



REPÚBLICA DE ANGOLA
MINISTÉRIO DAS TELECOMUNICAÇÕES, TECNOLOGIAS DE INFORMAÇÃO E
COMUNICAÇÃO SOCIAL
MINISTÉRIO DA EDUCAÇÃO
INSTITUTO DE TELECOMUNICAÇÕES

CHAVE DA PROVA DE TLP/SI_12^a

Questões fechadas:

Sobre os conceitos fundamentais de dados e informação, classifique as afirmações:

- 1- Dados são elementos isolados que, por si só, não têm significado específico;
- 2- Informação resulta do processamento e organização dos dados de forma significativa;
- 3- Um sistema de gestão de ficheiros elimina completamente a redundância de dados;
- 4- A inconsistência de dados ocorre quando existem cópias divergentes dos mesmos dados;
- 5- A informação deve ser útil para o utilizador num determinado contexto;

Opção correta: **V-V-F-V-V**

Sobre os Sistemas de Gestão de Base de Dados (SGBD) e seus objectivos:

- 1- Um SGBD é um software desenhado para gerir grandes coleções de dados;
- 2- O SGBD permite o controlo centralizado dos dados, evitando redundância;
- 3- O isolamento entre dados e programas é uma desvantagem dos SGBDs;
- 4- Um SGBD fornece uma visão abstrata dos dados, escondendo detalhes físicos;
- 5- O SGBD é a única entidade que manipula diretamente a base de dados;

Opção correta: **V-V-F-V-V**

Sobre os Níveis de Abstração e Modelos de Dados:

- 1- O Modelo Físico descreve como os dados são realmente armazenados no sistema;
- 2- O Modelo Lógico (conceptual) descreve quais dados estão armazenados e seus relacionamentos;
- 3- O Modelo de Visão é o nível mais baixo de abstração;
- 4- A independência lógica é a capacidade de alterar o esquema lógico sem reescrever programas;

5- O Modelo Relacional é um exemplo de modelo lógico baseado em registos;

Opção correta: **V-V-F-V-V**

No contexto do Modelo Entidade-Relacionamento (MER) e Modelo Relacional (MR):

1- Uma entidade representa um objeto do mundo real, concreto ou abstrato;

2- Atributos são propriedades que descrevem as entidades;

3- Entidades são representadas graficamente por losangos;

4- O domínio de um atributo é o conjunto de valores que ele pode assumir;

5- Um atributo composto pode ser subdividido em partes menores;

Opção correta: **V-V-F-V-V**

Sobre os tipos de Atributos e Entidades no MER:

1- Atributos simples ou atómicos não são divisíveis;

2- Atributos multi-valor podem ter apenas um valor por entidade;

3- Atributos derivados têm o seu valor calculado a partir de outros atributos;

4- Uma Entidade Fraca possui existência independente e chave primária própria forte;

5- O atributo chave identifica unicamente uma entidade;

Opção correta: **V-F-V-F-V**

Relacionamentos e Cardinalidade:

1- Um relacionamento representa a interação entre entidades;

2- Um relacionamento unário ocorre entre duas entidades distintas;

3- A cardinalidade 1:N significa "Um para Muitos";

4- Num relacionamento N:M, uma entidade A associa-se a várias B, e B a várias A;

5- A cardinalidade mínima 1 indica uma associação obrigatória;

Opção correta: **V-F-V-V-V**

Estrutura do Modelo Relacional:

1- Uma Relação é o termo matemático para uma tabela;

2- As linhas de uma tabela são chamadas de atributos;

3- As colunas de uma tabela são chamadas de tuplas;

4- O cruzamento de uma linha com uma coluna deve resultar num valor atómico;

5- A ordem das tuplas (linhas) na tabela é irrelevante;

Opcão correta: **V-F-F-V-V**

Chaves no Modelo Relacional:

1- Uma Superchave é qualquer conjunto de atributos que identifica unicamente uma tupla;

2- Uma Chave Candidata é uma superchave mínima;

3- A Chave Primária é escolhida dentre as chaves candidatas;

- 4- A Chave Primária pode conter valores nulos (NULL);
- 5- A Chave Estrangeira é um atributo que é chave primária noutra tabela;

Opção correta: **V-V-V-F-V**

Regras de Integridade no Modelo Relacional:

- 1- A Integridade de Domínio garante que os valores respeitem o tipo definido;
- 2- A regra de Integridade de Chave Primária exige Unicidade e Não Nulidade;
- 3- A Integridade Referencial valida as Chaves Estrangeiras entre tabelas;
- 4- O valor NULL é membro de todos os domínios;
- 5- Uma Chave Estrangeira não pode ter valores que não existam na tabela referenciada;

Opção correta: **V-V-V-V-V**

Linguagem SQL (Structured Query Language):

- 1- A SQL teve origem na linguagem SEQUEL da IBM na década de 70;
- 2- DDL é usada para manipular dados (Insert, Update);
- 3- DML é usada para definir estruturas (Create, Alter);
- 4- O comando SELECT faz parte da DML;
- 5- O comando DROP pertence à categoria DDL;

Opção correta: **V-F-F-V-V**

Comandos DDL (Data Definition Language):

- 1- CREATE DATABASE cria uma nova base de dados;
- 2- ALTER TABLE permite adicionar ou remover colunas de uma tabela;
- 3- DROP TABLE apaga a tabela e todos os seus dados permanentemente;
- 4- DESCRIBE exibe a estrutura de uma tabela;
- 5- Ao criar uma tabela, não é necessário definir os tipos de dados dos campos;

Opção correta: **V-V-V-V-F**

Comandos DML – INSERT e UPDATE:

- 1- INSERT introduz novos regtos numa tabela;
- 2- INSERT INTO tabela VALUES (...) exige que os valores sigam a ordem das colunas;
- 3- UPDATE altera valores de regtos já existentes;
- 4- WHERE no UPDATE é opcional, mas se omitida altera todos os regtos;
- 5- UPDATE apaga a tabela se não tiver cuidado;

Opção correta: **V-V-V-V-F**

DELETE e SELECT:

- 1- DELETE elimina regtos de uma tabela;

- 2- DELETE FROM tabela sem WHERE apaga todos os dados mas mantém a estrutura;
- 3- SELECT é usado para consultar dados;
- 4- O * no SELECT serve para selecionar apenas a chave primária;
- 5- DELETE elimina colunas específicas da tabela;

Opção correta: **V-V-V-F-F**

Restrições (Constraints) em SQL:

- 1- PRIMARY KEY define a chave primária e implica NOT NULL;
- 2- AUTO_INCREMENT gera valores automáticos para chaves primárias numéricas;
- 3- FOREIGN KEY cria chaves estrangeiras;
- 4- DEFAULT define um valor padrão;
- 5- NOT NULL permite que o campo fique vazio se o utilizador quiser;

Opção correta: **V-V-V-V-F**

MySQL e Workbench:

- 1- MySQL é um SGBD relacional multi-utilizador;
- 2- O Workbench é uma GUI para gerir o MySQL;
- 3- O utilizador padrão chama-se 'admin';
- 4- Scripts SQL no Workbench podem ser salvos com extensão .sql;
- 5- O Workbench permite modelagem de dados e execução de queries;

Opção correta: **V-V-F-V-V**

Mapeamento do Modelo ER para o Modelo Relacional:

- 1- Entidades tornam-se Tabelas;
- 2- Relacionamentos 1:N migram a chave primária para o lado "Muitos";
- 3- Relacionamentos M:N criam tabela intermédia;
- 4- Atributos multivvalorados são ignorados no relacional;
- 5- Chaves estrangeiras representam relacionamentos;

Opção correta: **V-V-V-F-V**

Backup e Recuperação em MySQL:

- 1- Backup garante integridade e recuperação;
- 2- Dump gera ficheiro SQL;
- 3- mysqldump faz backups;
- 4- Logs binários não servem para recuperação;
- 5- É possível backup só da estrutura;

Opção correta: **V-V-V-F-V**

Tipos de Dados em SQL:

- 1- INT é usado para inteiros;
- 2- VARCHAR(n) armazena texto variável;
- 3- DATE armazena data e hora;
- 4- CHAR(n) armazena texto de tamanho fixo;
- 5- FLOAT armazena números decimais;

Opção correta: **V-V-F-V-V**

Cláusula WHERE em SQL:

- 1- Filtra registos no SELECT/UPDATE/DELETE;
- 2- Permite operadores =, >, <;
- 3- É obrigatória em SELECT;
- 4- Permite combinar AND/OR;
- 5- DELETE sem WHERE apaga tudo;

Opção correta: **V-V-F-V-V**

Visão Geral de Bases de Dados:

- 1- Persistência = dados sobrevivem ao desligar;
- 2- Modelo Hierárquico é o mais usado hoje;
- 3- SQL é padrão para BD relacionais;
- 4- Schema agrupa objetos;
- 5- Modelagem representa o mundo real;

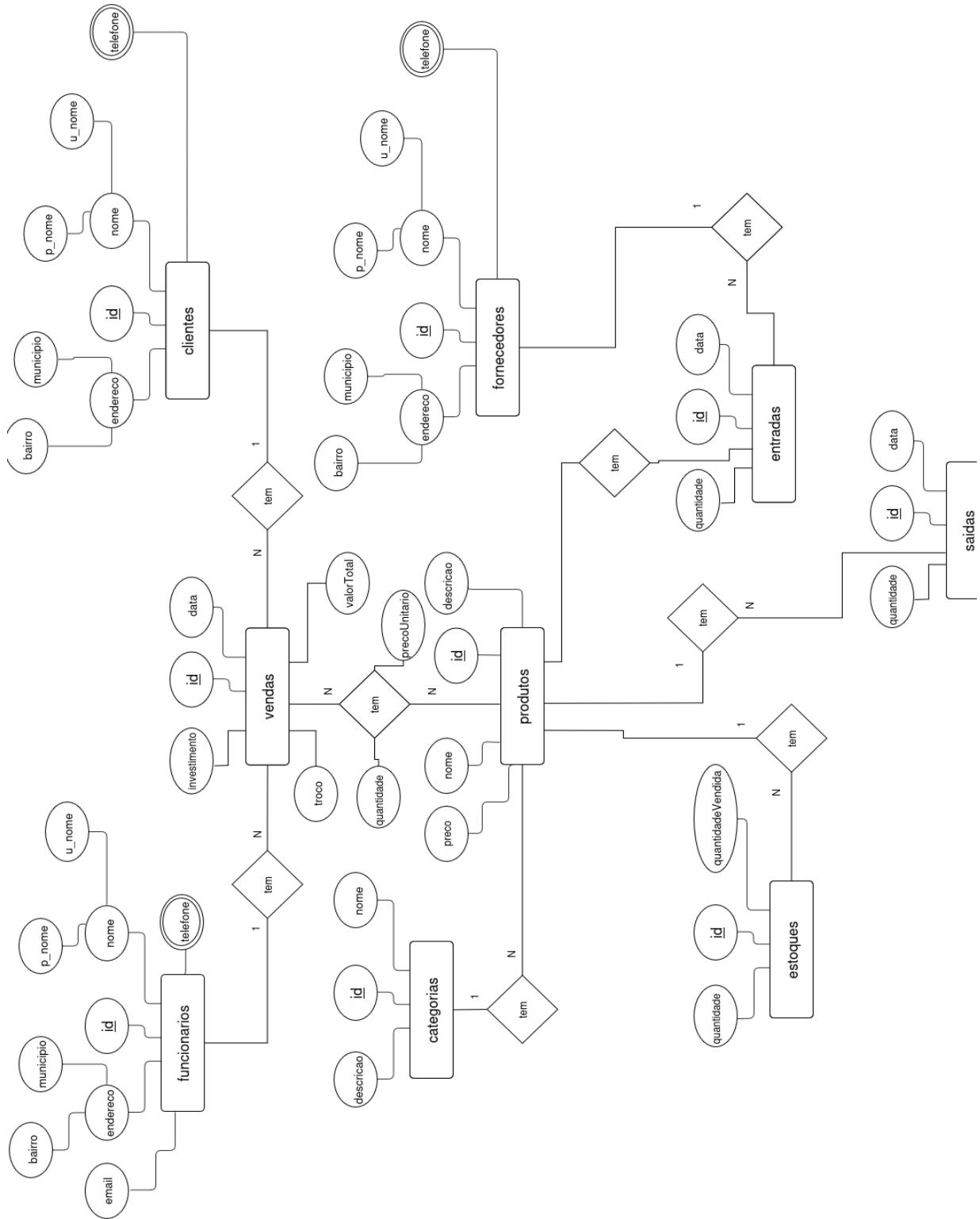
Opção correta: **V-F-V-V-V**

Questões abertas:

a) **MER:**

funcionarios (id, pNome, uNome, email, municipio, bairro);
telefoneFuncionarios (id, idFuncionario, numeroTelefone);
clientes (id, pNome, uNome, municipio, bairro);
telefoneClientes (id, idCliente, numeroTelefone);
fornecedores (id, pNome, uNome, municipio, bairro);
telefoneFornecedores (id, idFornecedor, numeroTelefone);
categorias (id, nome, descricao);
produtos (id, nome, descricao, idCategoria, preco);
estoques (id, idProduto, quantidade, quantidadeVendida);
entradas (id, idProduto, idFornecedor, quantidade, data);
saidas (id, idProduto, quantidade, data);
vendas (id, idFuncionario, idCliente, investimento, data);
vendaProduto (id, idvenda, idProduto, quantidade).

DER:



A “b)” é variável. Constam as seguintes:

Script genérico:

```
CREATE DATABASE ferrete;
```

```
USE ferrete;
```

```
CREATE TABLE funcionarios (
    id INT AUTO_INCREMENT PRIMARY KEY,
    pNome VARCHAR(50) NOT NULL,
    uNome VARCHAR(50) NOT NULL,
    email VARCHAR(100),
    municipio VARCHAR(50),
    bairro VARCHAR(50)
);
```

```
CREATE TABLE telefoneFuncionarios (
    id INT AUTO_INCREMENT PRIMARY KEY,
    idFuncionario INT NOT NULL,
    numeroTelefone VARCHAR(20) NOT NULL,
    FOREIGN KEY (idFuncionario) REFERENCES funcionarios(id)
);
```

```
CREATE TABLE clientes (
    id INT AUTO_INCREMENT PRIMARY KEY,
    pNome VARCHAR(50) NOT NULL,
    uNome VARCHAR(50) NOT NULL,
    municipio VARCHAR(50),
    bairro VARCHAR(50)
);
```

```
CREATE TABLE telefoneClientes (
    id INT AUTO_INCREMENT PRIMARY KEY,
    idCliente INT NOT NULL,
    numeroTelefone VARCHAR(20) NOT NULL,
    FOREIGN KEY (idCliente) REFERENCES clientes(id)
);
```

```
CREATE TABLE fornecedores (
    id INT AUTO_INCREMENT PRIMARY KEY,
    pNome VARCHAR(50) NOT NULL,
    uNome VARCHAR(50) NOT NULL,
    municipio VARCHAR(50),
    bairro VARCHAR(50)
);
```

```
CREATE TABLE telefoneFornecedores (
    id INT AUTO_INCREMENT PRIMARY KEY,
    idFornecedor INT NOT NULL,
    numeroTelefone VARCHAR(20) NOT NULL,
    FOREIGN KEY (idFornecedor) REFERENCES fornecedores(id)
);
```

```
CREATE TABLE categorias (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nome VARCHAR(100) NOT NULL,
    descricao VARCHAR(150)
);
```

```
CREATE TABLE produtos (
    id INT AUTO_INCREMENT PRIMARY KEY,
    nome VARCHAR(100) NOT NULL,
    descricao VARCHAR(150),
    idCategoria INT NOT NULL,
    preco DECIMAL(10,2) NOT NULL,
    FOREIGN KEY (idCategoria) REFERENCES categorias(id)
);
```

```
CREATE TABLE estoques (
    id INT AUTO_INCREMENT PRIMARY KEY,
    idProduto INT NOT NULL,
    quantidade INT NOT NULL,
    quantidadeVendida INT NOT NULL,
    FOREIGN KEY (idProduto) REFERENCES produtos(id)
);
```

```
CREATE TABLE entradas (
    id INT AUTO_INCREMENT PRIMARY KEY,
    idProduto INT NOT NULL,
    idFornecedor INT NOT NULL,
    quantidade INT NOT NULL,
    data DATE NOT NULL,
    FOREIGN KEY (idProduto) REFERENCES produtos(id),
    FOREIGN KEY (idFornecedor) REFERENCES fornecedores(id)
);
```

```
CREATE TABLE saidas (
    id INT AUTO_INCREMENT PRIMARY KEY,
    idProduto INT NOT NULL,
    quantidade INT NOT NULL,
    data DATE NOT NULL,
    FOREIGN KEY (idProduto) REFERENCES produtos(id)
);
```

```
CREATE TABLE vendas (
    id INT AUTO_INCREMENT PRIMARY KEY,
    idFuncionario INT NOT NULL,
    idCliente INT NOT NULL,
    investimento DECIMAL(10,2) NOT NULL,
    data DATE NOT NULL,
    FOREIGN KEY (idFuncionario) REFERENCES funcionarios(id),
    FOREIGN KEY (idCliente) REFERENCES clientes(id)
);
```

```
CREATE TABLE vendaProduto (
    id INT AUTO_INCREMENT PRIMARY KEY,
    idVenda INT NOT NULL,
    idProduto INT NOT NULL,
    quantidade INT NOT NULL,
    FOREIGN KEY (idVenda) REFERENCES vendas(id),
    FOREIGN KEY (idProduto) REFERENCES produtos(id)
);
```

Variável_1_UPDATE:

```
UPDATE produtos SET preco = 950.00 WHERE id = 1;
UPDATE produtos SET preco = 1800.00 WHERE id = 2;
```

```
UPDATE estoques SET quantidade = 300 WHERE idProduto = 1;
```

```
UPDATE clientes SET bairro = 'Zango 5' WHERE id = 1;
UPDATE clientes SET bairro = 'Kilamba Kaxi' WHERE id = 2;
```

```
UPDATE funcionarios SET email = 'novoemail@ferrete.co.ao' WHERE id = 1;
```

Variável_2_SELECT:

```
SELECT * FROM produtos;
```

```
SELECT * FROM funcionarios;
```

```
SELECT * FROM produtos WHERE preco > 1000;
```

Variável_3_INSERT:

```
INSERT INTO funcionarios (pNome, uNome, email, municipio, bairro) VALUES
('Carlos', 'Mendes', 'carlos@ferrete.ao', 'Viana', 'Zango'),
('Ana', 'Silva', 'ana@ferrete.ao', 'Cazenga', 'Hoji-Ya-Henda'),
('Paulo', 'Ramos', 'paulo@ferrete.ao', 'Talatona', 'Benfica'),
('Lidia', 'Costa', 'lidia@ferrete.ao', 'Luanda', 'Maianga');
```

```
INSERT INTO clientes (pNome, uNome, municipio, bairro) VALUES  
('Joao', 'Paulo', 'Viana', 'Estalagem'),  
('Maria', 'Fernandes', 'Kilamba', 'Centralidade'),  
('Luis', 'Pedro', 'Cacuaco', 'Sequele');
```

```
INSERT INTO fornecedores (pNome, uNome, municipio, bairro) VALUES  
('Pharma', 'Angola', 'Luanda', 'Mutamba'),  
('Saude', 'Global', 'Viana', 'Zango');
```

```
INSERT INTO produtos (nome, descricao, preco) VALUES  
('Paracetamol', '500mg', 500.00),  
('Ibuprofeno', '400mg', 800.00),  
('Vitamina C', '1000mg', 1200.00),  
('Vitamina D', 'Suplemento', 1500.00),  
('Amoxicilina', '500mg', 2000.00);
```

```
INSERT INTO estoques (idProduto, quantidade) VALUES  
(1, 200),  
(2, 150),  
(3, 100),  
(4, 80),  
(5, 60);
```

```
INSERT INTO entradas (idProduto, quantidade, data) VALUES  
(1, 50, '2025-03-01'),  
(2, 40, '2025-03-01'),  
(3, 30, '2025-03-01');
```

```
INSERT INTO vendas (data) VALUES ('2025-03-02');
```

```
INSERT INTO vendaProduto (idVenda, idProduto, quantidade) VALUES  
(1, 1, 2),  
(1, 3, 1);
```

Variável_4_ALTER:

```
ALTER TABLE clientes CHANGE pNome primeiroNome VARCHAR(50);  
ALTER TABLE clientes CHANGE uNome ultimoNome VARCHAR(50);  
ALTER TABLE clientes CHANGE municipio cidade VARCHAR(50);
```

```
ALTER TABLE produtos DROP descricao;  
ALTER TABLE produtos DROP preco;
```

```
ALTER TABLE funcionarios MODIFY email VARCHAR(150);
```

