

## Ficha de Trabalho nº 1

Disciplina	<b>Modelação de Dados em Engenharia</b>
Ano Letivo	2023/2024
Objetivo	Modelar dados de um problema num contexto de Eng. <sup>a</sup> Eletrotécnica
Aulas	4 aulas x 3 horas + 14 horas extra
Estudantes/Grupo	O trabalho deve ser realizado em grupos de <b>3 estudantes</b> . Os elementos do grupo devem pertencer <b>TODOS</b> ao <b>MESMO</b> turno prático.
Data de Entrega	2024/04/15

Objetivos concretos:

1. Modelação de dados
  - a. Compreender e interiorizar os conceitos de modelo e o papel da modelação no contexto de engenharia Eletrotécnica.
  - b. Representação de modelos (linguagens e ferramentas).
  - c. Tipo de modelos com enfase nos tipos conceptuais/lógicos/físicos para a modelação de dados.
  - d. O Modelo de Entidades e Relacionamentos (DER).
  - e. Modelar e implementar a solução para o problema apresentado em baixo.
2. DBMS (Database Management System)
  - a. Passagem dum esquema lógico para um esquema físico
  - b. SQL (para as componentes DDL<sup>1</sup> e DML<sup>2</sup>)
  - c. MySQL
3. [Extra] Valoriza-se (até 1 valor, **com limite máximo de 20 valores**) a ilustração da utilização do modelo num contexto prático de aplicação (JAVA/C++/C# + JDBC<sup>3</sup> ou PDO/PHP, ou outro ambiente onde o aluno se sinta mais confortável).

<sup>1</sup> Data Definition Language

<sup>2</sup> Data Manipulation Language

<sup>3</sup> JAVA Database Connectivity (ODBC: Open Database Connectivity) ou abordagem “Web Application”

## Descrição do trabalho a efetuar

Este trabalho destina-se a proporcionar aos alunos um contacto com os conceitos básicos acerca do projeto/modelação de bases de dados, tendo em vista a sua posterior utilização em atividades enquadradas no seio da Engenharia Eletrotécnica e de Computadores.

Concretamente, o estudo e a implementação do problema proposto neste trabalho servirão para fazer uma abordagem aos seguintes conceitos, que constituem as fases de um projeto de criação de uma base de dados:

- Estudo do problema apresentado.
- Modelação dum diagrama de Entidades e Relacionamentos (*DER*) que satisfaça os requisitos do problema apresentado.
- Transformação do diagrama num esquema/modelo relacional e implementação numa base-de-dados, utilizando *SQL*.
- Implementação de questões (*SQL queries*), vistas (*SQL views*), triggers (*SQL triggers*) e funções/procedimentos (*SQL functions/procedures*) de forma a implementar as funcionalidades pretendidas no problema.
- Valorizam-se (até um máximo de 1 valor, **com limite máximo de 20 valores**) os trabalhos que implementem um conjunto de interfaces amigáveis para a visualização e manipulação dos dados armazenados na BD (*JDBC, outros*).



## Descrição geral do problema (a parte a azul é a mais importante)



Você foi contratado para desenvolver um sistema de base de dados para uma empresa especializada em soluções de domótica para residências e edifícios inteligentes. Este sistema deve gerenciar informações sobre clientes, instalações, contratos, dispositivos inteligentes e a interação entre esses dispositivos, considerando os seguintes pontos:

- **Cientes:** A empresa atende uma ampla gama de clientes, incluindo indivíduos e empresas. Cada cliente é identificado por informações como nome, endereço principal, NIF (Número de Identificação Fiscal), telefone, entre outros.
- **Instalações:** Cada cliente pode ter várias instalações equipadas com sistemas de domótica, identificadas por um código único e um endereço. As instalações podem variar de apartamentos e casas a escritórios e lojas, todos equipados com tecnologia inteligente para automação.

- **Contratos:** Cada instalação é coberta por um contrato que especifica a data de início, a duração, os serviços de domótica contratados, os custos associados, etc.
- **Pacotes/Serviços:** Os serviços são diferenciados por níveis - *lowcost, normal* e *professional*, adaptados às necessidades de automação e controle inteligente:
  - **Lowcost:** Inclui soluções básicas de automação, como controle de iluminação e termostatos inteligentes (até 2 tipos de dispositivos).
  - **Normal:** Expande as opções para incluir controle de iluminação, termostatos, cortinas automáticas e sistemas de som ambiente (até 4 tipos de dispositivos).
  - **Professional:** Oferece uma solução completa de domótica, incluindo todos os dispositivos do pacote standard mais sistemas avançados de segurança, assistentes virtuais e sistemas de irrigação inteligente (sem limite de dispositivos).
- **Dispositivos Inteligentes:** Cada dispositivo é caracterizado por referência do fabricante, modelo, tipo, etc. São registados com data/hora de instalação, estado (ativo/inativo), e podem incluir especificações como consumo de energia, eficiência, entre outros.
- **Interação entre Dispositivos:** O sistema deve ser capaz de gerir cenários de automação, onde a ativação de um dispositivo pode desencadear ações em outros dispositivos (por exemplo, o desligamento de luzes pode ajustar automaticamente o termostato).
- **Simulação de Dados:** Devido à ausência de um sistema real operacional, os dados referentes ao estado e ao comportamento dos dispositivos devem ser simulados para testar e demonstrar as funcionalidades do sistema.
- **Dados Reais:** Posteriormente aos testes com dados simulados, será disponibilizada uma instalação de testes, com alguns dispositivos (medição de consumo energético, ligar/desligar AC, ligar/desligar luz, medição de temperatura e qualidade do ar, etc.). Deverá realizar uma instalação que utilize os dados reais disponibilizados e que atue no mesmo.
- **Gestão de Energia:** Inclui funcionalidades para monitorar e otimizar o consumo de energia, integrando dispositivos inteligentes para maximizar a eficiência energética.
- **Faturação:** Deve haver um registo de faturas emitidas para cada contrato, detalhando número da fatura, data, pacote de serviços contratados e estado da fatura (emitida/paga). As faturas são geradas automaticamente no dia 2 de cada mês.

Este sistema de base de dados deve fornecer uma plataforma robusta para a gestão de soluções de domótica em edifícios inteligentes, facilitando a vida dos usuários através da automação residencial e do controlo inteligente de dispositivos, enquanto promove a eficiência energética e o conforto ambiental.



### Instalação de Testes

A instalação de testes que será disponibilizada para gerar dados reais e possibilitar que seja possível atuar no “mundo real” é composto por alguns dispositivos, como os mostrados nas Figuras.

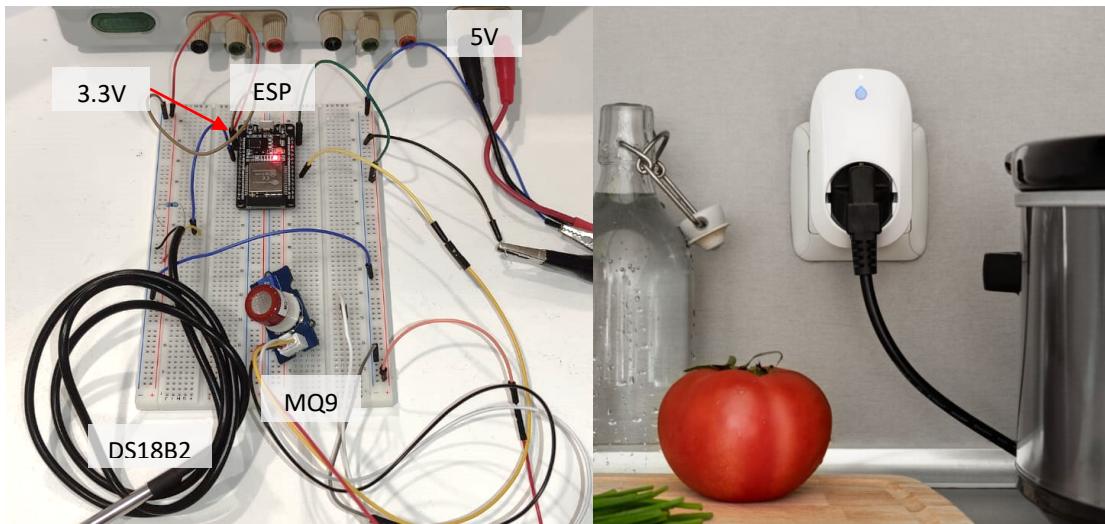


Figura 1 - Sensor de temperatura e qualidade do ar (à esquerda)  
 e sensor de consumo energético e interruptor (à direita)

A extração de dados e o acionamento dos dispositivos é possível através de um *gateway*, que recebe os dados dos dispositivos e envia para outros softwares (neste caso a nossa solução) e recebe os pedidos para acionar algo (por exemplo ligar e desligar um interruptor). Assim, a solução que é proposta deverá ter a seguinte arquitetura.

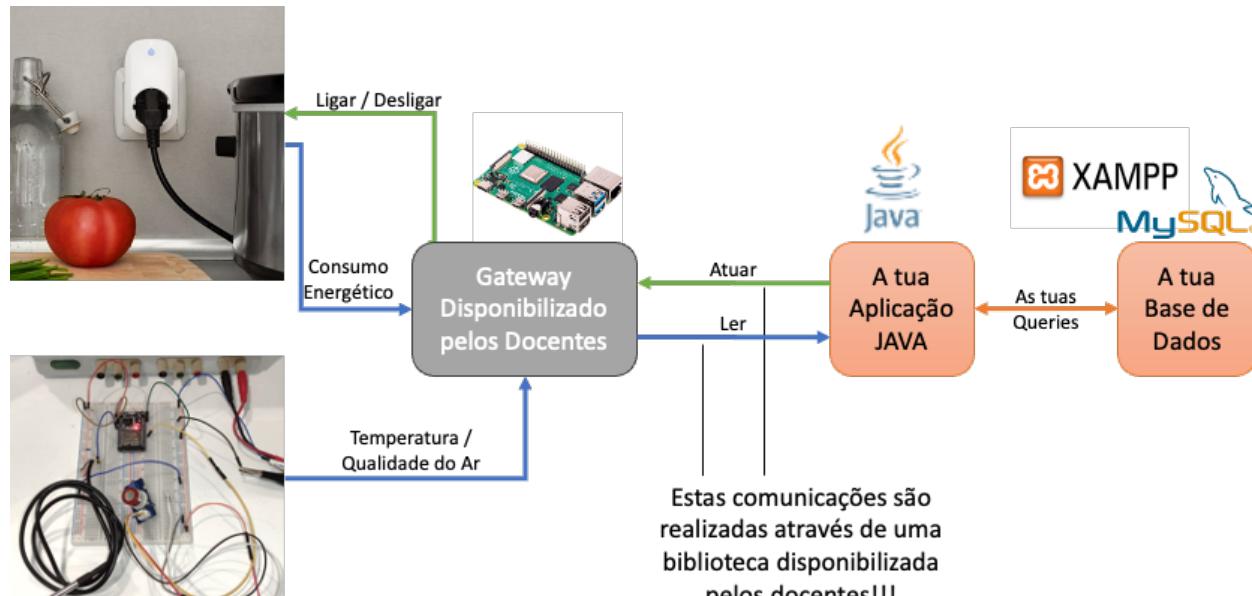


Figura 2 - Arquitetura da instalação de testes

**Criação do diagrama de entidades e relacionamentos (DER) e modelo relacional (a parte a azul é a mais importante)**

O DER e modelo relacional devem ser criados conforme descrito nos slides das aulas teórico-práticas.

Para um trabalho com mais qualidade, recomendamos fazer-se uma especificação do diagrama o mais abrangente possível em termos de entidades e relacionamentos respetivos. Para tal, complemente e complete a descrição informal feita acima (descrição geral do problema). Depois, com base nos slides das aulas TP, relativos ao processo de modelação proposto por Peter Chan, proceder à identificação das entidades e dos relacionamentos. Alguns aspectos relevantes para a fase de modelação:

- O modelo deve estar correto, nomeadamente, reduzindo a possibilidade de ocorrência de informação redundante e a ocorrência de inconsistência nos dados.
- Ter especificamente atenção ao seguinte:
  - Numa fase inicial, ainda não se possui massa crítica para se conseguir avaliar o modelo obtido, nomeadamente, se está correto e completo.
  - Frequentemente, tem ocorrido que alguns alunos ficam durante as semanas em que o trabalho decorre inseguros sobre o modelo obtido, só conseguindo avançar para a fase de implementação na última semana do trabalho.
  - Para se evitar esta situação, recomenda-se um processo mais iterativo.
  - Ou seja, conseguir um modelo inicial que satisfaça o problema apresentado, ainda que não se tenha a certeza.
  - Avançar com o modelo para a geração da BD e tentar satisfazer os requisitos funcionais.
  - Se durante esta fase, for identificada alguma lacuna no modelo, então exportar os dados já introduzidos (para não se perderem), refinar o modelo, e prosseguir deste ponto.
  - Pode ocorrer que o ficheiro com os dados exportados necessite de alguns ajustamentos, se tiverem ocorrido alterações mais radicais no modelo.
  - Ao nível industrial (uma BD já em modo de produção), este processo iterativo de refinação está sempre a ocorrer. No entanto, as modificações às bases-de-dados são feitas de forma muito cuidada e cirúrgica, para garantir-se que não se perde ou compromete a informação armazenada.

**No final, cada elemento do diagrama de entidades e relacionamentos especificado deverá ser justificável através do documento (em \*.docx) semelhante ao anexo1.**

### Requisitos funcionais para o problema apresentado

O desenvolvimento do projeto para o cenário apresentado requer a construção dum modelo para a base-de-dados, que consiga satisfazer um conjunto considerável de requisitos. No entanto, dado o tempo disponível para este trabalho, é apenas necessário que o modelo consiga satisfazer os requisitos funcionais definidos na tabela 1.

Tabela 1 – Requisitos funcionais (para a BD)

Req. Func.	Descrição
RF 1	Implementar as operações CRUD <sup>4</sup> da BD em termos das entidades identificadas para a base-de-dados obtida
RF 2	Proceder ao registo dos valores provenientes dos dispositivos/equipamentos.
RF 3	Proceder ao registo de automação (por exemplo modo noite, modo dia.), sempre que a condição para essa automação se iniciar se verifiquem (por exemplo luminosidade inferior a X, temperatura superior a Y, etc.).
RF 4	Visualizar os dados de cliente cujas instalações sejam de uma determinada tipologia.
RF 5	Visualizar os clientes que tenham contratado um determinado pacote/serviço.
RF 6	Visualizar todos os dispositivos instalados numa dada instalação dentro de um intervalo de tempo.
RF 7	Visualizar o valor médio de faturação (fatura paga) de um cliente num intervalo de tempo.
RF 8	Visualizar as instalações com automações, dentro de um intervalo de tempo.
RF 9	<i>Proponha um requisito relevante e ainda por identificar que implique a especificação de uma ou mais entidades. Implemente.</i>
RF 10	<i>Proponha um requisito relevante ainda por identificar e que requeira uma query simples para o satisfazer (Query ou View usada numa Query). Implemente.</i>
RF 11	<i>Proponha um requisito relevante ainda por identificar e que requeira uma query com funções de agregação (sum, max, min, avg, etc) para o satisfazer. Implemente.</i>
RF 12	<i>Proponha um requisito relevante ainda por identificar e que requeira o desenvolvimento de functions / procedures para o satisfazer. Implemente.</i>
RF 13	Implementação de uma aplicação em Java que seja cliente da base de dados desenvolvida (Realizar os requisitos anteriores numa aplicação)
RF 14	Extração de dados reais da instalação de testes (Extrair e guardar os dados na base de dados respeitando os requisitos anteriores)

Para que se consiga avaliar adequadamente a qualidade do modelo de dados, é necessário que a BD possua dados em quantidade razoável. Por exemplo, calcular a média dum valor com base numa lista que contenha apenas um (ou poucos valores) não é satisfatório. **É necessário que cada tabela possua uma quantidade suficiente de dados que permita ilustrar os requisitos.**

<sup>4</sup> CRUD - Create, Read, Update, Delete

**Nota:**

Ter em atenção que o projeto admite soluções alternativas, não existindo alguma que seja 100% correta à partida (e ainda não se tem experiência para isso). A melhor abordagem é ir começando com uma solução inicial plausível e experimentar o modelo com dados de testes. Conforme se vai testando cada requisito funcional, se algum não puder ser satisfeito, então reformula-se o modelo. Recomenda-se falar com os docentes sobre a estratégia para alterar o modelo, sem se perder os dados já existentes na BD.

De facto, num ambiente de produção (sistema em funcionamento em contexto real), o modelo (e respetivo esquema BD) é frequentemente alterado (ex: alter table ..., create view, ...) de forma a conseguir acompanhar a evolução do negócio/serviço/...

**Modelação**

No seu papel de analista do sistema, e face às necessidades da própria organização, já tomou a decisão que a melhor ferramenta para albergar os dados será o DBMS *MySQL*. Neste contexto, o seu trabalho consiste em três fases distintas, cada uma delas com um objectivo e um papel específico:

- Analista – Fazer a análise do sistema e construir um modelo abstrato (DER) dos dados
- Administrador de Base de Dados – Implementar o modelo abstrato em SQL, através de ferramentas adequadas, por exemplo o *MySQL Workbench*. Garantir que as bases-de-dados por si geridas estão “sempre de boa saúde”, ou seja, servem o propósito para que foram projetadas e estão sempre corretas tendo em conta o seu esquema relacional. O administrador também tem o papel de gerir as permissões de acesso às tabelas e recursos existentes na base-de-dados.
- Programador – Enquadrar num contexto de aplicação a BD obtida. Neste caso, irá recorrer a linguagem Java, ou outra da sua preferência, e à utilização da conectividade JDBC/outra. O programador/utilizador que utiliza a base-de-dados fica assim sujeito ao modelo e regras de integridade definidas no esquema relacional (Valoriza-se o desenvolvimento da aplicação, no entanto não está contemplada na cotação do trabalho).

Na sua futura vida profissional poderá ser confrontado com as três tarefas em separado (em empresas de grande dimensão) ou as três em conjunto (nas empresas de pequena e média dimensão - mais típicas em Portugal). O trabalho de organização dos dados de uma empresa, bem como a melhoria da troca de informação, é muito importante para o aumento dos níveis de produtividade de qualquer organização.

Como analista do sistema deverá questionar as várias pessoas chave da organização, de modo que a estrutura de informação seja a mais adequada, começando, portanto, pela obtenção de um modelo abstrato que albergue as características importantes do problema considerado. Num cenário em que se pretende desenvolver uma base-de-dados ou um esquema relacional, procede-se à respetiva modelação através das seguintes fases de projeto:

1. Identificação das entidades presentes no problema (nível conceptual).
2. Identificação dos relacionamentos existentes entre essas entidades (nível conceptual).

3. Determinação dos atributos de cada entidade e as correspondentes propriedades desses atributos (nível lógico).
4. Obtenção do esquema relacional (nível físico)

## Implementação

Após obtido um *DER* para o problema proposto, passa-se para a fase de transformação desse diagrama num esquema relacional, que neste caso é implementado num *DBMS*.

Posteriormente, deverá construir as questões (*SQL queries*), vistas (*SQL views*) e funções (*SQL functions/procedures*) de forma a implementar as funcionalidades pretendidas no problema. Nesta fase é necessário que a base de dados obtida possua dados que possam ser utilizados para implementar e testar as *queries/procedures/...*; caso esses dados não existam, cria-se então um bom conjunto de dados, que corresponda o mais possível com a “realidade” do problema.

A fase final consiste de construir a interface com o utilizador, neste caso usando Java ou outra linguagem que considere mais adequada ou preferível, e que o aluno tenha já um bom domínio. Tal como o nome indica, e passando a redundância, a interface serve apenas como interface de visualização de resultados. Ou seja, os requisitos funcionais e não funcionais de utilização da BD deverão ser implementados na camada de base-de-dados. Mais uma vez o desenvolvimento desta componente está fora do objectivo deste trabalho.

Durante a execução deste trabalho utilizar-se-ão as seguintes ferramentas:

- *MySQL workBench / Models* – Permite definir DERs e posterior conversão em esquemas relacionais de forma automática.
- *MySQL workBench / Queries* – Interagir com a BD através de *Queries*.
- *MySQL* – Sistema de Gestão de bases de dados de código-aberto, pertencente à Oracle.
- *JDBC/ODBC/...* – Componente de interface (*driver*) que estabelece a conectividade entre um programa e um DBMS, que neste caso é necessária para a ilustração do problema desenvolvido num contexto de aplicação. Em princípio já vem instalada no sistema operativo. Também pode ser usado o driver PDO com PHP
- Adicionalmente, a máquina deverá ter o JAVA SDK instalado. Em algumas situações em que ocorreram problemas na instalação das ferramentas, estas deveram-se ao facto do sistema operativo corresponder a uma das versões “Windows Home edition”.

## Elementos de Reflexão

Esta parte consiste em considerações mais reflexivas que poderiam servir como “inspiração” para a escrita dum relatório (no entanto, não é necessária entrega de relatório para este trabalho). No final do trabalho, as seguintes questões poderiam ser respondidas:

- O que se entende por modelação.
- Noção de modelo.
- Conceito de abstração (ilustrando com pormenores específicos do problema apresentado)
- A utilidade da modelação no contexto da Engenharia.

- O carácter ambíguo, redundante e incompleto da linguagem natural como formalismo de modelação.
- As características que um formalismo de modelação (ou linguagem) deve possuir para que possa representar modelos de forma adequada.

### **Planeamento das aulas**

Este trabalho segue o seguinte plano:

0<sup>a</sup> aula (em casa):

Instalação do *Software* necessário ao desenvolvimento do trabalho.

1<sup>a</sup> aula (**11.03.2024 – 15.03.2024**):

Interação com a base-de-dados MySQL. “Queries” SQL, exemplo de aplicação.  
Especificação de regras de integridade com SQL.

2<sup>a</sup> aula (**18.03.2024 – 22.03.2024**):

Leitura e análise do trabalho prático.  
Análise dos requisitos e modelação DER.  
Implementação da BD (forward engineering).

3<sup>a</sup> aula (**25.03.2024 – 05.04.2024**):

Implementação das funcionalidades requeridas no trabalho prático, CRUD e *Queries* para consulta de informação. Funções de agregação.

4<sup>a</sup> aula (**02.04.2024 – 12.04.2024**):

SQL/PSM : procedimentos, funções e *Triggers*.

**Entrega (no MOODLE)**

O trabalho é entregue UNICAMENTE via Moodle<sup>5</sup> e consiste em:

- ✓ [OBRIGATÓRIO] Preencher RELATÓRIO do trabalho (google form). Neste relatório serão pedidos (entre outros):
  - Descrição de como se identificaram as entidades e os relacionamentos a partir da descrição do problema (ver ANEXO 1).
  - Submissão do diagrama de entidades e relacionamentos (jpg, pdf, etc)
  - Submissão do modelo mySQL-Workbench respectivo (mwb)
  - Descrições como foram implementadas as queries relativas aos diferentes RFs
  - Submissão de pequeno vídeo (5-8 minutos) mostrando as funcionalidades desenvolvidas.
- ✓ [OBRIGATÓRIO] Submeter ficheiro ZIP/RAR file com tudo o que foi efectuado no desenvolvimento da base de dados:
  - DDL.sql (forward engineering – referente ao DER)
  - DML\_funcional\_requirements.sql (VIEWS e Queries para cada RF)
  - DML\_procedures\_functions\_triggers.sql,
  - Data.sql (com os dados de teste)
  - Outros scripts sql criados durante o desenvolvimento do trabalho (opcional).
- ✓ [OPCIONAL] Submeter projeto do programa Java/C++/Visual Basic/Web, etc., caso tenha sido feito.

É importante que cada grupo consiga assegurar-se que os docentes das práticas vão conseguir visualizar o modelo no “*mySQL workbench*” e executar os ficheiros “DDL.sql”, “DML.sql”, e “data.sql” dentro do “*mySQL workbench*”.

Para efeitos de verificação, aconselha-se que cada aluno/grupo construa um “zip/rar” do modelo e o extraia para outra directória, verificando se se consegue abrir o modelo.

---

<sup>5</sup> Entregas feitas através de email NÃO são consideradas.

## Avaliação do Trabalho

É sujeito a apresentação oral, onde têm que estar presentes todos os elementos do grupo (alunos ausentes ficam com **0 valores** na avaliação da componente prática). Apesar do projeto ser feito em grupo, a nota desta componente será sempre individual.

### Critérios de avaliação

Itens	Descrição	Cotação / penalização
<b>Modelos</b>	Diagrama de Entidades e Relacionamentos	1.5
	Modelo mySQL-Workbench	1.0
<b>Requisitos Funcionais</b>	RF 1      Implementar as operações CRUD <sup>6</sup> da BD em termos das entidades identificadas para a base-de-dados obtida	1.0
	RF 2      Proceder ao registo dos valores provenientes dos dispositivos/equipamentos.	0.5
	RF 3      Proceder ao registo de automação (por exemplo modo noite, modo dia.), sempre que a condição para essa automação se iniciar se verifiquem (por exemplo luminosidade inferior a X, temperatura superior a Y, etc.).	1.5
	RF4      Visualizar os dados de cliente cujas instalações sejam de uma determinada tipologia.	0.5
	RF 5      Visualizar os clientes que tenham contratado um determinado pacote/serviço.	1.0
	RF6      Visualizar todos os dispositivos instalados numa dada instalação dentro de um intervalo de tempo.	1.0
	RF7      Visualizar o valor médio de faturaçāo (fatura paga) de um cliente num intervalo de tempo.	1.5
	RF 8      Visualizar as instalações com automações, dentro de um intervalo de tempo.	1.25
	RF 9      Proponha um requisito relevante e ainda por identificar que implique a especificação de uma ou mais entidades. Implemente.	0.75
	RF 10     Proponha um requisito relevante ainda por identificar e que requeira uma query simples para o satisfazer (Query ou View usada numa Query). Implemente.	1.0
	RF 11     Proponha um requisito relevante ainda por identificar e que requeira uma query com funções de agregação (sum, max, min, avg, etc) para o satisfazer. Implemente.	1.0
	RF 12     Proponha um requisito relevante ainda por identificar e que requeira o desenvolvimento de functions / procedures para o satisfazer. Implemente.	1.0
	RF13     Implementação de uma aplicação em Java que seja cliente da base de dados desenvolvida (Realizar os requisitos anteriores numa aplicação)	1.0
	RF14     Extração de dados reais da instalação de testes (Extrair e guardar os dados na base de dados respeitando os requisitos anteriores)	1.0
<b>Discussão</b>	Prestação do grupo (projeto, desempenho discussão oral, relatório)	1.5
	Prestação individual	2.0
<b>Total</b>		<b>20.0</b>
<b>Vídeo</b>	<b>Entrega de vídeo demonstrando todas as funcionalidades é obrigatória (não entrega corresponde a uma não entrega de todo o projeto)</b>	<b>-20.0</b>

<sup>6</sup> CRUD - Create, Read, Update, Delete

**Docentes**

Teórico-prática:

João Rosas, [jrosas@uninova.pt](mailto:jrosas@uninova.pt)  
Ana Inês Oliveira, [aio@fct.unl.pt](mailto:aio@fct.unl.pt)

Prática:

Filipa Ferrada, [fam@fct.unl.pt](mailto:fam@fct.unl.pt)  
André Rocha, [ad.rocha@fct.unl.pt](mailto:ad.rocha@fct.unl.pt)

## Anexo 1 – Modelação: Criação do Diagrama de Entidades e Relacionamentos

Ilustrando como se identificam entidades a partir da descrição do problema. Para modelos de grandes dimensões, uma boa abordagem consiste em representar parcelas pequenas do modelo, ou seja, um fragmento de cada.

<p>Os componentes gastos na produção de produtos, são adquiridos em diversos fornecedores. Em cada venda, aplica-se um lucro de 50% sobre custo de produção, para obter-se o preço dos produtos que são vendidos aos clientes.</p>	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="padding: 5px;">Cliente</td><td style="padding: 5px;"></td></tr> <tr> <td style="padding: 5px;">Componente</td><td style="padding: 5px;">Fornecedor</td></tr> <tr> <td style="padding: 5px;">Produto</td><td style="padding: 5px;">Venda</td></tr> </table>	Cliente		Componente	Fornecedor	Produto	Venda
Cliente							
Componente	Fornecedor						
Produto	Venda						

### *Identificação dos relacionamentos*

<p>Os componentes gastos na produção de produtos, são adquiridos em diversos fornecedores. Em cada venda, aplica-se um lucro de 50% sobre custo de produção, para obter-se o preço dos produtos que são vendidos aos clientes.</p>	<pre> graph TD     Fornecedor -- "aquisições" --&gt; Componente     Componente -- "Comp_gastos" --&gt; Produto     Produto -- "Produtos_vendidos" --&gt; Venda     Venda -- "Vendas_clientes" --&gt; Cliente   </pre>
--	---

Para uma representação mais compacta, pode-se fundir as duas tabelas anteriores numa só, conseguindo-se identificar em simultâneo as entidades e os relacionamentos, utilizando duas cores distintas, por exemplo, destacar com azul os verbos e a verde os substantivos.