência das operações em caso de erro.

```
1 ALTER PROCEDURE RemoverCliente
2     @CC INT,
3     @Resultado VARCHAR(100) OUTPUT
4 AS
5 BEGIN
6     BEGIN TRY
7
8         -- Verificar se há uma transação em andamento e confirmá-la ou desfazê-la
9         IF @@TRANCOUNT > 0
10         BEGIN
11             COMMIT;
12         END
13
14         BEGIN TRANSACTION;
15
16         DELETE FROM Otica_Pagamento WHERE cliente_cc = @CC;
17
18         DELETE FROM Otica_Fatura WHERE cliente_cc = @CC;
19
20         DELETE FROM Otica_Receita WHERE cliente_cc = @CC;
21
22         DELETE FROM Otica_Consulta WHERE cliente_cc = @CC;
23
24         DELETE FROM Otica_Vendido_Ao WHERE cliente_cc = @CC;
25
26         DELETE FROM Otica_Cliente WHERE CC = @CC;
27
28         DELETE FROM Otica_Pessoa WHERE CC = @CC;
29
30         COMMIT;
31         -- Confirmação da transação
32         SET @Resultado = 'Cliente removido com sucesso.';
33     END TRY
34     BEGIN CATCH
35         IF @@TRANCOUNT > 0
36         BEGIN
37             ROLLBACK;
38         END
39         -- Rollback em caso de erro
40         SET @Resultado = 'Não foi possível remover o cliente.';
41     END CATCH;
42 END
```

Figura 2.3: Remover Cliente

- AgendarConsulta: Agendar uma consulta na ótica, facilitando a inserção de informações na tabela Otica_Consulta.
- RegistrarVendaProduto: Registrar a venda de um produto a um cliente, inserindo um registro na tabela Otica_Vendido_Ao com informações como

cliente, produto e quantidade vendida.

- **RegistrarEntregaProduto:** Registrar a entrega de um produto à ótica, inserindo um registo na tabela `Otica_Stoc`. Usado para atualizar o stock após uma entrega.
- **RegistrarPagamentoFatura:** Registrar o pagamento de uma fatura feita pelo cliente, inserindo um registo na tabela `Otica_Pagamento`. Ajuda a manter um registo organizado dos pagamentos dos clientes.

Transactions: Transação usada para inserir os dados do funcionário aquando o seu registo, ou seja garante que todas as operações de inserção sejam tratadas como uma unidade, para que todas as inserções sejam concluídas com êxito ou nenhuma alteração seja feita se ocorrer algum erro.

```
BEGIN TRANSACTION;

BEGIN TRY

    DECLARE @cc INT;
    DECLARE @nome VARCHAR(60);
    DECLARE @tele INT;
    DECLARE @email VARCHAR(60);
    DECLARE @rua VARCHAR(200);
    DECLARE @num_funcionario INT;
    DECLARE @username VARCHAR(50);
    DECLARE @password VARCHAR(50);

    -- Atribuir os valores dos parâmetros às variáveis
    SET @cc = @cartao;
    SET @nome = @nome_param;
    SET @tele = @tele_param;
    SET @email = @email_param;
    SET @rua = @rua_param;
    SET @num_funcionario = @nume_param;
    SET @username = @user_param;
    SET @password = @password_param;

    -- Inserir dados na tabela Pessoa
    INSERT INTO Otica_Pessoa (CC, nome, telemovel, email, morada)
    VALUES (@cc, @nome, @tele, @email, @rua);

    -- Inserir dados na tabela Funcionario
    INSERT INTO Otica_Funcionario (CC)
    VALUES (@cc);

    -- Inserir dados na tabela Rececionista
    INSERT INTO Otica_Rececionista (CC, num_funcionario)
    VALUES (@cc, @num_funcionario);

    -- Inserir dados na tabela Login
    INSERT INTO Otica_Login (CC, username, password)
    VALUES (@cc, @username, @password);

    COMMIT;
    -- Todos os inserts foram bem-sucedidos, confirmar a transação

END TRY
BEGIN CATCH
    ROLLBACK;
    -- Ocorreu um erro em algum dos inserts, desfazer todas as alterações
END CATCH;
```

Figura 2.4: Transaction

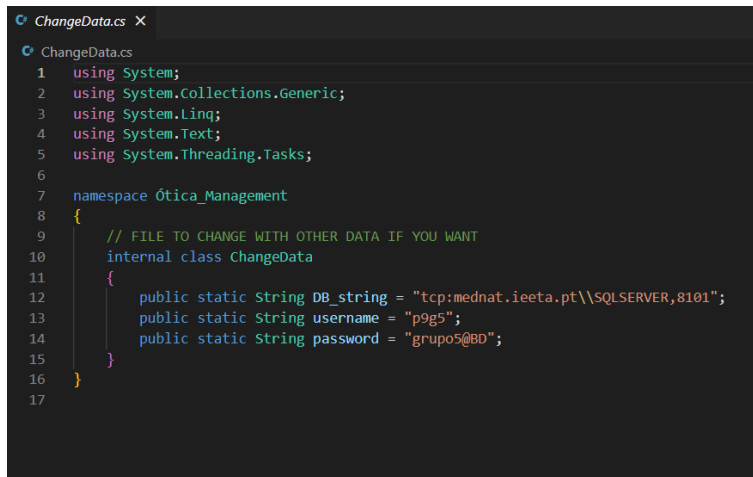
2.2 Interface

2.2.1 Linguagem de Desenvolvimento

Para o desenvolvimento da interface foi criado vários formulários gráficos utilizando Windows Form em Visual Studio (C#).

2.2.2 Conexão com a Base de Dados

Para que seja possível a testagem da aplicação noutra base de dados para além da base de dados das aulas práticas, é necessário aceder ao ficheiro **ChangeData.cs** e mudar os atributos correspondentes.



```
ChangeData.cs X
ChangeData.cs
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace Ótica_Management
8  {
9      // FILE TO CHANGE WITH OTHER DATA IF YOU WANT
10     internal class ChangeData
11     {
12         public static String DB_string = "tcp:mednat.ieeta.pt\\SQLSERVER,8101";
13         public static String username = "p9g5";
14         public static String password = "grupo5@BD";
15     }
16 }
17
```

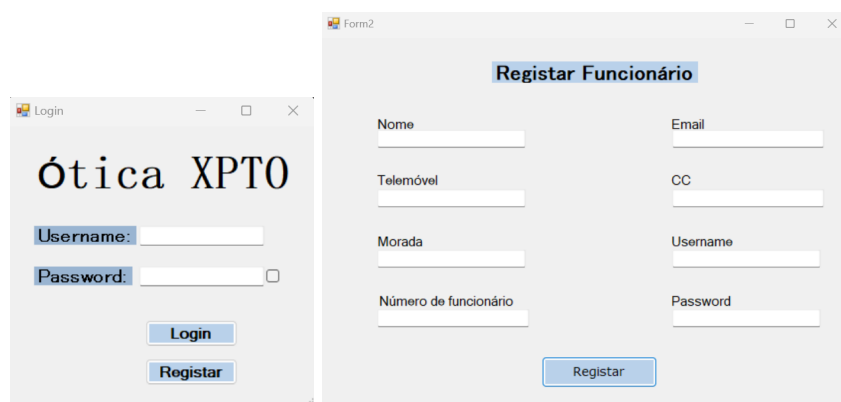
Figura 2.5: Change Data to connect to other database

2.2.3 Formulários

De forma a entender melhor o sistema, irá ser apresentado alguns dos formulários principais desenvolvidos e respetivas funcionalidades.

Login

Quando a aplicação é iniciada, aparece um sistema de Login para o recepcionista que tiver permissão para entrar. É possível por optar também por registar um novo recepcionista, completando os vários campos necessários.



The image displays two overlapping windows from a software application. The window on the left, titled 'Login', features the text 'ótica XPTO' at the top. Below it are input fields for 'Username:' and 'Password:', followed by 'Login' and 'Registar' buttons. The window on the right, titled 'Form2' and 'Registar Funcionário', contains a registration form with fields for 'Nome', 'Email', 'Telemóvel', 'CC', 'Morada', 'Número de funcionário', 'Username', and 'Password', along with a 'Registar' button at the bottom.

Figura 2.6: Login e Registo

Menu

Após efetuado o Login, aparece um menu com várias funcionalidades, entre estas:

- Registar Cliente;
- Ver informação dos clientes;
- Registar Doutor;
- Secção do Doutor;
- Secção do Cliente;
- Secção do Fornecedor.

Secção do Cliente

Nos formulários respetivos ao Registar Cliente e Informação Cliente, existem as funcionalidades:

- Registar um cliente, completando todos os campos referentes aos atributos;
- Ver a lista dos clientes presentes na base de dados;
- Clicando duas vezes num dos clientes na lista, ou procurando o cliente pelo seu CC, é aberto um formulário correspondente ao perfil do cliente. Neste formulário é possível:
 - Editar os dados do cliente;
 - Eliminar os dados do cliente;

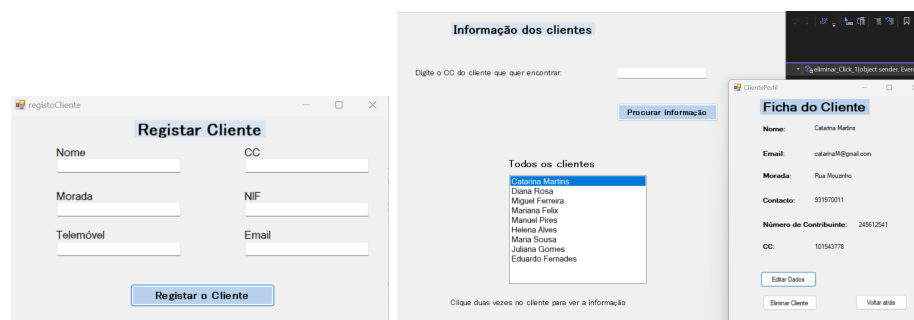


Figura 2.7: Registo e Informação do Cliente

No formulário respetivo ao Cliente, encontram-se as seguintes opções:

- Fazer compra, onde o funcionário escolhe um cliente, um produto e a quantidade que ele deseja comprar;
- Marcar consulta, onde o funcionário tem de inserir o CC do cliente, a data e hora, o preço e o nome do doutor que vai fazer a consulta. É importante mencionar que o funcionário consegue ver a disponibilidade do doutor para a data e hora marcadas;
- Ver recibos de um cliente;
- Gastos em produtos, onde o funcionário escolhe um cliente e vê quanto é que ele gastou em produtos na loja.

Secção do Doutor

Nos formulários respetivos ao Doutor, existem as funcionalidades:

- Registar um Doutor, completando todos os campos referentes aos atributos;
- Ver a lista dos clientes presentes na base de dados e a informação de cada;
- Ver consultas marcadas para cada doutor, apresentando:
 - o nome do cliente
 - data e hora;
- Passar receitas, onde indica o doutor, o produto, o CC do cliente e uma descrição.

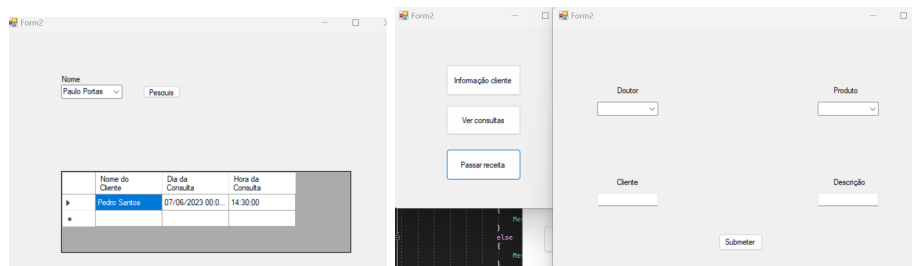


Figura 2.8: Receitas e ver consultas por doutor

Secção dos Fornecedores

Nos formulários respetivos ao Fornecedores, existem as funcionalidades:

- Adicionar um fornecedor, onde indica os campos respetivos;
- Adicionar produto a fornecedor, onde o funcionário associa um produto a um fornecedor;
- Fazer pedido, onde se escolhe um fornecedor, um produto que está associado a ele e uma quantidade. Neste momento, o pedido é inserido na tabela relativa ao stock, com o estado "pendente";
- Ver produtos, onde aparece todos os produtos e os seus dados respetivos na base de dados. Aqui também é possível verificar se ele está em stock, inserir um novo produto, ver o stock de todos os produtos e eliminar o produto;
- Ver o número de produtos fornecidos, onde pode-se escolher um fornecedor e é dado o número de produtos que ele forneceu à ótica;

- Ver pedidos, onde aparecem os pedidos com o estado "pendente" e os respectivos dados, e onde o funcionário pode aceitá-los, colocando o estado a "em stock".

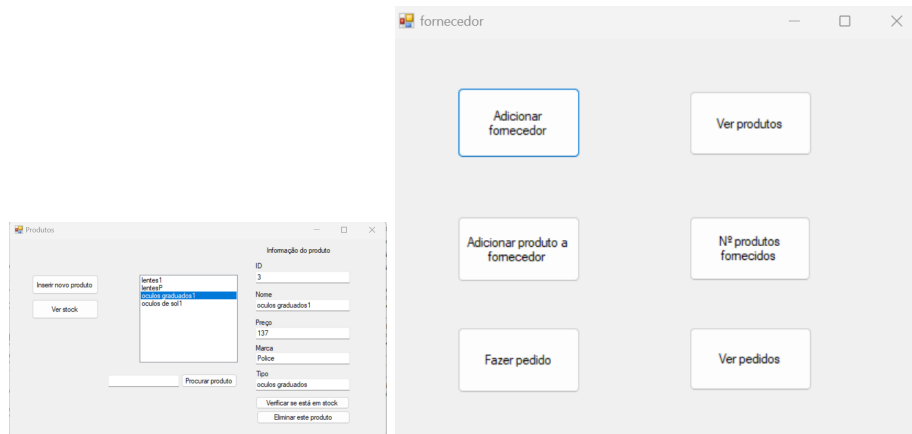


Figura 2.9: Funcionalidades do Fornecedor e Exemplo

2.3 Conclusão

Concluindo, todos os objetivos e funcionalidades principais da aplicação para a gestão de uma ótica foram cumpridas. O funcionário pode registrar, atualizar e eliminar um cliente, agendar consultas, ver pagamentos e reservas, adicionar fornecedores e fazer pedidos, entre outras funcionalidades.