

e-Fólio B

U.C. 21053

Fundamentos de Bases de Dados

2017-2018

Resolução e Critérios de Correção

INSTRUÇÕES:

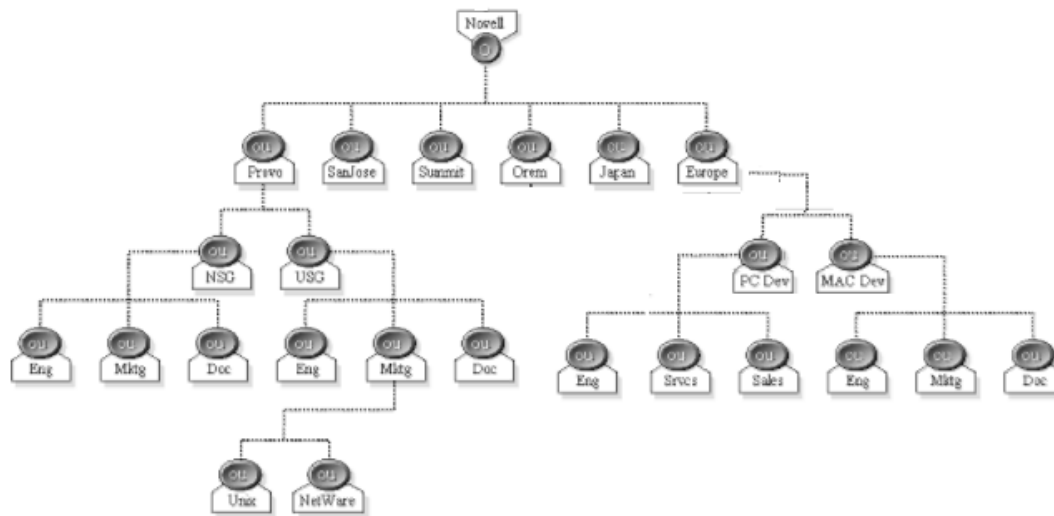
- 1) O e-fólio é constituído por 5 perguntas. A cotação global é de 5 valores.
- 2) O e-fólio deve ser entregue num único ficheiro PDF, não zipado, com fundo branco, com perguntas numeradas e sem necessidade de rodar o texto para o ler. Penalização de 10% a 100%.
- 3) Não são aceites e-fólios manuscritos, i.e. tem penalização de 100%.
- 4) O nome do ficheiro deve seguir a normal “eFolioB” + <nº estudante> + <nome estudante com o máximo de 3 palavras>
- 5) Na primeira página do e-fólio deve constar o nome completo do estudante bem como o seu número. Penalização de 10% a 100%.
- 6) Durante a realização do e-fólio, os estudantes devem concentrar-se na resolução do seu trabalho individual, não sendo permitida a colocação de perguntas ao professor ou entre colegas.
- 7) A interpretação das perguntas também faz parte da sua resolução, se encontrar alguma ambiguidade deve indicar claramente como foi resolvida.
- 8) A legibilidade, a objectividade e a clareza nas respostas serão valorizadas, pelo que, a falta destas qualidades serão penalizadas.

A informação da avaliação do estudante está contida no **vetor das cotações**:

Questão: 1 2 3 4 5

Cotação: 10 10 10, 5 15 décimas

1) Considere a seguinte figura relativa com a estrutura hierárquica com 5 níveis de uma empresa:



- a) (1/2 valor) crie uma base de dados em SQL que suporte a estrutura hierárquica
 b) (1/2 valor) crie uma consulta em SQL que devolva o número de níveis na empresa

Resposta:

a) A tabela que contém um ID que é a chave principal, um Nome e uma chave estrangeira IDPai que aponta (recursivamente) para a linha pai que pode ser NULL no caso de ser raiz.

```
CREATE TABLE Relacao
```

```
(
  ID INT,
  Nome VARCHAR(20),
  IDPai INT,
  PRIMARY KEY (ID),
  FOREIGN KEY (IDPai) REFERENCES Relacao(ID)
);
```

b) Neste caso usamos uma consulta recursiva

```
WITH RelacaoTmp AS (
  SELECT ID, IDPai, 1 AS Nivel -- nível da raiz
  FROM Relacao AS R1
  WHERE IDPai IS NULL
  UNION ALL
  SELECT ID, IDPai, Nivel+1 -- recursividade restantes níveis
  FROM Relacao AS R2
  JOIN RelacaoTmp ON RelacaoTmp.ID=R2.IDPai)
SELECT MAX (Nivel) AS NumeroNiveis
FROM RelacaoTmp
```

Critérios de correção:

- a) 5 dec. criar tabela com ID (chave principal) e IDPai (chave estrangeira)
- b) 5 dec. criar consulta recursiva com início da recursividade na raiz
- soluções com numerações prévias: - 50%
- erros, omissões ou redundâncias: -20% a -100%

2) Considere 3 utilizadores B1, B2 e B3 e suponha que o utilizador B1 cria a tabela FACTURA numa base de dados, de seguida diga:

2.a) (1/3 valor) Como B1 pode transmitir o privilégio de acesso a FACTURA a B2 e permite-lhe que propague esse privilégio a outros.

Resposta:

A transferência de privilégios é realizada por um utilizador com o comando GRANT, com a sintaxe:

```
GRANT <lista privilégios: ALL, SELECT, UPDATE INSERT, DELETE>  
ON <tabela, view>  
TO <utilizador ou lista>
```

```
GRANT SELECT  
ON FATURA  
TO B2  
WITH GRANT OPTION;
```

2.b) (1/3 valor) Como B2 transmite o privilégio de acesso a FACTURA a B3.

Resposta:

```
GRANT SELECT  
ON FATURA  
TO B3;
```

2.c) (1/3 valor) Como B1 cancela todos os privilégios de acesso a FACTURA.

Resposta:

A remoção de privilégios é realizada por um utilizador com o comando REVOKE, com a sintaxe:

```
REVOKE <lista privilégios: ALL, SELECT, UPDATE INSERT, DELETE>  
ON <tabela, view>  
FROM <utilizador ou lista>
```

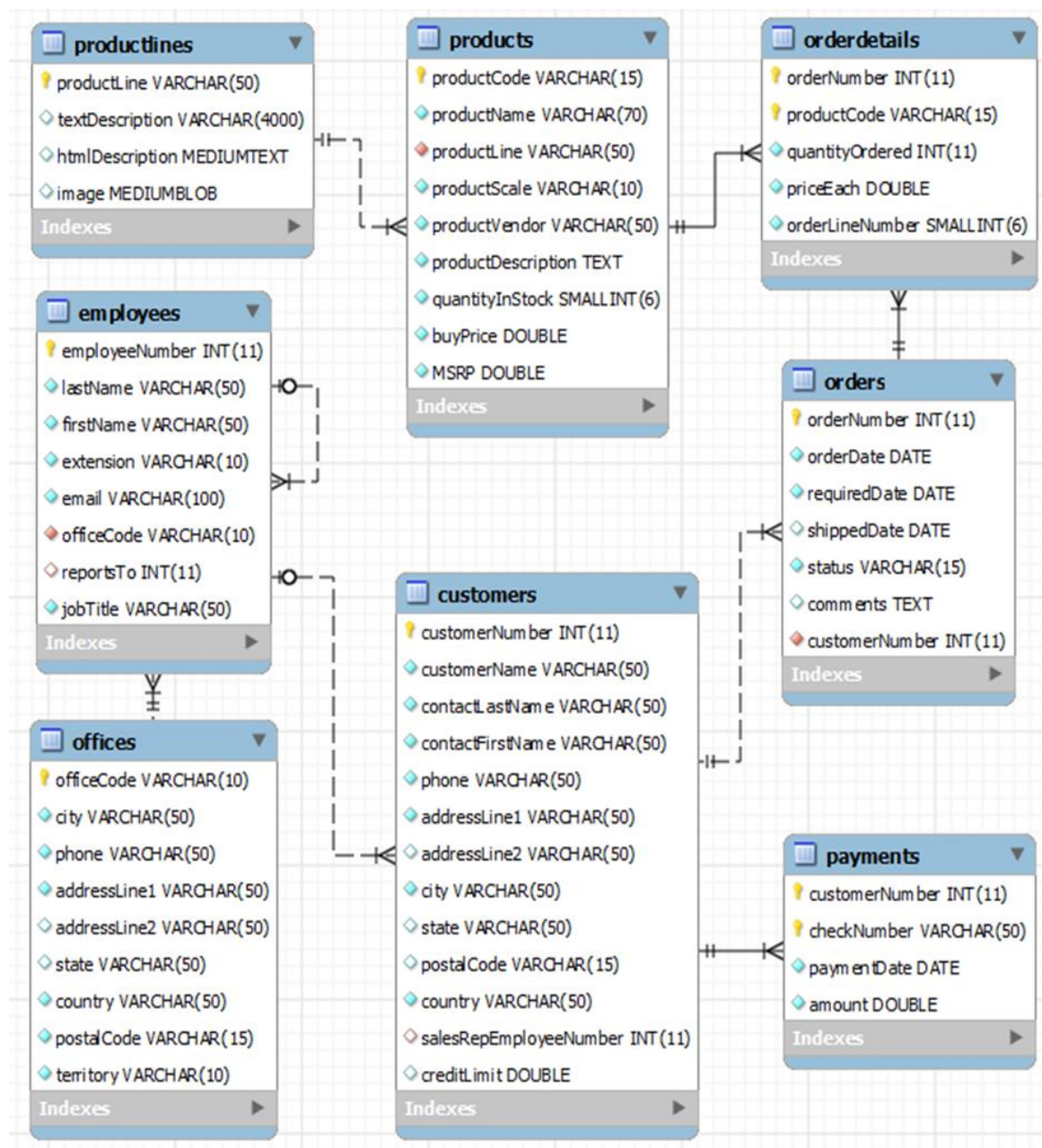
```
REVOKE SELECT  
ON FATURA  
FROM B2 CASCADE;
```

Critérios de correção:

- falta CASCADE – 1 décima

- erros, omissões ou redundâncias: -20% a -100%

3) Considere a seguinte base de dados e escreva em Álgebra Relacional, considerando os operadores de σ , Π , \bowtie e G para as funções agregadoras de G_{sum} , G_{count} , $G_{average}$, etc.



a) (1/2 valor) Qual o valor monetário das encomendas por estado?

b) (1/2 valor) Qual o número do cliente com a maior soma de pagamentos no ano 2017?

Respostas

a) (1/2 valor) Qual o valor monetário das encomendas por estado?

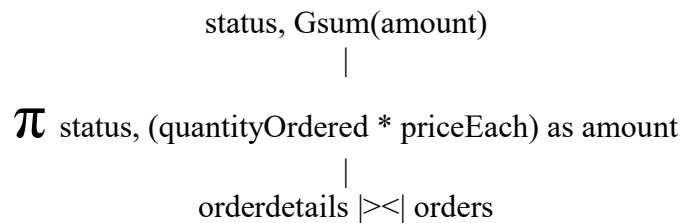
Resposta:

```
SELECT
  status,
  TRUNCATE(SUM(quantityOrdered * priceEach),2) AS amount
FROM orders o, orderdetails od
WHERE o.orderNumber= od.orderNumber
GROUP BY status;
```

status	amount
Cancelled	238854.18
Disputed	61158.77
In Process	135271.51
On Hold	169575.61
Resolved	134235.87
Shipped	8865094.64

temp $\leftarrow \pi$ status, quantityOrdered*priceEach (orderdetails \bowtie orders) as amount
 resultado \leftarrow status, Gsum (amount) [temp]

Podemos ainda desenhar a árvore do plano de execução:



Crítérios de correção:

- 1/3 cotação: utilizar as tabelas orders e orderdetails
- 1/3 cotação: realizar o produto de quantidade pelo preço
- 1/3 cotação: calcular a soma por 'status'
- dificuldade na leitura: - 1 a -4 décimas
- erros, omissões ou redundâncias: -20% a -100%

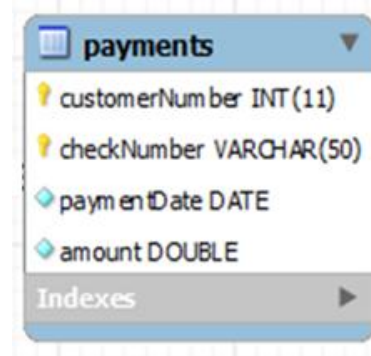
b) (1/2 valor) Qual o número do cliente com a maior soma de pagamentos no ano 2017?

Resposta: o número do cliente corresponde a *payments.customerNumber*

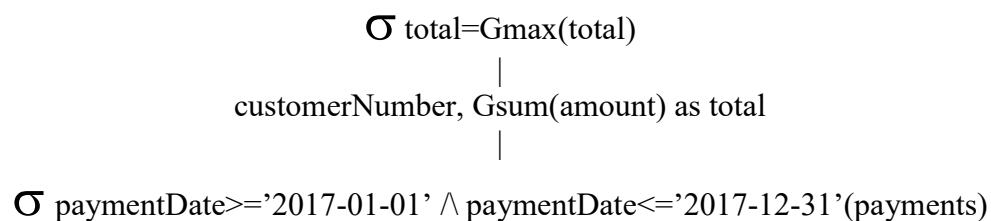
Em SQL teremos:

```
CREATE VIEW Consulta1
SELECT customerNumber, SUM (amount) as total
FROM payments
WHERE paymentDate>='2017-01-01'
AND paymentDate<='2017-12-31'
GROUP BY customerNumber) alias T
```

```
CREATE VIEW Consulta2
SELECT T.customerNumber
FROM Consulta1 T
WHERE T.total = (SELECT MAX(T.total)
FROM T)
```



Podemos ainda desenhar a árvore do plano de execução:



E finalmente em álgebra relacional:

temp1 $\leftarrow \sigma_{paymentDate>='2017-01-01' \wedge paymentDate<='2017-12-31'}[payments]$

temp2 $\leftarrow customerNumber, Gsum(amount) as total [temp1]$

resultado $\leftarrow \sigma_{total= Gmax(total)} [temp2]$

Crítérios de correção:

- 1/3 cotação: utilizar a tabela payments
- 1/3 cotação: realizar a soma amount
- 1/3 cotação: encontrar o cliente cujo total é igual ao máximo do total
- dificuldade na leitura: - 1 a -4 décimas
- erros, omissões ou redundâncias: -20% a -100%

4) (1/2 valor) Implemente em SQL o operador FACTOR para o exemplo *roster* (*passenger, class, smoking*) pp. 38 da bibliografia de David Maier do Capítulo 3.

Resposta:

<i>roster</i> (PASSENGER	CLASS	SMOKING)
Salazar	first	yes
Schick	first	no
Shockley	coach	no
Stewart	first	yes
Sayers	coach	no
Sands	coach	no
Sachs	coach	yes

There are only four possible {CLASS, SMOKING}-values. We can represent the same information in less space by splitting off the CLASS and SMOKING columns, and creating a new column, LINK, as shown below.

<i>roster1</i> (PASSENGER	LINK)	<i>roster2</i> (LINK	CLASS	SMOKING)
Salazar	1	1	first	yes
Schick	2	2	first	no
Shockley	4	3	coach	yes
Stewart	1	4	coach	no
Sayers	4			
Sands	4			
Sachs	3			

-- Criação da tabela roster2

```
CREATE TABLE roster2 AS (SELECT DISTINCT CLASS, SMOKING
                          FROM roster
                          ORDER BY CLASS DESC, SMOKING DESC);
```

-- Inserção da coluna LINK na primeira posição

```
ALTER TABLE roster2
ADD LINK INT NOT NULL AUTO_INCREMENT PRIMARY KEY FIRST;
```

-- Criação da tabela roster1

```
CREATE TABLE roster1 AS (SELECT roster.PASSENGER, roster2.LINK
                          FROM roster, roster2
                          WHERE roster.CLASS=roster2.CLASS
                          AND roster.SMOKING=roster2.SMOKING);
```

Critérios de correção:

- 1/2 cotação: roster2

- 1/2 cotação: roster1

- erros, omissões ou redundâncias: -20% a -100%

5) Projeto de Bases de Dados

Numa empresa multinacional, dispersa por vários continentes, vários países e várias cidades existem escritórios com empregados. Os dados dos empregados incluem o nome, contactos (e-mails e telemóvel), função atual, salário atual e escritório atual. Cada empregado tem um histórico. Os empregados podem mudar de escritório, de função e de salário durante a sua vida na empresa. Podem ainda sair e voltar a entrar na empresa passado algum tempo. Para cada função, existe um título e um salário máximo e mínimo. Pretende-se manter a informação atual e histórica de todos os empregados da empresa multinacional.

5.a) (1/2 valor) Comece por identificar as diferentes entidades do Modelo Entidade-Relação. De seguida, identifique os relacionamentos entre as entidades do Modelo Entidade-Relação. Dê um nome ao relacionamento. Classifique cada relacionamento quanto à cardinalidade (1:1, 1:N, N:N). De seguida, desenhe o diagrama do Modelo Entidade-Relação. Identifique os relacionamentos e os atributos das entidades.

Resposta:

Entidades:

escritorio (id_escritorio, nome_escritório, continente, país, cidade)

empregado (id_empregado, nome, email, telemovel)

historico (id, id_empregado, id_escritorio, id_funcao, salario, data_inicio, data_fim)

funcao (id_funcao, título, salário_máximo, salário_mínimo)

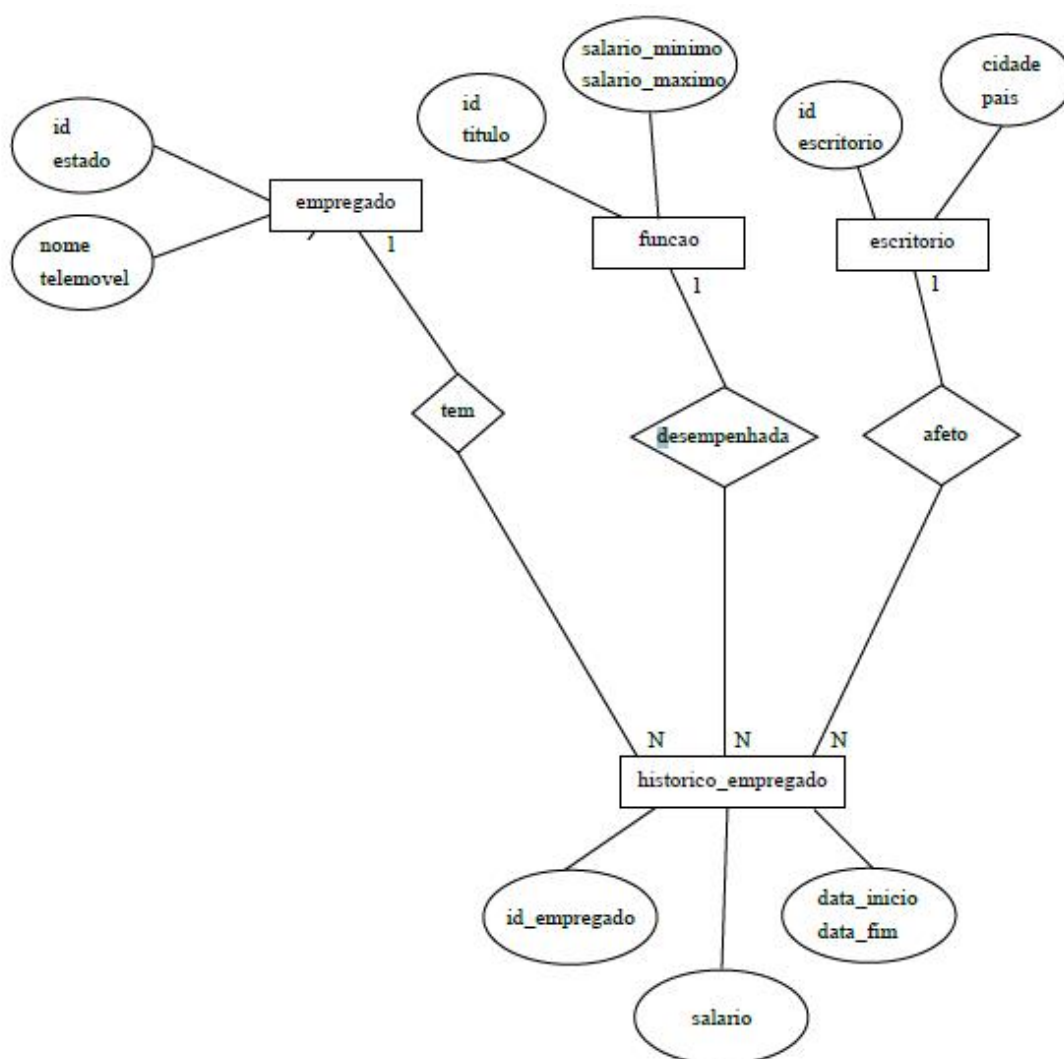
Relações:

escritorio-historico, 1:N, um escritório tem um ou mais empregados ativos

função-historico, 1:N, uma função tem ou mais empregados ativos

empregado-historico, 1:N, cada empregado tem um ou mais registos no histórico

Nota: empregados ativo i.e. com data_fim=NULL



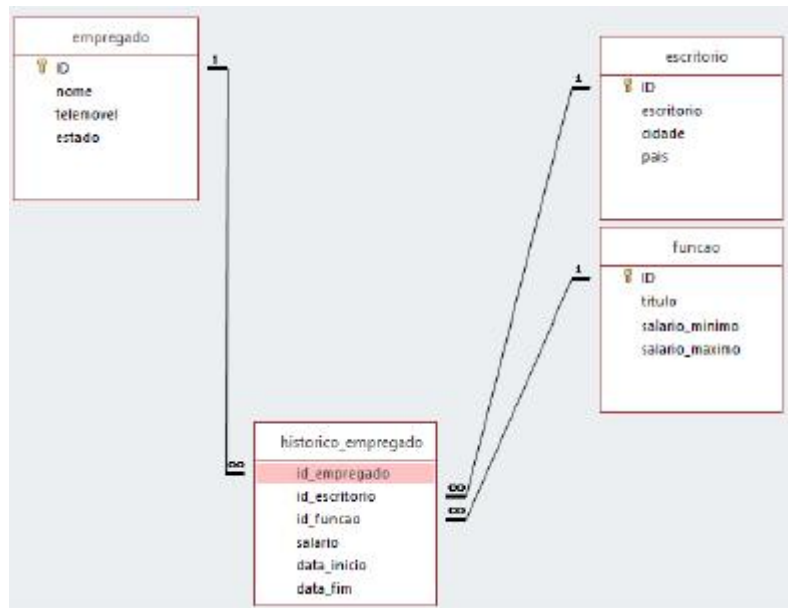
Crítérios de correção:

- 1/2 cotação: entidades e relações (cardinalidade)

- 1/2 cotação: modelos E/R

- erros, omissões ou redundâncias: -20% a -100%

5.b) (1/2 valor) Desenhe a base de dados relacional correspondente ao modelo anterior, em que nas ligações de 1:N, a tabela com uma única linha é desenhada em cima e da tabela com várias linhas é desenhada por baixo. A base de dados não deve exceder as 7 tabelas. Se utilizar ligações que denotem transitividade, deve justificá-la convenientemente.



Critérios de correção:

- 1/2 cotação: tabelas
- 1/2 cotação: cardinalidade
- valoriza-se a simplicidade, a representação e a não-transitividade
- erros, omissões ou redundâncias: -20% a -100%

5.c) (1/2 valor) Identifique dois fatores críticos de sucesso e escreva as respectivas consultas em SQL.

FCS_1 - Consultar o total de salários dos empregados, por escritório para os empregados ativos i.e. com data_fim=NULL

```
SELECT id_escritorio, SUM(empregado.salario)
FROM historico
WHERE data_fim =NULL
GROUP BY id_escritorio
```

FCS_2 - Consultar o histórico de um empregado com base no nome:

```
SELECT *
FROM historico
WHERE empregado.nome= X
```

Crítérios de correção:

- 1/2 cotação: FCS1
- 1/2 cotação: FCS2
- erros, omissões ou redundâncias: -20% a -100%