

Relatório do trabalho da disciplina de Armazenamento e Acesso a Dados

O Negócio de Aluguer de Veículos

Fábio Rafael Gomes Costa - 22997

Lino Emanuel Oliveira Azevedo - 23015

Pedro Lourenço Moraes Rocha – 23009

Licenciatura em Engenharia de Sistemas Informáticos

dezembro de 2024

Afirmo por minha honra que não recebi qualquer apoio não autorizado na realização deste trabalho prático. Afirmo igualmente que não copieei qualquer material de livro, artigo, documento web ou de qualquer outra fonte exceto onde a origem estiver expressamente citada.

Fábio Rafael Gomes Costa - 22997

Lino Emanuel Oliveira Azevedo - 23015

Pedro Lourenço Morais Rocha - 23009

Índice

Índice	II
Índice de Figuras	III
1. Descrição do modelo de negócio	1
2. Diagrama ER	3
3. Diagrama ER Corrigido	4
4. Query	5
5. Aplicação com acesso e manipulação a Base de Dados.....	9
5.1 Descrição da aplicação desenvolvida	9
5.1.1 Funcionalidades Principais	9
5.1.2 Funcionalidades Adicionais.....	10
5.2 Interface(s) da aplicação	11
5.3 Descrição da forma de interação com a Base de dados	14
5.3.1 Conexão com a Base de Dados	14
5.3.2 Consultas SQL e Preenchimento de Dados	14
5.3.3 Comandos SQL Parametrizados	15
5.3.4 <i>Stored Procedure</i>	15
Anexos	17
Anexo I – Diagrama ER (Tamanho A3).....	18
Anexo II– Diagrama ER Corrigido (Tamanho A3).....	19

Índice de Figuras

Figura 1 - Diagrama ER.....	3
Figura 2 - Diagrama ER Corrigido	4
Figura 3 - Interface Inicial	11
Figura 4 - Interface 'Inserir Cliente'	11
Figura 5 - Interface 'Atualizar Dados Cliente'	12
Figura 6 - Interface 'Apagar Veículo'	12
Figura 7 - Interface 'Gerir Seguro'	13
Figura 8 - Diagrama ER (TamanhoA3).....	18
Figura 9 - Diagrama ER Corrigido (Tamanho A3)	19

1. Descrição do modelo de negócio

O início de todo o processo está no contacto que um cliente (particular ou organização) estabelece com a empresa, através de uma loja física, ou através de contacto telefónico.

O interesse do potencial cliente prende-se com o facto de pretender o aluguer de um veículo ou a sua reserva. Este veículo podendo ser um carro, uma mota, uma carrinha ou um camião.

Esse interesse firma-se no preenchimento de um formulário, onde se solicitam dados básicos tais como, data de levantamento e de devolução do veículo, informações pessoais do cliente (nome, endereço, data de nascimento, contactos, NIF (Número de Identificação Fiscal), método de pagamento, histórico do cliente, entre outros), o veículo escolhido pelo cliente (marca, modelo, matrícula, número de identificação do veículo, estado do veículo, preço por dia do veículo, entre outros).

O formulário preenchido é entregue no estabelecimento da empresa, onde a partir dos dados é realizado um orçamento, tendo em conta o histórico do cliente (infrações cometidas anteriormente) e o valor diário do veículo. O orçamento consiste em quatro valores sendo estes o valor de entrada, o valor da caução, o valor da quitação e desconto.

Com o orçamento pronto o mesmo é enviado ao cliente que posteriormente informa se pretende dar seguimento ao aluguer.

Com a aprovação do cliente, é celebrado um contrato, contendo os dados anteriormente obtidos e os termos de utilização dos veículos vigentes na empresa (pagamento de taxa por atraso na entrega, taxas de manutenção e limpeza, entre outros). O contrato é entregue ao cliente para este ser assinado, aquando da assinatura do contrato é exigido ao cliente o pagamento do valor de entrada e o valor da caução, assim alterando o estado de disponibilidade do veículo de disponível para alugado.

No dia de levantamento do veículo é pago o restante valor estipulado no contrato, e é dado ao cliente uma formação básica sobre as utilidades do veículo.

Enquanto o veículo se encontra alugado, se achar necessário, a empresa consegue aceder a localização do mesmo, em tempo real, a partir do seu número de identificação e do GPS, sendo uma funcionalidade fundamental para caso o cliente solicite o serviço de estrada em caso de avaria, assim a empresa pode enviar uma das empresas de manutenção automóvel com a qual tenha parceria sem a necessidade de solicitar a localização ao cliente.

No dia da entrega do veículo, é realizada uma perícia ao mesmo, e caso o veículo não se encontre dentro dos parâmetros aceitáveis (sem danos a viatura e limpo) é aplicada uma taxa para cada situação identificada, essas taxas por danos e limpeza ou até de atraso são debitadas ao valor da caução, caso o valor da caução seja inferior ao valor a debitar, este deve ser pago pelo cliente posteriormente, estas infrações ficam registadas no histórico do cliente. Após a verificação do veículo é assinado o comprovativo de entrega, dando fim ao contrato, e alterando o estado de disponibilidade do veículo para disponível caso este não vá para a manutenção. Além deste é solicitado ao cliente o feedback sobre o serviço prestado.

Os veículos devolvidos que não se encontram em condições de voltarem a ser alugados, são encaminhadas as empresas com a qual já tenhamos um contrato firmado para a manutenção dos mesmos, ou em casos excecionais são enviados para as fornecedoras (se e apenas se este ainda se encontrar dentro do período de garantia).

Quando estes retornam o seu estado de disponibilidade retorna a ser disponível. No início de cada ano é renovado o seguro de todas as viaturas e é pago o IUC (Imposto Sobre Circulação) das mesmas, além de serem agendadas as inspeções e as revisões (sendo estas semestrais). Caso estes não estejam em dia, a disponibilidade do veículo é alterada para indisponível até a sua situação ser regularizada.

4. Query

1. Quantos veículos foram alugados no dia 12 de janeiro de 2019, mas que nunca precisaram de uma manutenção?
2. Qual o Cliente que realizou mais alugueres?
3. Qual a marca cujo veículos são mais alugados?
4. Quais os carros com mais de 2 anos foram alugados 3 ou mais vezes, deve mostra quantas vezes foi alugado?
5. Quais os 5 clientes com menos infrações?
6. Quais os carros que realizaram manutenção no segundo trimestre de 2023?
7. Quais os veículos, e as datas de aluguer se existirem, da seguradora “Fidelidade”?
8. Qual a despesa anual dos 10 veículos com a despesa mais elevada no ano de 2022?
9. Quais os modelos que foram fornecidos pela 1 e 2?
10. Qual a média de classificação de cada marca (deve calcular primeiro a média por veículo, e só depois por marca)?

--1. Quantos veículos foram alugados no dia 12 de janeiro de 2019, mas que nunca precisaram de uma manutenção

```
Select Count( Distinct VeiculoIDVeiculo)
From Aluguer
Where DataLevantamento = '2019-01-12'
AND VeiculoIDVeiculo not in (Select VeiculoIDVeiculo
                             From Manutencao)
```

--2. Qual o Cliente que realizou mais alugueres

```
Select IdCliente, NomeCliente, NAlugueres=count(IDAluguer)
From Cliente Join Aluguer on
Cliente.IDCliente=Aluguer.ClienteIDCliente
Group by IdCliente, NomeCliente
Having Count(IDAluguer) >= All (Select count(IDAluguer)
                               From Cliente Join Aluguer on
                               Cliente.IDCliente=Aluguer.ClienteIDCliente
                               group by IDCliente)
```

--3. Qual a marca cujo veículos são mais alugados

```

Select top 1 IDMarca, MarcaVeiculo.DescMarca,
NALugueres=Count(IDAluguer)
From Aluguer Join Veiculo on
Aluguer.VeiculoIDVeiculo=Veiculo.IDVeiculo
Join ModeloVeiculo on
Veiculo.ModeloVeiculoIDModelo=ModeloVeiculo.IDModelo
Join MarcaVeiculo on
ModeloVeiculo.MarcaVeiculoIDMarca=MarcaVeiculo.IDMarca
Group by IDMarca, MarcaVeiculo.DescMarca
Having Count(IDAluguer) >= All (Select Count(IDAluguer)
From Aluguer
Group by IDAluguer)
Order by NAlugueres Desc

```

--OU--

```

Select IDMarca, DescMarca, Count(IDAluguer) as NAlugueres
From Aluguer Join Veiculo on
Aluguer.VeiculoIDVeiculo=Veiculo.IDVeiculo
Join ModeloVeiculo on
Veiculo.ModeloVeiculoIDModelo=ModeloVeiculo.IDModelo
Join MarcaVeiculo on
ModeloVeiculo.MarcaVeiculoIDMarca=MarcaVeiculo.IDMarca
Group by IDMarca, DescMarca
Having Count(IDAluguer) >= All (Select Count(IDAluguer)
From Aluguer Join Veiculo on
Aluguer.VeiculoIDVeiculo=Veiculo.IDVeiculo
Join ModeloVeiculo on
Veiculo.ModeloVeiculoIDModelo=ModeloVeiculo.IDModelo
Join MarcaVeiculo on
ModeloVeiculo.MarcaVeiculoIDMarca=MarcaVeiculo.IDMarca
Group by IDMarca, DescMarca)

```

--4. Quais os carros com mais de 2 anos foram alugados 3 ou mais vezes, deve mostra quantas vezes foi alugado

```

Select Veiculo.IDVeiculo, Veiculo.MatriculaVeiculo, count
(Aluguer.VeiculoIDVeiculo) as TotAlugueres
From Veiculo Join Aluguer on
Veiculo.IDVeiculo=Aluguer.VeiculoIDVeiculo
Where DATEDIFF(YEAR, Veiculo.DataFabricacao, GETDATE()) >= 2
Group by Veiculo.IDVeiculo, Veiculo.MatriculaVeiculo
Having count (Aluguer.VeiculoIDVeiculo) >= 3

```

--5. Quais os 5 clientes com menos infrações

```
Select TOP(5) Cliente.IDCliente, NomeCliente,  
COUNT(IDInfracao) AS TotalInfracoes  
From Cliente  
Left JOIN Infracoes ON Infracoes.ClienteIDCliente =  
Cliente.IDCliente  
GROUP BY Cliente.IDCliente, NomeCliente  
ORDER BY COUNT(IDInfracao) ASC;
```

--6. Quais os carros que realizaram manutenção no segundo trimestre de 2023

```
Select Veiculo.IDVeiculo, Veiculo.MatriculaVeiculo  
From Veiculo Join Manutencao on  
Veiculo.IDVeiculo=Manutencao.VeiculoIDVeiculo  
Where DataManutencao > '2023-04-01' AND DataManutencao <  
'2023-06-30'
```

--7. Quais os veículos, e as datas de aluguer se existirem, da seguradora “Fidelidade”

```
Select Distinct IDVeiculo, Aluguer.DataLevantamento as  
Levantamento, Aluguer.DataDevolucao as Devolucao  
From Veiculo Left Join Aluguer on  
Aluguer.VeiculoIDVeiculo=Veiculo.IDVeiculo  
Join Seguro on  
Veiculo.IDVeiculo=Seguro.VeiculoIDVeiculo  
Join Seguradora on  
Seguro.SeguradoraIDSeguradora=Seguradora.IDSeguradora  
Where Seguradora.DescSeguradora = 'Fidelidade'
```

--8. Qual a despesa anual dos 10 veículos com a despesa mais elevada no ano de 2022

```
Select Top(10) Veiculo.IDVeiculo, Veiculo.MatriculaVeiculo,  
SUM(ValorDespesa) as DespesaTotal  
From Veiculo Join Despesa on  
Veiculo.IDVeiculo=Despesa.VeiculoIDVeiculo  
Where Year (DataPagDes) = '2022'  
Group by Veiculo.IDVeiculo, Veiculo.MatriculaVeiculo  
ORDER BY DespesaTotal DESC;
```

--9. Quais os modelos que foram fornecidos pela 'Auto Supplies Ltda' e 'Carros e Peças'

```

Select ModeloVeiculo.IDModelo, DescModelo
From ModeloVeiculo Join Veiculo on ModeloVeiculo.IDModelo =
Veiculo.ModeloVeiculoIDModelo
Join Fornecedor on Veiculo.FornecedorIDFornecedor =
Fornecedor.IDFornecedor
Where NomeFornecedor = 'Auto Supplies Ltda'
Intersect
Select ModeloVeiculo.IDModelo, DescModelo
From ModeloVeiculo Join Veiculo on ModeloVeiculo.IDModelo =
Veiculo.ModeloVeiculoIDModelo
Join Fornecedor on Veiculo.FornecedorIDFornecedor =
Fornecedor.IDFornecedor
Where NomeFornecedor = 'Carros e Peças'

```

--10. Qual a média de classificação de cada marca (deve calcular primeiro a média por veiculo, e só depois por marca)

```

SELECT DescMarca, AVG(MediaA) AS MediaPorMarca
FROM (
    SELECT
        Veiculo.IDVeiculo,
        MarcaVeiculo.DescMarca,
        AVG(Feedback.Classificacao) AS MediaA
    FROM Veiculo
    JOIN ModeloVeiculo
        ON Veiculo.ModeloVeiculoIDModelo = ModeloVeiculo.IDModelo
    JOIN MarcaVeiculo
        ON ModeloVeiculo.MarcaVeiculoIDMarca =
        MarcaVeiculo.IDMarca
    JOIN Aluguer
        ON Veiculo.IDVeiculo = Aluguer.VeiculoIDVeiculo
    JOIN Feedback
        ON Feedback.AluguerIDAluguer = Aluguer.IDAluguer
    GROUP BY Veiculo.IDVeiculo, MarcaVeiculo.DescMarca
) Media
GROUP BY Media.DescMarca

```

5. Aplicação com acesso e manipulação a Base de Dados

5.1 Descrição da aplicação desenvolvida

A aplicação desenvolvida é uma solução de gestão baseada em Windows Forms utilizando a linguagem C# e conectada a nossa base de dados SQL Server. Esta aplicação permite realizar operações CRUD (Criar, Ler, Atualizar, Eliminar) em várias entidades relacionadas, como clientes, contactos do mesmo, veículos e seguros.

5.1.1 Funcionalidades Principais

1. Inserir Cliente

- Inserção de novos clientes com informações como nome, NIF, morada e tipo de cliente.
- Associação de contactos a clientes.

2. Atualizar dados do Cliente

- Atualização dos dados de clientes.

3. Eliminação de Veículos

- Consulta de veículos registados na base de dados.
- Eliminação de veículos através de um *stored procedure*.

4. Pesquisa de Seguros

- Carregamento de informações sobre seguros e seguradoras.
- Associação de seguros a veículos.
- Pesquisa de seguros filtrados por seguradora.
- Os filtros de pesquisa ajudam a limitar os resultados apresentados, facilitando a localização de informações relevantes.

5.1.2 Funcionalidades Adicionais

A aplicação inclui diversas funcionalidades adicionais para otimizar a experiência do utilizador:

1. Validação de Dados:

- A aplicação verifica a validade dos dados inseridos antes de os enviar para a base de dados, prevenindo erros comuns.

2. Mensagens de Feedback:

- Após cada operação, o utilizador recebe uma mensagem de confirmação ou de erro, conforme o resultado da operação.

3. Listas Suspensas Dinâmicas:

- As listas suspensas (*ComboBox*) são atualizadas dinamicamente com os dados mais recentes da base de dados, garantindo que o utilizador tem sempre acesso à informação atualizada.

5.2 Interface(s) da aplicação

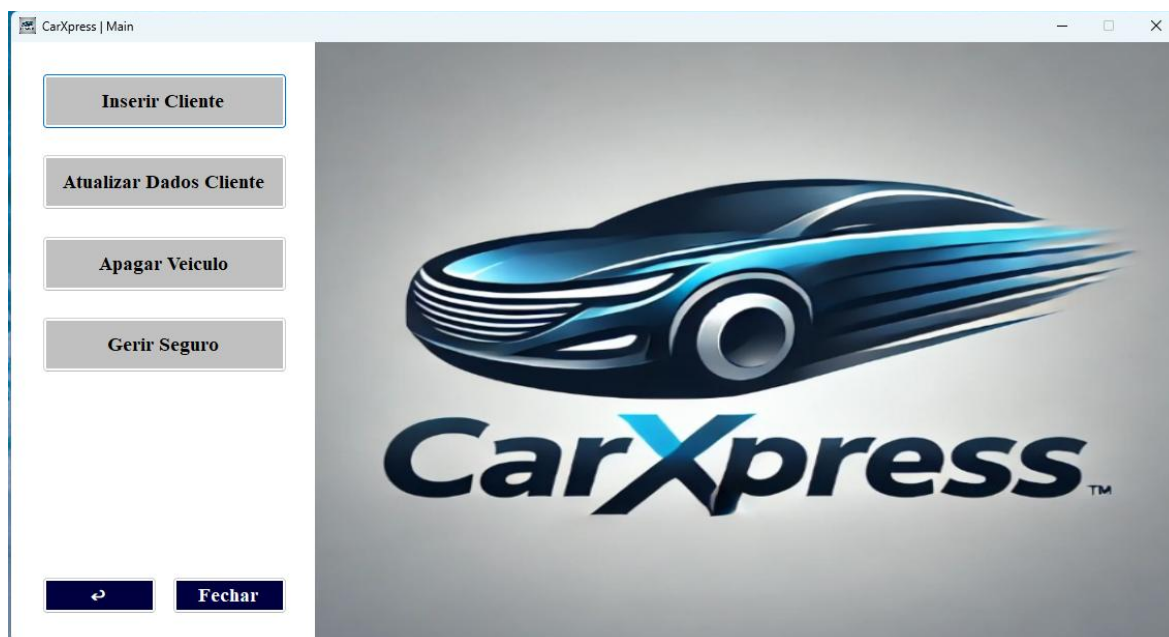


Figura 3 - Interface Inicial

	IDCliente	NomeCliente	DataNascCliente	NIFCliente	RuaCliente	TipoClienteIDTC	CodigoPo
▶	1	Paulo Almeida Fa...	10/04/1992	112223334	Rua das Palmeira...	1	4700300
	2	Helena Ferreira	23/11/1998	223344556	Av. do Litoral, 45	6	4700300
	3	Paulo Almeida	10/04/1992	112223334	Rua das Palmeira...	1	4700300
	4	Carla Martins Ave...	30/09/1970	445566778	Cristiano Renaldo...	8	4710090
	5	Rita Fonecas	04/02/1948	556677889	Largo da Paz, 50	9	4750700
	6	Empresa ABC		667788990	Parque Industrial, 1	10	4720300

Figura 4 - Interface 'Inserir Cliente'

CarXpress | Main

Inserir Cliente

Atualizar Dados Cliente

Apagar Veículo

Gerir Seguro

Nome
Paulo Almeida Faustino

NIF
112223334

Tipo Cliente
Particular

Rua
Rua das Palmeiras, 200

Localidade
São Vitor, Braga

Data Nascimento
10 de abril de 1992

IDCliente: 1 **Atualizar**

IDCliente	NomeCliente	DataNascCliente	NIFCliente	RuaCliente	TipoClienteIDTC	CodigoPostalC
1	Paulo Almeida Fa...	10/04/1992	112223334	Rua das Palmeira...	1	4700300
2	Helena Ferreira	23/11/1998	223344556	Av. do Litoral, 45	6	4700300
3	Paulo Almeida	10/04/1992	112223334	Rua das Palmeira...	1	4700300
4	Carla Martins Ave...	30/09/1970	445566778	Cristiano Renaldo...	8	4710090
5	Rita Fonecas	04/02/1948	556677889	Largo da Paz, 50	9	4750700
6	Empresa ABC	667788990	667788990	Parque Industrial, 1	10	4720300
7	Startup XYZR	10/04/1992	778899001	Rua novação, 300	4	4710090
8	Distribuidora Cent...		889900112	Rua dos Fomece...	12	4840080

↩ Fechar

Figura 5 - Interface 'Atualizar Dados Cliente'

CarXpress | Main

Inserir Cliente

Atualizar Dados Cliente

Apagar Veículo

Gerir Seguro

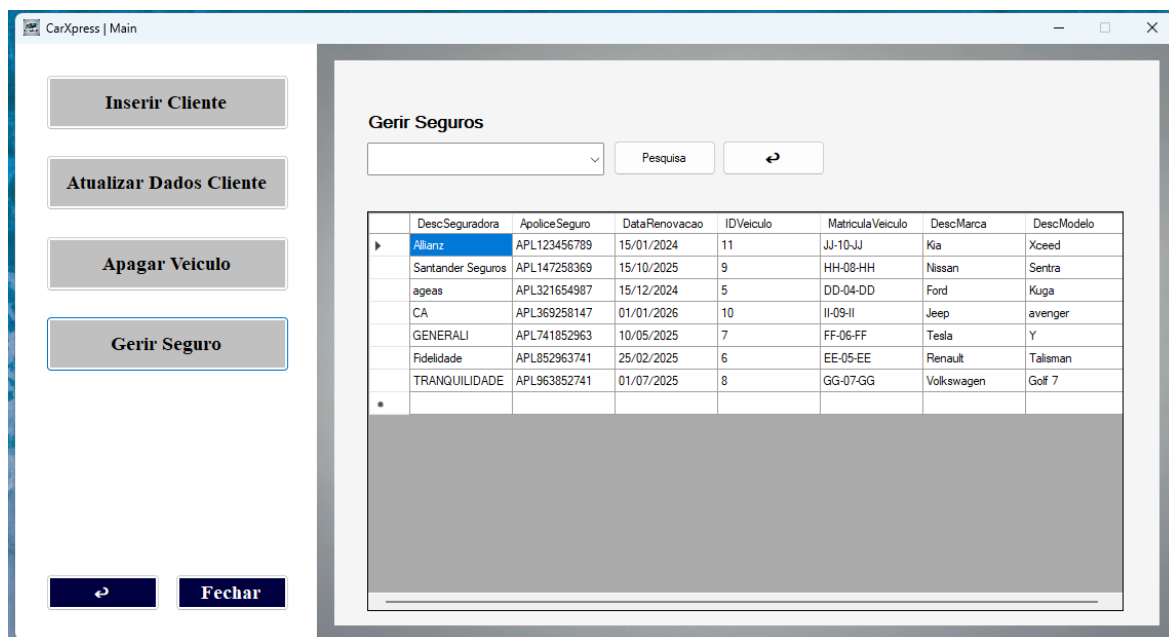
Insira o IDVeículo

Delete

IDVeículo	MatriculaVeiculo	LotacaoVeiculo	TaraVeiculo	DescCor	DataLegal	DataFabricacao
1	AA-16-PP	5	1600	Azul	01/06/2023	15/03/2023
5	DD-04-DD	4	1800	Vermelho	10/12/2020	01/10/2020
6	EE-05-EE	5	1400	Prata	07/07/2019	01/05/2019
7	FF-06-FF	5	1600	Branco	01/05/2023	01/03/2023
8	GG-07-GG	5	1400	Cinza	01/04/2021	01/11/2020
9	HH-08-HH	5	1250	Preto	10/10/2020	01/07/2020
10	II-09-II	5	2000	Verde	15/05/2018	01/02/2018
11	JJ-10-JJ	5	1350	Azul Escuro	01/11/2021	01/07/2021
12	KK-11-KK	7	2500	Prata	01/09/2015	01/06/2015
13	LL-12-LL	5	1250	Amarelo	01/03/2022	01/12/2021
14	MM-13-MM	4	1600	Vermelho	01/10/2022	01/08/2022
15	NN-14-NN	5	1400	Cinza	01/07/2020	01/04/2020
*						

↩ Fechar

Figura 6 - Interface 'Apagar Veículo'

*Figura 7 - Interface 'Gerir Seguro'*

5.3 Descrição da forma de interação com a Base de dados

A interação com a base de dados é realizada através de consultas SQL executadas pela aplicação em diferentes momentos. A aplicação utiliza objetos da biblioteca `System.Data.SqlClient` para estabelecer a conexão e realizar as operações necessárias.

5.3.1 Conexão com a Base de Dados

A aplicação utiliza uma *string* de conexão configurada para se ligar ao servidor SQL Server local.

```
string connectionString = "Data Source=ASUS-FC;Initial Catalog=AAD-22997-23008-23015;Integrated Security=True;";
```

5.3.2 Consultas SQL e Preenchimento de Dados

As consultas SQL são utilizadas para carregar dados das tabelas relevantes. Os dados são carregados e em seguida preenchidos em controlos da interface gráfica, como `DataGridView` e `ComboBox`.

```
string query = "SELECT * FROM Cliente";

try
{
    // conexão a base de dados
    using (SqlConnection connection = new SqlConnection(connectionString))
    {
        SqlDataAdapter adapter = new SqlDataAdapter(query, connection);
        DataTable dataTable = new DataTable();
        adapter.Fill(dataTable);
        // Preenche o DataGridView com os dados dos clientes
        dataGridView1.DataSource = dataTable;
    }
}
catch (Exception ex)
{
    MessageBox.Show("Erro ao carregar os dados: " + ex.Message);
}
```

5.3.3 Comandos SQL Parametrizados

Para inserir, atualizar e eliminar, são utilizados comandos SQL parametrizados, que ajudam a evitar vulnerabilidades de injeção de SQL. Estes comandos permitem a passagem segura de valores para as consultas.

Exemplo de inserção de um cliente:

```
string query = "INSERT INTO Cliente (NomeCliente, NIFCliente, RuaCliente,
CodigoPostalCP, TipoClienteIDTC, DataNascCliente) OUTPUT INSERTED.IDCliente
VALUES (@Name, @NIF, @Address, @Cp, @TipoClienteId, @BirthDate)";

using (SqlConnection connection = new SqlConnection(connectionString))
{
    SqlCommand command = new SqlCommand(query, connection);
    command.Parameters.AddWithValue("@Name", name);
    command.Parameters.AddWithValue("@NIF", nif);
    command.Parameters.AddWithValue("@Address", address);
    command.Parameters.AddWithValue("@Cp", cp);
    command.Parameters.AddWithValue("@TipoClienteId", tipoClienteId);
    command.Parameters.AddWithValue("@BirthDate", birthDate); // Passa a data
de nascimento

    connection.Open();
    int clientId = (int)command.ExecuteScalar(); // Retorna o ID do cliente
inserido
    return clientId;
}
```

5.3.4 Stored Procedure

A aplicação utiliza também *Stored Procedure* (Um *stored procedure* é um conjunto de instruções SQL que é armazenado na base de dados e pode ser executado repetidamente, como uma função ou script pré-definido. É utilizado realizar operações complexas ou simplesmente reduzir a repetição de código SQL).

Esta abordagem garante que a aplicação interage com a base de dados de forma estruturada, eficiente e principalmente **segura**.

Exemplo de chamada de um *Stored Procedure* no código da aplicação:

```
// Chama o procedimento armazenado para deletar o veículo
using (SqlCommand cmd = new SqlCommand("usp_DeleteVeiculo", connection))
{
    cmd.CommandType = CommandType.StoredProcedure;
    cmd.Parameters.AddWithValue("@VeiculoID", veiculoID);

    cmd.ExecuteNonQuery();
}
```

Stored Procedure na Base de Dados:

```

-- Procedimento Armazenado para Apagar Veículo e Registos Relacionados
CREATE PROCEDURE usp_DeleteVeiculo
    @VeiculoID INT
AS
BEGIN
    SET NOCOUNT ON;

    BEGIN TRY
        BEGIN TRANSACTION;

        -- Verifica se o veículo existe
        IF NOT EXISTS (SELECT 1 FROM Veiculo WHERE IDVeiculo = @VeiculoID)
        BEGIN
            RAISERROR('Veículo com ID %d não existe.', 16, 1, @VeiculoID);
            ROLLBACK TRANSACTION;
            RETURN;
        END

        -- Exclui infrações relacionadas aos alugueres do veículo
        DELETE FROM Infracoes
        WHERE AluguerIDAluguer IN (
            SELECT IDAluguer FROM Aluguer WHERE VeiculoIDVeiculo = @VeiculoID
        );
        -- Exclui alugueres do veículo
        DELETE FROM Aluguer WHERE VeiculoIDVeiculo = @VeiculoID;
        -- Exclui despesas relacionadas ao veículo
        DELETE FROM Despesa WHERE VeiculoIDVeiculo = @VeiculoID;
        -- Exclui seguros relacionados ao veículo
        DELETE FROM Seguro WHERE VeiculoIDVeiculo = @VeiculoID;
        -- Exclui manutenções relacionadas ao veículo
        DELETE FROM Manutencao WHERE VeiculoIDVeiculo = @VeiculoID;
        -- Finalmente, exclui o veículo da tabela Veiculo
        DELETE FROM Veiculo WHERE IDVeiculo = @VeiculoID;

        COMMIT TRANSACTION;
    END TRY
    BEGIN CATCH
        IF @@TRANCOUNT > 0
            ROLLBACK TRANSACTION;

        DECLARE @ErrorMessage NVARCHAR(4000) = ERROR_MESSAGE();
        DECLARE @ErrorSeverity INT = ERROR_SEVERITY();
        DECLARE @ErrorState INT = ERROR_STATE();

        RAISERROR (@ErrorMessage, @ErrorSeverity, @ErrorState);
    END CATCH
END

```

Este *Stored Procedure*, chamado *usp_DeleteVeiculo*, é uma ferramenta para eliminar um veículo da base dados. Ele não apenas remove o veículo, mas também todos os dados relacionados a ele em outras tabelas, como alugueres, infrações, despesas, seguros e manutenções. Tudo isto é feito de forma organizada e **segura**.

Anexos

Anexo I – Diagrama ER (Tamanho A3)

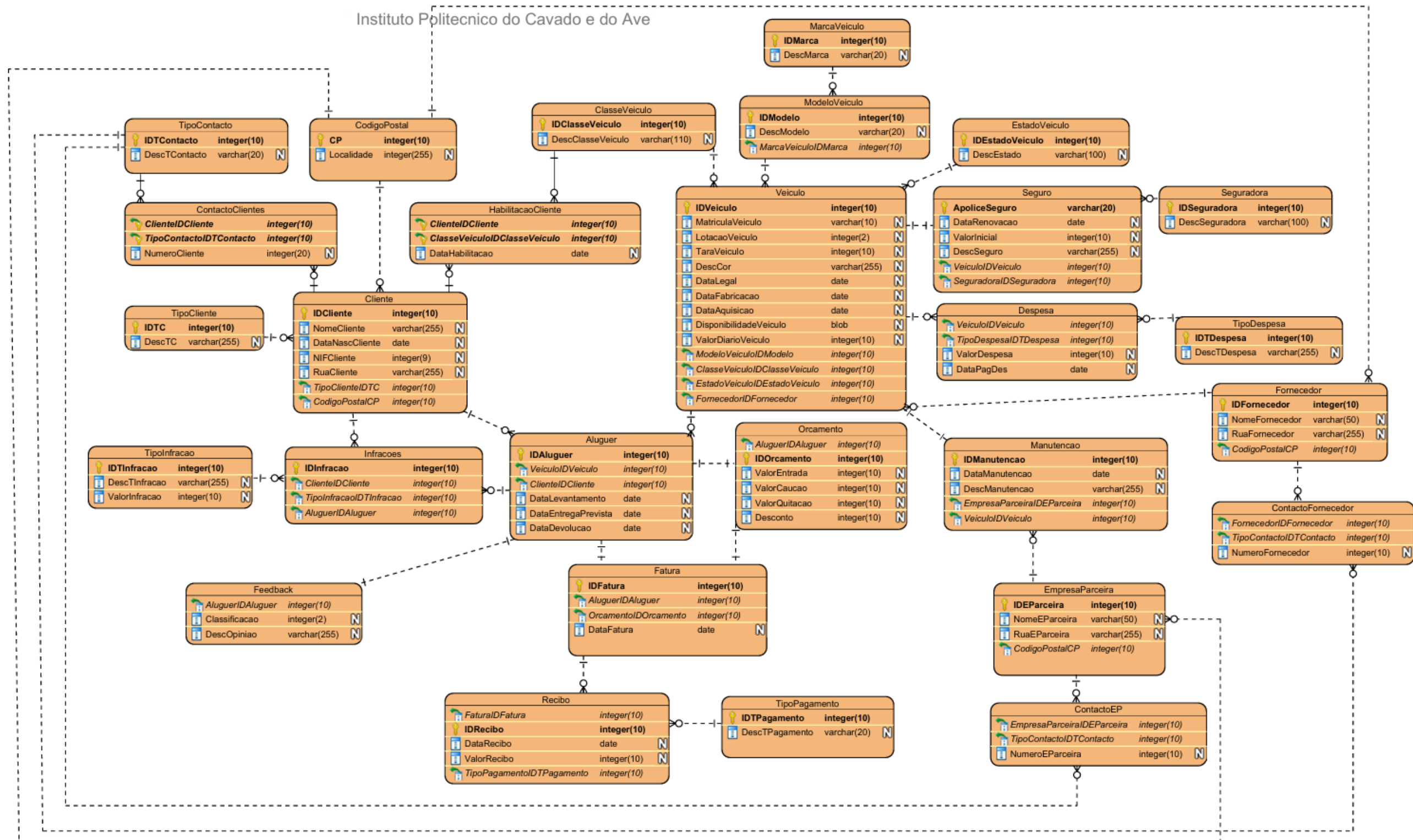


Figura 8 - Diagrama ER (TamanhoA3)

Anexo II– Diagrama ER Corrigido (Tamanho A3)

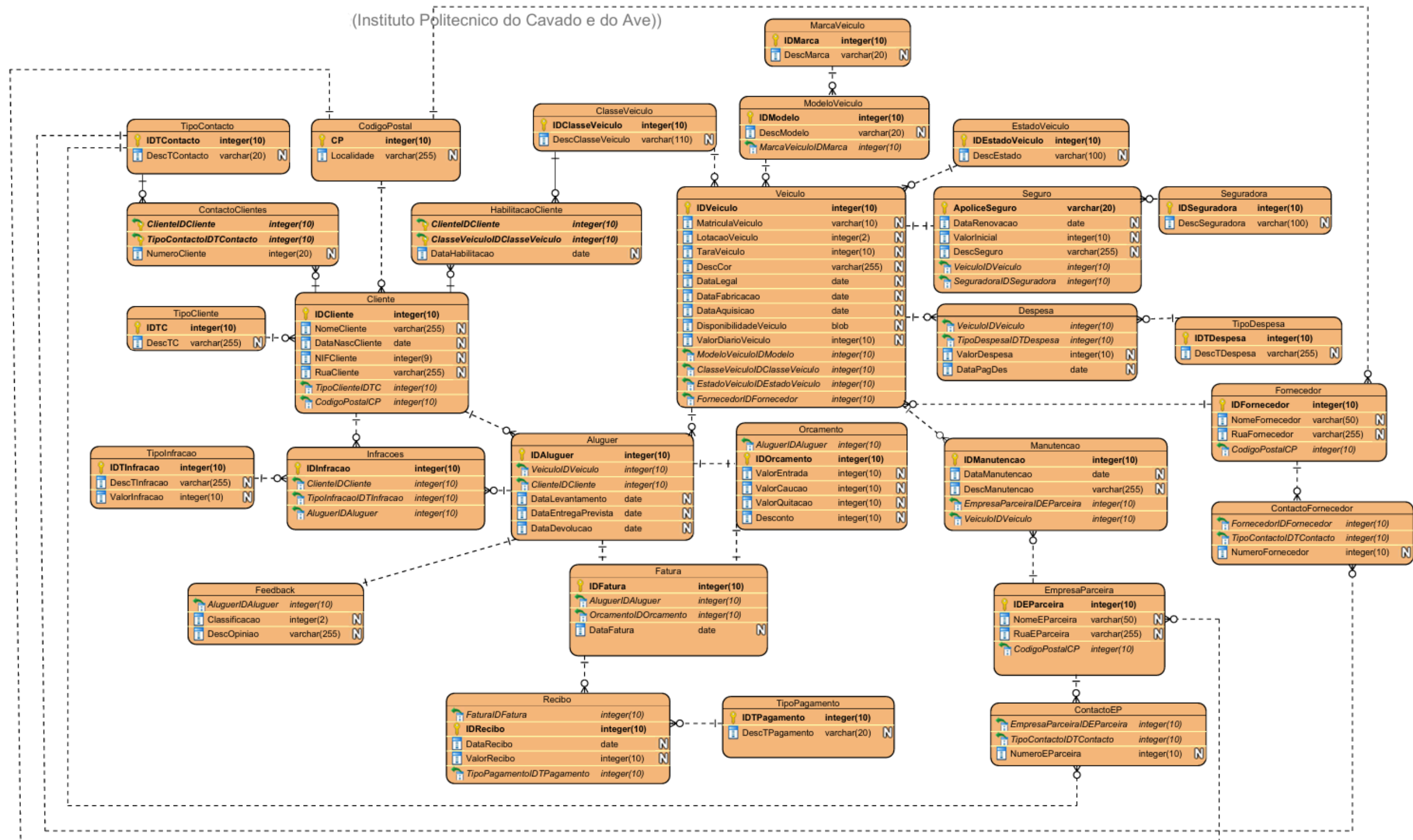


Figura 9 - Diagrama ER Corrigido (Tamanho A3)