



## **CTeSP em Informática**

**PROGRAMAÇÃO ORIENTADA POR OBJETOS**

**Enunciado do Trabalho**

**Época de Frequência + Época Normal de Exame  
2025-2026**

## REGRAS DE REALIZAÇÃO

1. Antes de começar o trabalho leia atentamente o enunciado, consultando o mesmo periodicamente, durante a execução do trabalho.
2. O trabalho deve ser realizado por grupos de 2 alunos. Exceções a esta regra devem ser solicitadas ao docente, desde que apresentem uma justificação 'de força maior'. O docente da disciplina é soberano na decisão de aceitação ou não aceitação.
3. As datas irrevogáveis de **entrega do trabalho nas várias épocas são os dias/horas publicados** nos calendários oficiais de avaliação, e que consta também em cada *slot* de submissão. A submissão deve ser efetuada através da página da disciplina na plataforma MS Teams, no *slot* aberto especificamente para esse fim.
4. O trabalho deve ser submetido eletronicamente na página da disciplina, num único ficheiro compactado e nomeado:
  - a. <nº aluno1-nº aluno2>-trabalho final poo 2223-{efre | enex}.zip
  - b. Exemplos para Época de Frequência e Época Normal de Exame, respetivamente:
    - i. 11111-22222-trabalho poo 2223-efre.zip
    - ii. 11111-22222-trabalho poo 2223-enex.zip

Não são aceites trabalhos submetidos por outras vias que não eletronicamente, através da página da disciplina no MS Teams, no *slot* aberto para o efeito. **Não serão aceites justificações como falta de ligação internet ou quaisquer outras.**

Por esta razão não deixe a submissão para o último minuto.

5. Cada trabalho deve conter obrigatoriamente os seguintes itens:
  - a. Diretoria nomeada "Codigo Fonte", contendo o Código Fonte de todos os ficheiros da *Solution* com todos os projetos, pronto a ser aberto no editor do **MS Visual Studio 2022 (versão maior ou igual 17.4)**, e executado sem qualquer erro.
  - b. Diretoria nomeada "Executavel" contendo o executável e todos os ficheiros/diretorias auxiliares, correspondente ao Código Fonte enviado, para a sua execução. Os eventuais acessos a essas diretorias devem ser relativos e nunca absolutos. Se o executável não correr, o projeto não será avaliado e a nota será zero (0) valores.
  - c. Diretoria "Relatorio", contendo o relatório do projeto em formato PDF;
6. A avaliação será feita com a seguinte distribuição de cotação:
  - a. Relatório conforme relatório exemplo na página da disciplina **(2 valores)**;
  - b. Qualidade do código: Código bem estruturado; Utilização dos paradigmas *Object Oriented*; Código Fonte devidamente comentado bem como devidamente *indentado*; Variáveis escritas em camelCase e os métodos, propriedades, classes, *delegates*, eventos e outros em PascalCase **(4 valores)**;
  - c. Implementação em ADO.NET/Entity Framework Core **(11 valores)**;
  - d. Melhorias ao enunciado.....**(3 valores)**;
7. Discussão: Haverá lugar à discussão presencial sobre o trabalho/relatório desenvolvido e entregue. **A discussão faz parte do projeto, sendo que a nota do projeto será de zero (0) valores se o aluno faltar à discussão do mesmo.**
8. Fazem parte deste enunciado, como anexos em ficheiros à parte:
  - a. O script para criação da base de dados em MS SQL Server;
  - b. O script para carregamento de alguns registos de amostra na base de dados;
  - c. O modelo de relatório.

9. **Trabalhos copiados de outras fontes serão classificados com nota 0 (zero) independentemente de qual(is) a(s) cópia(s) e qual o original. e têm ainda como consequência a eliminação do processo de avaliação em 2025/2026, para além do acionamento dos mecanismos disciplinares em vigor na ESTA.**

## **CONTEXTO DE REALIZAÇÃO**

O trabalho foi pensado de forma a levar os alunos a utilizarem e consolidarem os conhecimentos adquiridos.

O objetivo principal deste projeto é o desenvolvimento duma aplicação interativa desenvolvida em paradigma Windows Form Application ou Web, com recurso à linguagem de programação C#, com persistência de dados em MS SQL Server ou MySQL (os Scripts de criação da BD e Carregamento de dados estão validados para MS SQL).

Esta aplicação pretende gerir uma base de dados de informações escolares relativas a alunos, cursos e outras.

## **REQUISITOS/FUNCIONALIDADES**

### **Base de Dados para persistência dos dados**

A base de dados para persistência dos dados é muito simples e o script para a sua criação é disponibilizado no ficheiro anexo em separado e designado por “EscolaDB Schema.sql”. O ficheiro “EscolaDB Data.sql”, contém alguns registos de amostra, que podem ser carregados para testes.

Deve abrir este ficheiro no SQL Server Management Studio, e executar o mesmo de forma a criar a base de dados EscolaDB, as respetivas tabelas e se pretender, carregar também alguns dados de teste.

Para melhor compreensão da mesma apresenta-se de seguida o diagrama EA, com as várias tabelas e suas associações:

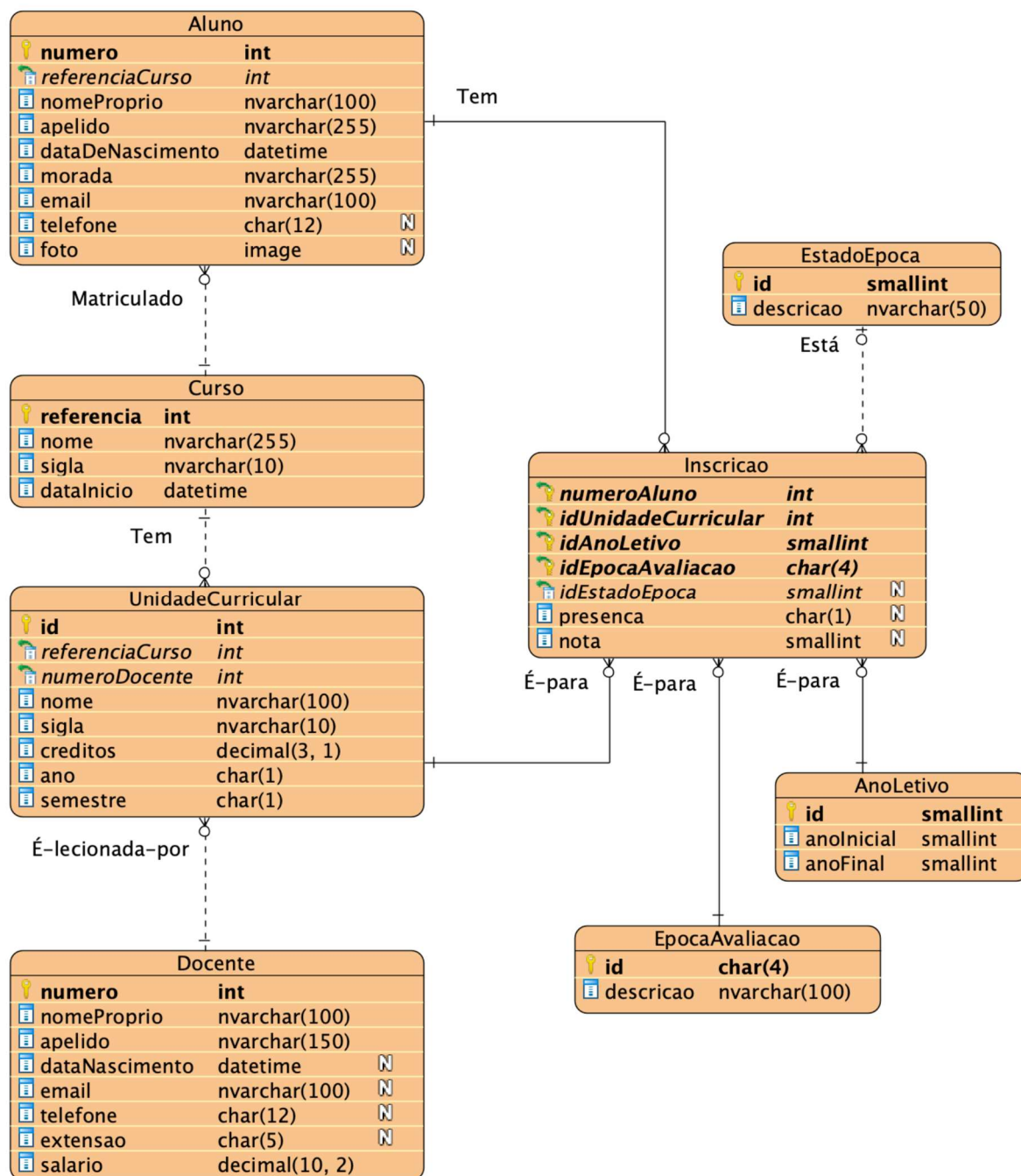


Figura 1 – ERD físico base de dados

**Esta estrutura de dados permite entre outras situações, que:**

- Um curso (Curso) esteja registado na base de dados (bd) sem ter ainda nenhum aluno (Aluno) e/ou nenhuma unidade curricular (UnidadeCurricular) associados.
- Um docente (Docente) esteja registado na bd sem estar associado a nenhuma unidade curricular (UnidadeCurricular).
- Uma unidade curricular (UnidadeCurricular) esteja registada na bd sem estar associada a nenhuma inscrição (Inscricao).
- Um aluno (Aluno) esteja registado na bd sem estar associado a nenhuma inscrição (Inscricao).
- Um ano letivo (AnoLetivo) esteja registado na bd sem ter ainda nenhuma associação com a inscrição (Inscrição).
- Uma época de avaliação (EpocaAvaliacao) esteja registada na bd sem ter ainda nenhuma associação com a inscrição (Inscrição).

Note que tabelas diferentes podem ter campos com o mesmo nome, sem que isso signifique qualquer correspondência entre os mesmos tal como id em AnoLetivo e id em EpocaAvaliacao. A exceção a esta afirmação são eventualmente, claro está, as chaves estrangeiras.

Os tipos de dados dos vários campos das várias tabelas podem ser também visualizados no script de criação da base de dados.

**Regras de Negócio**

1. A primeira inscrição de um aluno numa disciplina num dado ano letivo tem de ser sempre para a Época de Avaliação de Frequência;
2. Uma inscrição em Época Normal de Exame é feita automaticamente quando:
  - a. É lançada uma nota com valor menor que dez (10), caso em que a presença deve ficar com o valor "P" e o idEstadoEpoca correspondente à descrição "Admitido".
  - b. No caso de o aluno faltar a uma época de avaliação o campo presença deve ser preenchido com "F" e o idEstadoEpoca automaticamente preenchido com o correspondente à descrição "Admitido".
3. A inscrição em Época de Avaliação de Recurso e Época Especial devem ser efetuadas manualmente e têm obrigatoriamente de existir uma época de avaliação imediatamente anterior válida, isto é, com o idEstadoEpoca preenchido com o correspondente à descrição "Reprovado". A época imediatamente anterior à Época de Recurso é a Época Normal de Exame e a imediatamente anterior à Época Especial é a Época de Recurso.
4. Em qualquer época de avaliação, o lançamento de uma nota com valor maior ou igual que dez (10), implica que a presença deve ficar com o valor "P" e o idEstadoEpoca correspondente à descrição "Aprovado".
5. Não pode existir a inscrição numa época de avaliação subsequente a uma época de avaliação com o o idEstadoEpoca correspondente à descrição "Aprovado".

## **Manipulação de dados independente (CRUD)**

Deve ser possível Inserir (C - Create), Consultar (R - Read), Alterar (U - Update) e Apagar (D - Delete) os dados de qualquer uma das tabelas, de forma independente.

## **Consultas diversas**

Todas as consultas são para ser efetuadas no ecrã.

Todas as tabelas devem ter um ecrã onde possam ser realizadas as operações de CRUD.

Para além das consultas já atrás mencionadas, devem também ser implementadas as seguintes consultas em ecrã:

- a) Ecrã com uma zona superior com a informação multilinha (gridview) relativa às disciplinas e uma zona inferior (gridview), onde apareça a informação multilinha relativa aos alunos inscritos. À medida que se seleciona uma unidade curricular na parte superior, devem ser mostrados automaticamente, na parte inferior, todos os alunos inscritos na unidade curricular selecionada, (vulgo Master-Detail).

Neste ecrã deve ser possível filtrar as unidades curriculares por curso, por ano e por semestre. Se o ano e/ou o semestre podem não ser preenchidos, mas o curso tem sempre de ser preenchido.

O campo do tipo imagem, da tabela Aluno não deve fazer parte das grelhas (gridview), qualquer que seja o ecrã onde se visualizem alunos. Deve ser um campo do tipo pictureBox, ou equivalente, estar à parte e estar sincronizados com a linha do aluno que se encontrar selecionado. O foco inicial deve ser sempre a primeira linha da gridview.

- b) b. Ecrã de épocas e notas. Ecrã onde seja possível selecionar um aluno com base no seu número obrigatoriamente, e opcionalmente o ano letivo, o semestre e opção Inscrito/Não Inscrito/Todas.  
Com base na informação selecionada deve mostrar as unidades curriculares, ano letivo, época de avaliação, estado da inscrição, conforme parâmetros da consulta, e a respetiva informação adicional Nota e Estado da Época

Pode implementar outras consultas à sua escolha, se pretender, estas serão tidas em conta na valorização global do trabalho.

## Apresentação e Robustez da aplicação

- As várias funcionalidades devem ser acessíveis através de um menu/submenus horizontal (MenuStrip).
- Deve fazer uso do paradigma Orientação por Objetos em toda a sua extensão.
- A aplicação deve ser o mais modular possível, isto é, as várias funcionalidades devem ser implementadas de forma autónoma e não todas ou várias agregadas num único form.
- A aplicação deve ser robusta, isto é, deve estar protegida contra manuseamento indevido. Em termos de código, significa que o aluno deve prever introdução indevida de tipos de dados
- Os *forms* devem ser sóbrios (sem fundo) e os objetos neles contidos devem estar bem organizados e alinhados.
- A implementação deve ser efetuada com separação entre a camada de acesso a dados e a camada de apresentação.

## Outras questões

Para conexão à base de dados pode utilizar ADO.NET como ensinado nas aulas, ou *Entity Framework Core 8 (EF 8)*.

**Se optar pela EF8, deve utilizar a abordagem Code-First com a base de dados já pré-existente.**

Neste caso, a aplicação deve recorrer a um ficheiro de configuração (appsettings.json), que contenha a *string* de conexão. Esta funcionalidade é obrigatória. Trabalhos que não apresentem esta funcionalidade não serão avaliados, sendo cotados com zero (0) valores.

**Se optar pelo ADO.NET, tem de utilizar obrigatoriamente o modo desconectado. Implementações em modo conectado não são aceites e serão cotadas com zero (0) valores.**

Neste caso, a aplicação deve recorrer a um ficheiro de configuração (app.config), que contenha a *string* de conexão. Esta funcionalidade é obrigatória. Trabalhos que não apresentem esta funcionalidade não serão avaliados, sendo cotados com zero (0) valores.

O primeiro menu da aplicação deve ser o menu *Ficheiro* onde deve existir um submenu Exit, com a funcionalidade de sair da aplicação.

O último menu da aplicação deve ser o menu *Ajuda* onde deve existir um submenu *Acerca*, que quando selecionada abra uma janela com informação referente à ficha técnica do projeto (número e nome dos alunos que desenvolveram e outras).

Nota: Este trabalho é de programação, a base de dados, é apenas um meio, pelo que qualquer necessidade que o aluno tenha de efetuar um comando SQL e não saiba elaborá-lo, pode sempre solicitar ao docente da disciplina, que disponibilizará a *query* (comando SQL) pretendida.

## Outras Melhorias e dúvidas

Outras melhorias que valorizem o trabalho podem e devem ser efetuadas pelos alunos, desde que as mesmas não violem os requisitos enunciados anteriormente.

Exemplos de melhorias podem ser a utilização de um objeto calendário nos campos do tipo data, menus com ícones, consultas várias, listagens várias, etc.

Qualquer esclarecimento sobre o trabalho pode ser colocado ao docente, no canal “Chat” da página da disciplina no MS TEAMS.