|  |  |
| --- | --- |
|  | Instituto Politécnico de Leiria  Escola Superior de Tecnologia e Gestão  **Departamento de Engenharia Informática** |
| Curso Técnico Superior Profissional de  Programação de Sistemas de Informação  Ano Letivo 2022/23  **Conceção e Desenvolvimento de Bases de Dados**  1.º ano | 2.º Semestre |

**Projeto de Base de Dados**

*Gestão de Visitas a Crianças/Jovens Institucionalizados*

Versão 1.0 - 9 de Maio de 2023

Autores:

Estudante David Domingues, n.º 2220897

Estudante Hugo Gomes, n.º 2220893

Estudante Ruben Soares, n.º 2220900

# Índice

[Índice 2](#_Toc138271069)

[1. Introdução 3](#_Toc138271070)

[2. Descrição do Sistema 4](#_Toc138271071)

[3. Modelo Conceptual 5](#_Toc138271072)

[3.2 Diagrama de Entidade-Relacionamento 8](#_Toc138271073)

[4. Modelo Lógico 10](#_Toc138271074)

[5. Permissões de acesso 12](#_Toc138271075)

[6. Consultas à base de dados 13](#_Toc138271076)

[6.1 – Consulta de visitas de cada visitante com o respetivo utente, ordenado por data mais antiga. 13](#_Toc138271077)

[6.2 –Consultar os dados da tabela Pessoa com o respetivo perfil 13](#_Toc138271078)

[6.3 – Qual o tipo de acolhimento de cada utente? 14](#_Toc138271079)

[6.4 –Quantidade de visitas por faixa etária dos utentes. 14](#_Toc138271080)

[6.5 - Quais funcionários são também visitantes 14](#_Toc138271081)

[6.6 - Quantas salas existem para realização de visitas na instituição? 15](#_Toc138271082)

[6.8 - Qual a função de cada um dos funcionários? E há quantos anos estão contratados? 15](#_Toc138271083)

[6.9 - Qual a idade e o parentesco de cada visitante 15](#_Toc138271084)

[6.10 -Total de visitas por tipo de visita 15](#_Toc138271085)

[6.11 - Consulta que apresenta dos dados da visita, com todos os valores substituídos pela informação adjacente ao id registado 16](#_Toc138271086)

[6.12 - Consulta que apresenta dos dados da visita, com todos os valores substituídos pela informação adjacente ao id registado 16](#_Toc138271087)

[7. Triggers 18](#_Toc138271088)

[7.1 –Trigger para Pessoas 18](#_Toc138271089)

[7.2 –Trigger para Visitas 19](#_Toc138271090)

[7.3 –Trigger para Funcionário 20](#_Toc138271091)

[7.4 –Trigger para Utentes 21](#_Toc138271092)

[8. Referências 22](#_Toc138271093)

# 1. Introdução

Nos dias de hoje, mesmo com todo o avanço tecnológico que presenciamos, muitos trabalhos de gestão ainda são feitos em formato de papel.

O presente documento apresenta uma Análise de Dados realizada sobre o cenário Gestão de Visitas a Crianças/Jovens Institucionalizados. O documento foi realizado no âmbito da Unidade Curricular de Conceção e Desenvolvimento de Bases de Dados do curso Tecnológico Superior Profissional de Programação em Sistemas de Informação.

Este projeto vai ser constituído por diversas tabelas na qual serão utilizadas para a gestão das visitas. Como as seguintes: Pessoa, Funcionário, Utente, Visitante, Visita, Categoria de Funcionário, Acolhimento, tipoVisita e sala.

# 2. Descrição do Sistema

Este Sistema permitirá fazer a gestão das visitas a crianças/jovens institucionalizados.

Para alem dos dados de utentes, funcionários e visitantes, a base de dados desenvolvida apresenta os dados das visitas e locais das visitas.

Com o armazenamento destes dados, o sistema pretende responder com clareza a às seguintes questões:

**Q1:** Qual o tipo de acolhimento de cada utente?

**Q2:** Quais funcionários são também visitantes de utentes?

**Q3:** Que utentes receberam visitas em determinada data?

**Q4:** Quantas salas existem para realização de visitas na instituição?

**Q5:** Qual a função de cada um dos funcionários?

**Q6:** Há quantos anos está contratado cada funcionário?

**3. Modelo Conceptual**

Este capítulo apresenta as várias componentes do Modelo Conceptual da base de dados a construir. Do Modelo Conceptual fazem parte o Diagrama de Entidade-Relacionamento (**Erro! A origem da referência não foi encontrada.**) e ainda descrição pormenorizada das características de cada atributo de cada entidade (**Erro! A origem da referência não foi encontrada.**).

3.1. Descrição das entidades e dos relacionamentos

Esta secção descreve os atributos de cada entidade e as características de cada um. As entidades surgem por ordem alfabética do seu nome.

| **Entidade Pessoa** | | | | |
| --- | --- | --- | --- | --- |
| Atributo | Descrição | Domínio | Preenchimento  Obrigatório | Chave Primária |
| Id | Identificador de pessoa | INTEGER | **X** | **X** |
| Pnome | Primeiro Nome | VARCHAR(15) | **X** |  |
| Apelido | Apelido | VARCHAR(15) |  |  |
| DtaNascimento | Data de nascimento da pessoa | DATE | **X** |  |
| Género | Género da Pessoa | ENUM | **X** |  |

Tabela 1 – Entidade Pessoa

| **Entidade Pessoa-> Funcionário** | | | | |
| --- | --- | --- | --- | --- |
| Atributo | Descrição | Domínio | Preenchimento  Obrigatório | Chave Primária |
| IdPessoa | Identificador do funcionário | INTEGER | **X** | **X** |
| Salario | Salário do funcionário | Float | **X** |  |
| idCategoria | Chave primária da tabela categoria | Mesmo tipo que a coluna alvo | **X** |  |
| dtaContrato | Data Contrato | DATETIME | **X** |  |
| idPessoa | Chave de Herança | Mesmo tipo que a coluna alvo | **X** |  |

Tabela 2 - Entidade Funcionário

| **Entidade Pessoa->Utente** | | | | |
| --- | --- | --- | --- | --- |
| Atributo | Descrição | Domínio | Preenchimento  Obrigatório | Chave Primária |
| Id | Identificador do utente | INTEGER | **X** | **X** |
| dtaEntrada | Data de entrada | DATETIME | **X** |  |
| idAcolhimento | Chave primária tabela Acolhimento | Mesmo tipo que a coluna alvo | **X** |  |
| idPessoa | Chave de Herança | Mesmo tipo que a coluna alvo | **X** |  |

Tabela 3 - Entidade Utente

| **Entidade Pessoa->Visitante** | | | | |
| --- | --- | --- | --- | --- |
| Atributo | Descrição | Domínio | Preenchimento  Obrigatório | Chave Primária |
| Id | Identificador | INTEGER | **X** | **X** |
| idPessoa | Chave de Herança | Mesmo tipo que a coluna alvo | **X** |  |

Tabela 4 - Entidade Visitante

| **Entidade Categoria** | | | | |
| --- | --- | --- | --- | --- |
| Atributo | Descrição | Domínio | Preenchimento  Obrigatório | Chave Primária |
| Id | Identificador da categoria | INTEGER | **X** | **X** |
| Função | Função do funcionário | VARCHAR(150) | **X** |  |

Tabela 5 - Entidade Categoria

| **Entidade Sala** | | | | |
| --- | --- | --- | --- | --- |
| Atributo | Descrição | Domínio | Preenchimento  Obrigatório | Chave Primária |
| Id | Identificador da sala | INTEGER | **X** | **X** |
| Porta | Número de porta | INTEGER | **X** |  |
| Andar | Andar do edifício | INTEGER |  |  |
| Edifício | Edifício da Sala | VARCHAR(2) | **X** |  |

Tabela 6 - Entidade Sala

| **Entidade Visita** | | | | |
| --- | --- | --- | --- | --- |
| Atributo | Descrição | Domínio | Preenchimento  Obrigatório | Chave Primária |
| Id | Identificador da sala | INTEGER | **X** | **X** |
| dtaVisita | Data da visita | DATETIME | **X** |  |
| idTipo | Chave primária tabela Tipo | Mesmo tipo que a coluna alvo | **x** |  |
| idSala | Local Visita | Mesmo tipo que a coluna alvo | **X** |  |
| idUtente | Número de utente associado | Mesmo tipo que a coluna alvo | **X** |  |

Tabela 7 - Entidade Visita

| **Entidade Tipo** | | | | |
| --- | --- | --- | --- | --- |
| Atributo | Descrição | Domínio | Preenchimento  Obrigatório | Chave Primária |
| Id | Identificador de tipo | INTEGER | **X** | **X** |
| tipo | Tipo de visita | ENUM | **X** |  |
| Descrição | Descrição de visita | VARCHAR(250) | **x** |  |

Tabela 8 - Entidade Tipo

| **Entidade Acolhimento** | | | | |
| --- | --- | --- | --- | --- |
| Atributo | Descrição | Domínio | Preenchimento  Obrigatório | Chave Primária |
| Id | Identificador do acolhimento | INTEGER | **X** | **X** |
| Data | Data do acolhimento | DATETIME | **X** |  |
| Delegação | Quem realizou o acolhimento | ENUM |  |  |

Tabela 9 - Entidade Acolhimento

| **Tabela Contacto** | | | | |
| --- | --- | --- | --- | --- |
| Atributo | Descrição | Domínio | Preenchimento  Obrigatório | Chave Primária |
| Id\_visitante | Chave estrangeira identificada no modelo lógico | Mesmo tipo que a coluna alvo | **X** | **Sim – faz parte da chave primária** |
| Contacto | Telefone do cliente | Varchar(15) | **X** | **Sim – faz parte da chave primária** |

Tabela 10 - Tabela de dados multi - valor Contacto

3.2 Diagrama de Entidade-Relacionamento

O diagrama de Entidade-Relacionamento apresentado nesta secção (**Erro! A origem da referência não foi encontrada.**) representa os dados e relacionamentos entre dados do sistema. A simbologia usada no diagrama respeita os formalismos lecionados nas aulas de Conceção e Desenvolvimento de Bases de Dados do ano letivo 2016/2017. Uma legenda desta simbologia é apresentada na **Erro! A origem da referência não foi encontrada.**.



Figura 1 - Legenda da simbologia usada no Diagrama de Entidade-Relacionamento do presente relatório.

Uma imagem com diagrama, desenho, esboço, Desenho de linha

Descrição gerada automaticamente

Figura 2 - Diagrama de Entidade-Relacionamento da base de dados

# 4. Modelo Lógico

Este capítulo apresenta o Modelo Lógico derivado a partir do Modelo Conceptual da base de dados.

A derivação do Modelo Lógico a partir do Modelo conceptual foi realizada por aplicação das regras de transformação lecionadas na Unidade Curricular de Conceção e Desenvolvimento de Bases de Dados.

O Modelo Lógico segue a seguinte abordagem de representação:

* As chaves primárias das tabelas são representadas com estilo **negrito e sublinhado**.
* As chaves estrangeiras das tabelas são representadas com estilo *itálico*.
* A representação da chave primária correspondente a cada chave estrangeira é feita por intermédio de uma seta dirigida para o nome da tabela onde se encontra essa chave primária.

Visita(id)

Visitante(idPessoa)

TipoVisita(id)

Sala(id)

Visitante(idPessoa)

Pessoa(id)

Pessoa(id)

Utente(id)

Pessoa(id)

Acolhimento(id)

Categoria(id)

Pessoa(**id**, pnome, apelido, genero, dtaNascimento)

Funcionario(***idPessoa***, salario, dtaContrato*, idCategoria*)

Utente(***idPessoa***, dtaEntrada, *idAcolhimento*)

Visitante(***idPessoa***, parentesco)

Contacto(**idVisitante**, telemovel)

Categoria(**id**, funcao)

Acolhimento(**id**, delegacao, descricao)

Visita(**id**, dtaVisita, *idUtente, idSala, idTipoVisita*)

TipoVisita(**id**, tipo, descricao)

Sala(id, edifício, andar, sede)

VisitanteVisita(**idVisitante, idVisita**)

Vista idade para calcular idade de Pessoa

Figura 3 - Modelo Lógico da base de dados (tabelas por ordem alfabética do nome)

# 5. Permissões de acesso

Este capítulo apresenta as permissões de acesso aos dados associadas a cada tipo de utilizador final da base de dados. As tabelas seguintes resumem as permissões de consulta, de inserção, de eliminação e de atualização de dados que cada perfil de utilizador possui sobre a base de dados.



Tabela 12 – Regras de acesso a dados para utilizadores do perfil "DIRETOR".



Tabela 13 – Regras de acesso a dados para utilizadores do perfil "TASOCIAL".

# 6. Consultas à base de dados

Este capítulo apresenta as consultas programadas para responder a questões operacionais relevantes.

## 6.1 – Consulta de visitas de cada visitante com o respetivo utente, ordenado por data mais antiga.

|  |  |
| --- | --- |
| **Justificação** | **Código SQL** |
| Organização e consulta numa forma bem organizada pelo visitante com o respetivo utente. | SELECT v.id AS idVisita,  CONCAT(pv.pnome, ' ', pv.apelido) AS NomeVisitante,  CONCAT(pu.pnome, ' ', pu.apelido) AS NomeUtente  FROM Visita v  JOIN Utente u ON v.idUtente = u.idPessoa  JOIN VisitanteVisita vv ON v.id = vv.idVisita  JOIN Visitante vi ON vv.idVisitante = vi.idPessoa  JOIN Pessoa pv ON vi.idPessoa = pv.id  JOIN Pessoa pu ON u.idPessoa = pu.id  ORDER BY v.dtaVisita ASC; |
| Output | |
|  | |

## 6.2 –Consultar os dados da tabela Pessoa com o respetivo perfil

|  |  |
| --- | --- |
| **Justificação** | **Código SQL** |
| Visualização clara do perfil de cada pessoa | SELECT p.\*,  CASE  WHEN f.idPessoa IS NOT NULL THEN 'Funcionario'  WHEN u.idPessoa IS NOT NULL THEN 'Utente'  WHEN v.idPessoa IS NOT NULL THEN 'Visitante'  END AS Perfil  FROM Pessoa p  LEFT JOIN Funcionario f ON p.id = f.idPessoa  LEFT JOIN Utente u ON p.id = u.idPessoa  LEFT JOIN Visitante v ON p.id = v.idPessoa  ORDER BY id ASC; |
| **Output** | |
|  | |

## 6.3 – Qual o tipo de acolhimento de cada utente?

|  |  |
| --- | --- |
| **Justificação** | **Código SQL** |
| Dados de cada utente. | SELECT \* FROM v\_selectutente; |
| **Output** | |
|  | |

## 6.4 –Quantidade de visitas por faixa etária dos utentes.

|  |  |
| --- | --- |
| **Justificação** | **Código SQL** |
| Consulta para saber uma a faixa etária sobre os utentes. | SELECT CASE  WHEN TIMESTAMPDIFF(YEAR, p.dtaNascimento, CURDATE()) BETWEEN 0 AND 5 THEN ' 0 - 5 anos'  WHEN TIMESTAMPDIFF(YEAR, p.dtaNascimento, CURDATE()) BETWEEN 6 AND 12 THEN ' 6 - 12 anos'  WHEN TIMESTAMPDIFF(YEAR, p.dtaNascimento, CURDATE()) BETWEEN 13 AND 18 THEN '13 - 18 anos'  ELSE 'Maior de 18 anos'  END AS FaixaEtaria,  COUNT(\*) AS NumVisitas  FROM Visita v  INNER JOIN Utente u ON v.idUtente = u.idPessoa  INNER JOIN Pessoa p ON u.idPessoa = p.id  GROUP BY FaixaEtaria  ORDER BY FaixaEtaria ASC; |
| **Output** | |
|  | |

## 6.5 - Quais funcionários são também visitantes

|  |  |
| --- | --- |
| **Justificação** | **Código SQL** |
| Saber qual dos funcionários também possa ser um visitante. | SELECT  p.id,          CONCAT(p.pnome, ' ', p.apelido) AS Nome,          c.funcao AS Função,          v.parentesco AS Parentesco  FROM Pessoa p  JOIN Funcionario f ON p.id = f.idPessoa  JOIN Visitante v ON p.id = v.idPessoa  JOIN categoria c ON f.idCategoria = c.id; |
| **Output** | |
|  | |

## 6.6 - Quantas salas existem para realização de visitas na instituição?

|  |  |
| --- | --- |
| **Justificação** | **Código SQL** |
| Número de salas disponíveis para visitas. | SELECT  COUNT(\*) AS NumeroDeSalas  FROM sala  WHERE edificio != '0'; |
| **Output** | |
|  | |

## 6.8 - Qual a função de cada um dos funcionários? E há quantos anos estão contratados?

|  |  |
| --- | --- |
| **Justificação** | **Código SQL** |
| Dados dos funcionários, mais concreto os seus anos de serviço e a sua função. | **SELECT \* FROM v\_selectfuncionario;** |
| **Output** | |
|  | |

## 6.9 - Qual a idade e o parentesco de cada visitante

|  |  |
| --- | --- |
| **Justificação** | **Código SQL** |
| Perfil dos visitantes.  Vista criada para selecionar toda a | SELECT \* FROM v\_SelectVisitante; |
| **Output** | |
|  | |

## 6.10 -Total de visitas por tipo de visita

|  |  |
| --- | --- |
| **Justificação** | **Código SQL** |
| Qual tipo de visita é mais frequente. | SELECT tv.descricao AS TipoVisita,      (          SELECT COUNT(\*)          FROM Visita v          WHERE v.idTipoVisita = tv.id      ) AS TotalVisitas  FROM TipoVisita tv; |
| **Output** | |
|  | |

## 6.11 - Consulta que apresenta dos dados da visita, com todos os valores substituídos pela informação adjacente ao id registado

|  |  |
| --- | --- |
| **Justificação** | **Código SQL** |
| Dados da visita e possíveis alterações. | SELECT v.id, v.dtaVisita,         CONCAT(ut.pnome, ' ', ut.apelido) AS NomeUtente,         CONCAT(f.pnome, ' ', f.apelido) AS NomeFuncionario,         CONCAT(vi.pnome, ' ', vi.apelido) AS NomeVisitante,         CONCAT(s.edificio, '.', s.andar, '.', s.porta) AS Sala,         (SELECT descricao FROM TipoVisita WHERE id = v.idTipoVisita) AS TipoVisita  FROM Visita v  INNER JOIN Utente u ON v.idUtente = u.idPessoa  INNER JOIN Pessoa ut ON u.idPessoa = ut.id  INNER JOIN Funcionario fu ON v.idFuncionario = fu.idPessoa  INNER JOIN Pessoa f ON fu.idPessoa = f.id  INNER JOIN VisitanteVisita vv ON v.id = vv.idVisita  INNER JOIN Pessoa vi ON vv.idVisitante = vi.id  LEFT JOIN Sala s ON v.idSala = s.id  ORDER BY year(dtaVisita) DESC; |
| **Output** | |
|  | |

## 6.12 - Consulta que apresenta dos dados da visita, com todos os valores substituídos pela informação adjacente ao id registado

|  |  |
| --- | --- |
| **Justificação** | **Código SQL** |
| Número total de visitas por ano. | SELECT EXTRACT(YEAR FROM dtaVisita) AS Ano, COUNT(\*) AS NumVisitas  FROM Visita  GROUP BY Ano; |
| **Output** | |
| Uma imagem com texto, Tipo de letra, captura de ecrã, file  Descrição gerada automaticamente | |

# 7. Triggers

Este capítulo apresenta os *triggers* criados.

## 7.1 –Trigger para Pessoas

|  |
| --- |
| **Código SQL** |
| CREATE TRIGGER Insert\_Pessoa  BEFORE INSERT ON Pessoa  FOR EACH ROW      BEGIN          IF NEW.dtaNascimento > CURDATE()    THEN              SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = "Data inválida";          END IF;      END$$  DELIMITER ; |
| **Justificação** |
| O *trigger* valida a data de nascimento da pessoa, na qual se a data de nascimento for maior que a data atual, irá aparecer “Data inválida”. |

|  |
| --- |
| **Código SQL** |
| CREATE TRIGGER Update\_Pessoa  BEFORE UPDATE ON Pessoa  FOR EACH ROW      BEGIN          IF NEW.dtaNascimento > CURDATE()    THEN              SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = "Data inválida";          END IF;      END$$  DELIMITER ; |
| **Justificação** |
| O *trigger* valida a data de nascimento da pessoa, na qual se a data de nascimento for maior que a data atual, irá aparecer “Data inválida”. |

## 7.2 –Trigger para Visitas

|  |
| --- |
| **Código SQL** |
| CREATE TRIGGER Insert\_Visita  BEFORE INSERT ON visita  FOR EACH ROW      BEGIN          IF NEW.dtaVisita > CURDATE() THEN              SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = "Data de visita não pode ser menor que a atual";          END IF;      END$$  DELIMITER ; |
| **Justificação** |
| O *trigger* valida a data da visita não pode ser maior que a data atual, irá aparecer “Data de visita não pode ser menor que a atual”. |

|  |
| --- |
| **Código SQL** |
| CREATE TRIGGER Update\_Visita  BEFORE  UPDATE ON visita  FOR EACH ROW      BEGIN          IF NEW.dtaVisita > CURDATE() THEN              SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = "Data de visita não pode ser maior que a atual";          END IF;      END$$  DELIMITER ; |
| **Justificação** |
| O *trigger* valida a data da visita não pode ser maior que a data atual, irá aparecer “Data de visita não pode ser menor que a atual”. |

## 7.3 –Trigger para Funcionário

|  |
| --- |
| **Código SQL** |
| CREATE TRIGGER Insert\_Funcionario  BEFORE INSERT ON Funcionario  FOR EACH ROW      BEGIN          IF NEW.dtaContrato < (SELECT dtaNascimento FROM pessoa WHERE id = NEW.idPessoa) THEN              SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = "Data de contrato não pode ser menor que a de Nascimento";          END IF;      END$$  DELIMITER ; |
| **Justificação** |
| O *trigger* valida a data da contratação não pode ser menor que a data de nascimento, irá aparecer “Data de contrato não pode ser menor que a de Nascimento”. |

|  |
| --- |
| **Código SQL** |
| CREATE TRIGGER Update\_Funcionario  BEFORE UPDATE ON Funcionario  FOR EACH ROW      BEGIN          IF NEW.dtaContrato < (SELECT dtaNascimento FROM pessoa WHERE id = OLD.idPessoa) THEN              SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = "Data de contrato não pode ser menor que a de Nascimento";          END IF;      END$$  DELIMITER ; |
| **Justificação** |
| O *trigger* valida a data da contratação não pode ser menor que a data de nascimento, irá aparecer “Data de contrato não pode ser menor que a de Nascimento” |

## 7.4 –Trigger para Utentes

|  |
| --- |
| **Código SQL** |
| CREATE TRIGGER Update\_Utente  BEFORE UPDATE ON Utente  FOR EACH ROW      BEGIN          IF NEW.dtaEntrada < (SELECT dtaNascimento FROM pessoa WHERE id = OLD.idPessoa) THEN              SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = "Data de entrada não pode ser menor que a de Nascimento";          END IF;      END$$  DELIMITER ; |
| **Justificação** |
| O *trigger* valida a data da entrada não pode ser menor que a data de nascimento, irá aparecer “Data de entrada não pode ser menor que a de Nascimento”. |

|  |
| --- |
| **Código SQL** |
| CREATE TRIGGER Insert\_Utente  BEFORE INSERT ON Utente  FOR EACH ROW      BEGIN          IF NEW.dtaEntrada < (SELECT dtaNascimento FROM pessoa WHERE id = NEW.idPessoa) THEN              SIGNAL SQLSTATE '45000' SET MESSAGE\_TEXT = "Data de entrada não pode ser menor que a de Nascimento";          END IF;      END$$  DELIMITER ; |
| **Justificação** |
| O *trigger* valida a data da entrada não pode ser menor que a data de nascimento, irá aparecer “Data de entrada não pode ser menor que a de Nascimento” |

# 8. Referências

* Materiais da unidade curricular Conceção e Desenvolvimento de Bases de Dados (ano letivo 2019/2020, em *ead.ipleiria.pt*)
* *Fundamentals of Database Systems*, R. Elmasri & S. B. Navathe, Addison Wesley Pub. Co.
* SQL - *Structured Query Language* (6.ª edição), L. Damas, FCA, 2005
* ...