

# Relatório Final

Bases de Dados

Universidade de Aveiro

Daniel Pedrinho N<sup>o</sup>107378 (50%)

Tomás Laranjo N<sup>o</sup>107542 (50%)

P9G4



# Índice

<b>1</b>	<b>Introdução</b>	<b>2</b>
<b>2</b>	<b>Análise de Requisitos</b>	<b>3</b>
2.1	Requisitos Funcionais . . . . .	3
2.2	Requisitos Não-Funcionais . . . . .	5
<b>3</b>	<b>Diagramas</b>	<b>6</b>
3.1	Diagrama Entidade-Relacionamento (DER) . . . . .	6
3.2	Esquema Relacional (ER) . . . . .	7
<b>4</b>	<b>SQL</b>	<b>8</b>
4.1	Data Definition Language . . . . .	8
4.2	Data Manipulation Language . . . . .	11
4.3	Stored Procedures . . . . .	13
4.4	Views . . . . .	13
4.5	Triggers & Índices . . . . .	13
<b>5</b>	<b>Windows Forms App</b>	<b>14</b>
5.1	Form Principal . . . . .	14
5.2	Form de Adição . . . . .	15
5.3	Form de Edição . . . . .	16
<b>6</b>	<b>Utilização</b>	<b>17</b>

# Capítulo 1

## Introdução

No âmbito da unidade curricular de Bases de Dados, pertencente à Licenciatura foi realizado um projeto, cujo objetivo deste trabalho consistiu na implementação de um sistema suportado por uma base de dados. O grupo decidiu por desenvolver um sistema informático para uma oficina de reparações de máquinas industriais, que, além de gerir o inventário e os recursos humanos, também seria utilizado para registar as operações de negócio.

## Capítulo 2

# Análise de Requisitos

### 2.1 Requisitos Funcionais

#### 1. Registrar um Pedido de Manutenção

- Permitir a inserção de novos pedidos de manutenção na base de dados,
- Associar cada pedido a uma máquina específica e incluir detalhes como descrição, data de início e data de fim.

#### 2. Atribuir um funcionário a um pedido

- Permitir a designação de funcionários específicos para cada pedido de manutenção,
- Garantir que cada funcionário atribuído a um pedido está registado na base de dados.

#### 3. Registrar a Conclusão de um Pedido

- Permitir a atualização do estado de um pedido de manutenção para "concluído",
- Registrar a data de conclusão e quaisquer observações finais sobre a manutenção realizada.

#### 4. Consultar o Histórico de Pedidos

- Disponibilizar uma funcionalidade para visualizar todos os pedidos de manutenção realizados,
- Incluir filtros e critérios de pesquisa para facilitar a consulta.

#### 5. Consultar o Histórico de Pedidos de um funcionário

- Permitir a visualização de todos os pedidos de manutenção atribuídos a um funcionário específico,

- Incluir detalhes como datas, máquinas envolvidas e estados dos pedidos.

#### **6. Consultar o Histórico de Pedidos de uma Máquina**

- Permitir a visualização de todos os pedidos de manutenção associados a uma máquina específica,
- Incluir detalhes sobre as manutenções realizadas, funcionários envolvidos e estado dos pedidos.

#### **7. Consultar o Histórico de Pedidos de um Supervisor**

- Permitir a visualização de todos os pedidos de manutenção supervisionados por um supervisor específico,
- Incluir detalhes como funcionários envolvidos, máquinas, estados dos pedidos e datas.

#### **8. Consultar o Histórico de Pedidos de uma Peça**

- Permitir a visualização de todos os pedidos de manutenção que envolveram uma peça específica,
- Incluir detalhes sobre quantidades utilizadas, estado das peças e máquinas onde foram aplicadas.

#### **9. Consultar o Histórico de Pedidos de um Fornecedor**

- Permitir a visualização de todos os pedidos de manutenção que envolveram peças ou serviços de um fornecedor específico,
- Incluir detalhes sobre os materiais fornecidos, quantidades, e estado dos pedidos.

#### **10. Consultar o Histórico de Pedidos de uma Compra**

- Permitir a visualização de todos os pedidos de manutenção relacionados a uma compra específica,
- Incluir detalhes sobre as peças adquiridas, quantidades e fornecedor.

## 2.2 Requisitos Não-Funcionais

### 1. Performance

- A base de dados deve responder a consultas e inserções em tempo razoável, mesmo com um grande volume de dados,
- Otimizar índices para melhorar a velocidade das operações de leitura e escrita.

### 2. Escalabilidade

- A arquitetura da base de dados deve permitir a expansão para acomodar um aumento no volume de dados e número de utilizadores,
- Estruturar as tabelas de modo a facilitar a integração futura com outros sistemas.

### 3. Confiabilidade e Disponibilidade

- Garantir alta disponibilidade da base de dados com mecanismos de backup e recuperação de desastres,
- Minimizar o tempo de inatividade com manutenção e atualizações planeadas.

### 4. Usabilidade

- Desenvolver interface de consulta intuitivas e amigáveis par os utilizadores finais,
- Fornecer documentação clara e detalhada para a utilização e manutenção do sistema.

### 5. Compatibilidade

- Assegurar que a base de dados seja compatível com diferentes sistemas operativos e ambientes de desenvolvimento,
- Utilizar padrões abertos e amplamente aceites para garantir a interoperabilidade com outros sistemas.

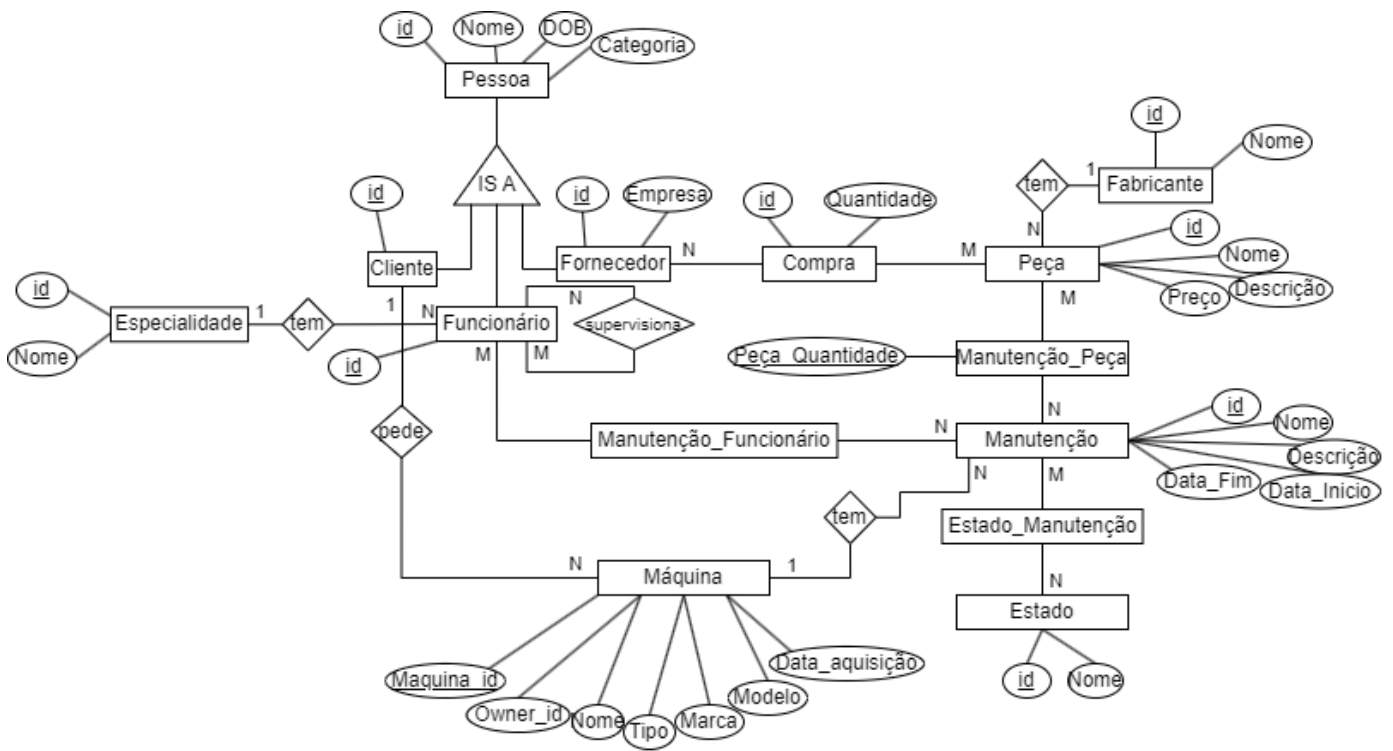
### 6. Consistência dos Dados

- Implementar restrições e validações para manter a integridade e a consistência dos dados,
- Garantir que as transações sejam tratadas de forma a preservar a integridade dos dados mesmo em casos de falha do sistema.

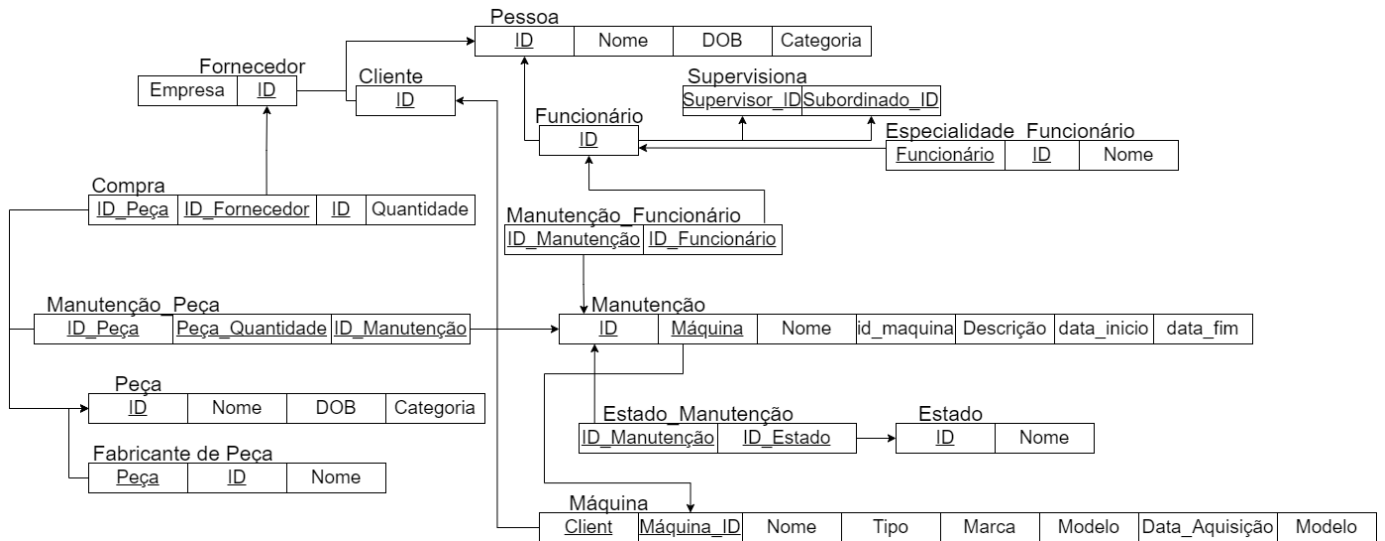
## Capítulo 3

# Diagramas

### 3.1 Diagrama Entidade-Relacionamento (DER)



## 3.2 Esquema Relacional (ER)





## Capítulo 4

# SQL

### 4.1 Data Definition Language

```
1 DROP TABLE IF EXISTS Estado_Manutencao;
2 DROP TABLE IF EXISTS Manutencao_Funcionario;
3 DROP TABLE IF EXISTS Manutencao_Peca;
4 DROP TABLE IF EXISTS Manutencao;
5 DROP TABLE IF EXISTS Compra;
6 DROP TABLE IF EXISTS Maquina;
7 DROP TABLE IF EXISTS Funcionario;
8 DROP TABLE IF EXISTS Fornecedor;
9 DROP TABLE IF EXISTS Cliente;
10 DROP TABLE IF EXISTS Especialidade;
11 DROP TABLE IF EXISTS Estado;
12 DROP TABLE IF EXISTS Peca;
13 DROP TABLE IF EXISTS Fabricante;
14 DROP TABLE IF EXISTS Pessoa;
15
16 CREATE TABLE Pessoa(
17     id int primary key not null,
18     nome varchar(150) not null,
19     date_birth date not null,
20     categoria varchar(150) not null
21 );
22
23 CREATE TABLE Especialidade(
24     id int primary key,
25     nome varchar(150) not null
26 );
27
28 CREATE TABLE Fornecedor(
29     id int primary key,
30     empresa varchar(150),
31
32     foreign key (id) references Pessoa(id)
33 );
34
35 CREATE TABLE Cliente(
36     id int primary key,
```

```

37         foreign key (id) references Pessoa(id)
38     );
39
40
41 CREATE TABLE Funcionario(
42     id int primary key,
43     id_especialidade int,
44     id_supervisor int,
45
46     foreign key (id) references Pessoa(id),
47     foreign key (id_especialidade) references Especialidade(id),
48     foreign key (id_supervisor) references Funcionario(id)
49 );
50
51 CREATE TABLE Maquina(
52     maquina_id int primary key not null,
53     nome varchar(150) not null,
54     tipo varchar(150) not null,
55     marca varchar(150) not null,
56     modelo varchar(150) not null,
57     data_aquisicao date not null,
58     owner_id int not null,
59     foreign key (owner_id) references Pessoa(id)
60 );
61
62 CREATE TABLE Fabricante(
63     id int primary key,
64     nome varchar(150) not null
65 );
66
67 CREATE TABLE Peca(
68     id int primary key,
69     nome varchar(150) not null,
70     descricao varchar(150) not null,
71     preco float not null,
72     id_fabricante int not null,
73
74     foreign key (id_fabricante) references Fabricante(id)
75 );
76
77 CREATE TABLE Compra(
78     id int primary key,
79     id_fornecedor int,
80     id_pecas int,
81     quantidade int,
82
83     foreign key (id_fornecedor) references Pessoa(id),
84     foreign key (id_pecas) references Peca(id)
85 );
86
87 CREATE TABLE Estado (
88     id int primary key,
89     nome varchar(150) not null
90 );
91
92 CREATE TABLE Manutencao(
93     id int primary key identity(1,1),

```

```

94     nome varchar(150) not null,
95     id_maquina int not null,
96     descricao varchar(150) not null,
97     data_inicio date not null,
98     data_fim date
99 );
100
101 CREATE TABLE Estado_Manutencao(
102     id_manutencao int,
103     id_estado int,
104
105     foreign key (id_manutencao) references Manutencao(id),
106     foreign key (id_estado) references Estado(id)
107 );
108
109 CREATE TABLE Manutencao_Funcionario(
110     id_manutencao int,
111     id_funcionario int,
112
113     foreign key (id_manutencao) references Manutencao(id),
114     foreign key (id_funcionario) references Funcionario(id)
115 );
116
117 CREATE TABLE Manutencao_Peca (
118     id_manutencao int,
119     id_peca int,
120     peca_quantidade int,
121
122     foreign key (id_manutencao) references Manutencao(id),
123     foreign key (id_peca) references Peca(id)
124 );

```

## 4.2 Data Manipulation Language

Nesta secção, apresentamos apenas uma pequena parte das inserções iniciais.

```
1      insert into Pessoa (id, nome, date_birth, categoria) values (1,
      'Geri Juris', '1993-01-15', 'Funcionario');
2 insert into Pessoa (id, nome, date_birth, categoria) values (2, '
      Abel Agneau', '2002-11-05', 'Cliente');
3 insert into Pessoa (id, nome, date_birth, categoria) values (3, '
      Rebeka Annear', '1984-11-06', 'Fornecedor');
4
5 INSERT INTO Especialidade(id, nome) VALUES
6 (1, 'Revisor'),
7 (2, 'Eletricista'),
8 (3, 'Pintor'),
9
10 INSERT INTO Fabricante(id, nome) VALUES
11 (1, 'Boschh'),
12 (2, 'Makita'),
13 (3, 'DeWalt'),
14
15 -- Insert Fornecedor
16 DECLARE @id INT = 1;
17 WHILE @id <= 100
18 BEGIN
19     INSERT INTO Fornecedor(id) SELECT id FROM Pessoa WHERE
      categoria = 'Fornecedor' AND id = @id;
20
21     -- Set empresa
22     IF @id % 3 = 0
23     BEGIN
24         UPDATE Fornecedor SET empresa = 'Fornecedor ' + CAST((@id /
      3) AS VARCHAR) WHERE id = @id;
25     END
26
27     SET @id = @id + 1;
28 END
29
30 INSERT INTO Cliente(id) SELECT id FROM Pessoa WHERE categoria = '
      Cliente'
31
32 -- Insert Funcionario
33 DECLARE @id_kek INT = 1;
34 WHILE @id_kek <= 100
35 BEGIN
36     INSERT INTO Funcionario(id) SELECT id FROM Pessoa WHERE
      categoria = 'Funcionario' AND id = @id_kek;
37
38     -- Set id_especialidade and id_supervisor
39     IF @id_kek % 3 = 1
40     BEGIN
41         UPDATE Funcionario SET id_especialidade = 1 WHERE id =
      @id_kek;
42     END
43     ELSE IF @id_kek % 3 = 2
44     BEGIN
45         UPDATE Funcionario SET id_especialidade = 3 WHERE id =
      @id_kek;
```

```

46     END
47     ELSE
48     BEGIN
49         UPDATE Funcionario SET id_especialidade = 2 WHERE id =
        @id_kek;
50     END
51
52     IF @id_kek % 4 = 0
53     BEGIN
54         UPDATE Funcionario SET id_supervisor = 1 WHERE id = @id_kek
        ;
55     END
56     ELSE IF @id_kek % 10 = 0
57     BEGIN
58         UPDATE Funcionario SET id_supervisor = 7 WHERE id = @id_kek
        ;
59     END
60
61     SET @id_kek = @id_kek + 1;
62 END
63
64
65 INSERT INTO Maquina(maquina_id, nome, tipo, marca, modelo,
        data_aquisicao, owner_id) VALUES
66 (1, 'Solda 1', 'Tipo 1', 'Marca 1', 'Modelo 1', '2010-01-01', 2),
67 (2, 'Esmagador 1', 'Tipo 2', 'Marca 2', 'Modelo 2', '2011-02-02',
        6),
68 (3, 'Pintura 1', 'Tipo 3', 'Marca 3', 'Modelo 3', '2012-03-03', 15)
        ,
69 (4, 'Solda 2', 'Tipo 1', 'Marca 1', 'Modelo 1', '2013-04-04', 17)

```

## 4.3 Stored Procedures

Implementámos uma *Stored Procedure* para a deleção da tabela de Manutenção, pois esta está ligada a 3 outras tabelas.

Como forma de agilizar a deleção, a *Stored Procedure* apaga das 4 tabelas simultaneamente.

```
1      GO
2  CREATE PROCEDURE delete_manutencao @id int
3  AS
4  BEGIN
5      DELETE FROM Estado_Manutencao WHERE id_manutencao = @id;
6      DELETE FROM Manutencao_Funcionario WHERE id_manutencao = @id;
7      DELETE FROM Manutencao_Peca WHERE id_manutencao = @id;
8      DELETE FROM Manutencao WHERE id = @id;
9  END
```

## 4.4 Views

Para o *SELECT* mais complexo que é usado, foi criada uma view para agilizar a seleção e manipulação dos dados, que seleciona todas as informações relevantes à apresentação de dados na interface.

```
1  CREATE VIEW Manuten_oView AS
2  SELECT TOP (100) PERCENT dbo.Manutencao.id, dbo.Manutencao.
   nome, dbo.Manutencao.data_inicio, dbo.Manutencao_Peca.
   id_manutencao, dbo.Manutencao_Peca.id_pecas, dbo.
   Manutencao_Funcionario.id_manutencao AS Expr1,
3      dbo.Manutencao_Funcionario.id_funcionario, dbo.
   Estado_Manutencao.id_manutencao AS Expr2, dbo.
   Estado_Manutencao.id_estado
4  FROM      dbo.Manutencao INNER JOIN
5      dbo.Manutencao_Peca ON dbo.Manutencao.id = dbo.
   Manutencao_Peca.id_manutencao INNER JOIN
6      dbo.Manutencao_Funcionario ON dbo.Manutencao.id
   = dbo.Manutencao_Funcionario.id_manutencao INNER JOIN
7      dbo.Estado_Manutencao ON dbo.Manutencao.id =
   dbo.Estado_Manutencao.id_manutencao
8  ORDER BY dbo.Manutencao.nome
```

## 4.5 Triggers & Índices

Não achamos necessários a criação de índices pois todos os *SELECT*'s são únicos, não havendo o uso do mesmo *SELECT* em mais que uma localização.

De igual modo, a única operação de deleção é realizada exclusivamente através de uma *Stored Procedure*, dispensando, portanto, a utilização de *Triggers*.

## Capítulo 5

# Windows Forms App

### 5.1 Form Principal

No *Form* principal, apresentamos duas tabelas. A da esquerda apresenta todos os funcionários, clientes, fornecedores e máquinas, bem como um filtro para apenas disponibilizar os funcionários e máquinas que estão a ser utilizados em manutenções ativas. Também podem ser marcadas como completas manutenções existentes, e podem ser pesquisadas manutenções específicas pelo nome dado.

Na tabela da esquerda apresentamos todas as manutenções existentes, com um filtro para aprensentar as manutenções em progresso.

ID	Name	DOB	Especialidade	Supervisor
33	Aldin Palant	26/03/1983	Eletricista	
58	Anastasia Woodhouse	18/02/1996	Revisor	
94	Anetta Adolphe	10/07/1989	Revisor	
78	Avri Vedriyev	04/08/1982	Eletricista	
57	Bo Ridgwell	19/12/1987	Eletricista	
43	Daloris Chat	29/07/1996	Revisor	
40	Delaney Copyn	15/11/1997	Revisor	Gerl Juts
95	Des Tresler	27/02/1989	Pintor	
52	Devland Scholtz	09/06/1997	Revisor	Gerl Juts
23	Donalyn McMichan	08/08/1998	Pintor	
12	Douglas Tidcombe	12/09/1997	Eletricista	Gerl Juts
16	Duky Brandenburg	14/04/2002	Revisor	Gerl Juts
31	Eim Galland	28/09/1982	Revisor	
24	Florentia Fam	11/11/1986	Eletricista	Gerl Juts
44	Frankie Ytkovitz	01/07/1982	Pintor	Gerl Juts
13	Gan Johantiges	21/08/1997	Revisor	
1	Gerl Juts	15/01/1993	Revisor	
37	Haywood Ayllett	31/01/1997	Revisor	
49	Janet Toop	26/03/1995	Revisor	
72	Karlen Trennell	25/11/1980	Eletricista	Gerl Juts
71	Kikela Mison	05/09/1999	Pintor	
98	Lucinda Kingston	04/08/2001	Pintor	
88	Myer Inchan	27/06/1986	Revisor	Gerl Juts
56	Paolo Peters	19/07/1996	Pintor	Gerl Juts
74	Rhona Danelut	26/04/1986	Pintor	
75	Riccardo Swallu	07/03/1987	Eletricista	

Nome	Maquina	Pecas	Funcionarios	Estado	
teste 1	Solda 2	2	3	Em progresso	X
teste 2	Esmagador 5	2	6	Em progresso	X

☐ Display Only In Use ☐ Display Only In Progress

## 5.2 Form de Adição

Este *Form* serve como mediador para adicionar novas manuntenções. Seleciona-se a máquina a manter, peças a usar, e funcionários a trabalharem.

The image shows a software window titled "Mediador" with standard Windows window controls (minimize, maximize, close). The form is divided into several sections for data entry:

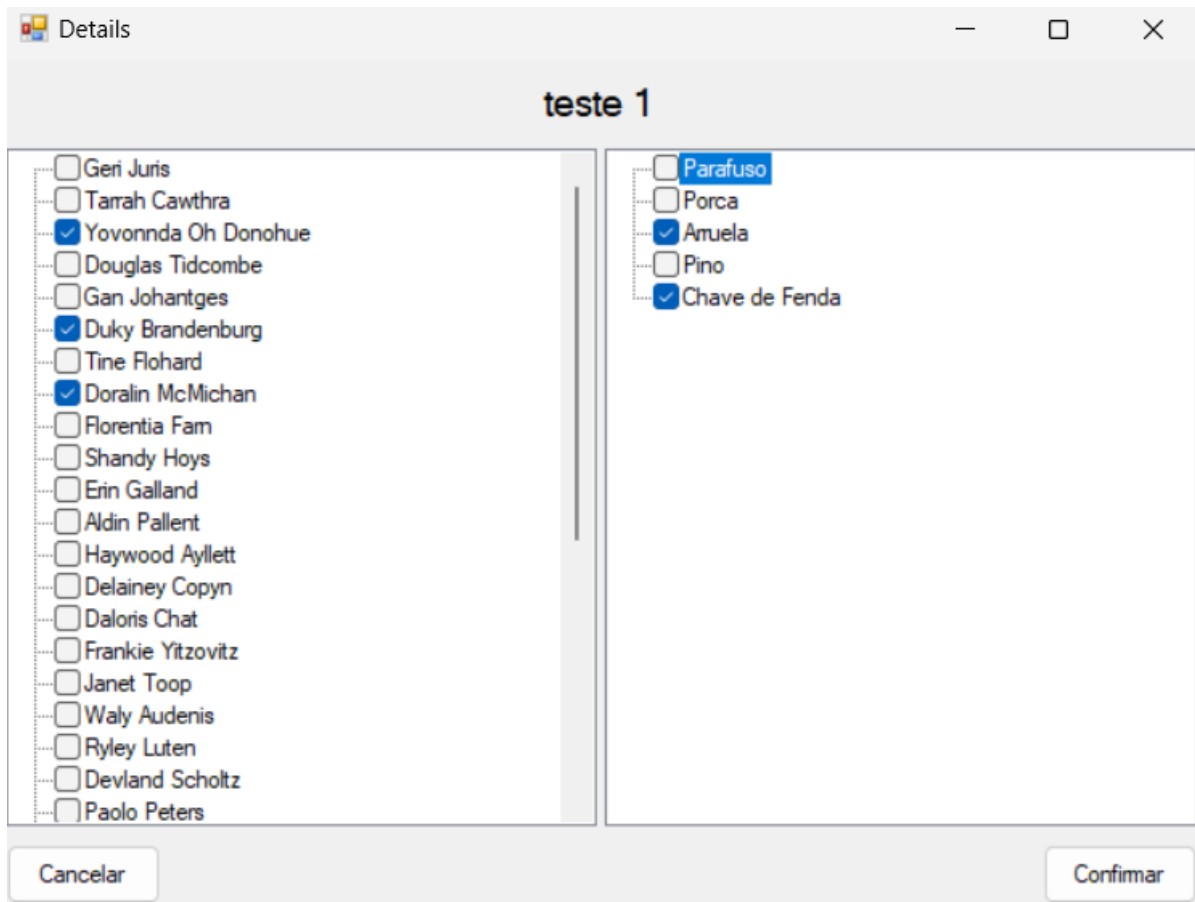
- Nome da Manuntenção:** A text input field containing the word "Teste".
- Selecionar Máquina:** A dropdown menu currently showing "Esmagador 2".
- Selecionar Peças:** A list of checkboxes with the following items:
  - ☐ Parafuso
  - ☐ Porca
  - ☒ Arruela
  - ☐ Pino
  - ☐ Chave de Fenda
- Selecionar Funcionário(s):** A list of checkboxes with the following items:
  - ☐ Geri Juris
  - ☐ Tarrah Cawthra
  - ☒ Yovonnda Oh Donohue
  - ☐ Douglas Tidcombe
  - ☐ Gan Johantges
  - ☐ Duky Brandenburg
  - ☐ Tine Flohard
  - ☒ Doralin McMichan
  - ☐ Florentia Fam
  - ☐ Shandy Hoys

At the bottom of the window, there are two buttons: "Cancelar" on the left and "Confirmar" on the right.



### 5.3 Form de Edição

Neste último *Form* é permitida a edição de uma manutenção, podendo alterar, remover e adicionar peças e funcionários novos.



The screenshot shows a window titled "Details" with a subtitle "teste 1". It contains two columns of checkboxes. The left column lists 20 names, with three checked: Yovonnda Oh Donohue, Duky Brandenburg, and Doralin McMichan. The right column lists four items, with three checked: Parafuso, Aruela, and Chave de Fenda. At the bottom are "Cancelar" and "Confirmar" buttons.

Employee/Part	Selected
Gerri Juris	<input type="checkbox"/>
Tarah Cawthra	<input type="checkbox"/>
Yovonnda Oh Donohue	<input checked="" type="checkbox"/>
Douglas Tidcombe	<input type="checkbox"/>
Gan Johantges	<input type="checkbox"/>
Duky Brandenburg	<input checked="" type="checkbox"/>
Tine Flohard	<input type="checkbox"/>
Doralin McMichan	<input checked="" type="checkbox"/>
Florentia Fam	<input type="checkbox"/>
Shandy Hoys	<input type="checkbox"/>
Erin Galland	<input type="checkbox"/>
Aldin Pallent	<input type="checkbox"/>
Haywood Ayllett	<input type="checkbox"/>
Delainey Copyn	<input type="checkbox"/>
Daloris Chat	<input type="checkbox"/>
Frankie Yitzovitz	<input type="checkbox"/>
Janet Toop	<input type="checkbox"/>
Waly Audenis	<input type="checkbox"/>
Ryley Luten	<input type="checkbox"/>
Devland Scholtz	<input type="checkbox"/>
Paolo Peters	<input type="checkbox"/>
Parafuso	<input checked="" type="checkbox"/>
Porca	<input type="checkbox"/>
Aruela	<input checked="" type="checkbox"/>
Pino	<input type="checkbox"/>
Chave de Fenda	<input checked="" type="checkbox"/>

## Capítulo 6

# Utilização

Começando no form principal, as funções para disponibilizar dados estão disponíveis desde o início.

Adicionando uma nova manutenção, colocamos um nome, selecionamos a máquina, e escolhemos as peças e funcionários associados. Ao confirmar, é inserido a nova manutenção na tabela de manutenção e nas 3 tabelas associadas.

Podem agora ser utilizados os filtros em ambas as tabelas. Clicando com o botão direito na manutenção, podemos mostrar os detalhes, abrindo uma nova interface que permite as funções descritas anteriormente.

Pode também ser marcado como completa uma manutenção, alterando na tabela "Estado\_Manutenção" o estado para "Completo".

Finalmente, pode ser pesquisada uma certa manutenção pelo nome dado anteriormente, e também pode ser apagada uma manutenção. Para apagar uma manutenção, é usada uma *Stored Procedure*, que apaga da tabela manutenção e das 3 tabelas associadas pelo ID.

Não existe login nesta aplicação. A connection string encontra-se no ficheiro "SQL\_Handler.vb", devidamente comentada e identificada. Para ter a base de dados com todas as configurações iniciais, basta executar os seguintes ficheiros, pela ordem aqui indicada:

1. SQL\_DDL
2. SQL\_DML