

Desenvolvimento de Aplicações

Curso Técnico Superior Profissional de
Programação de Sistemas de Informação



Projeto Final

CineGest

Enunciado



Página em branco

Nota: V1.0 – O presente enunciado pode sofrer alterações, caso sejam detetados erros ou omissões

1. Descrição

Um cinema pretende criar um sistema de gestão próprio que lhe permita gerir todo o seu negócio, desde sistemas internos até aos bilhetes de clientes. Para isso, será necessário um software que implemente todas as funcionalidades necessárias para gerir toda a operação, incluindo criar e manter os filmes em exibição, gerir as salas do cinema, emitir bilhetes para clientes, guardar dados de trabalhadores, entre outros.

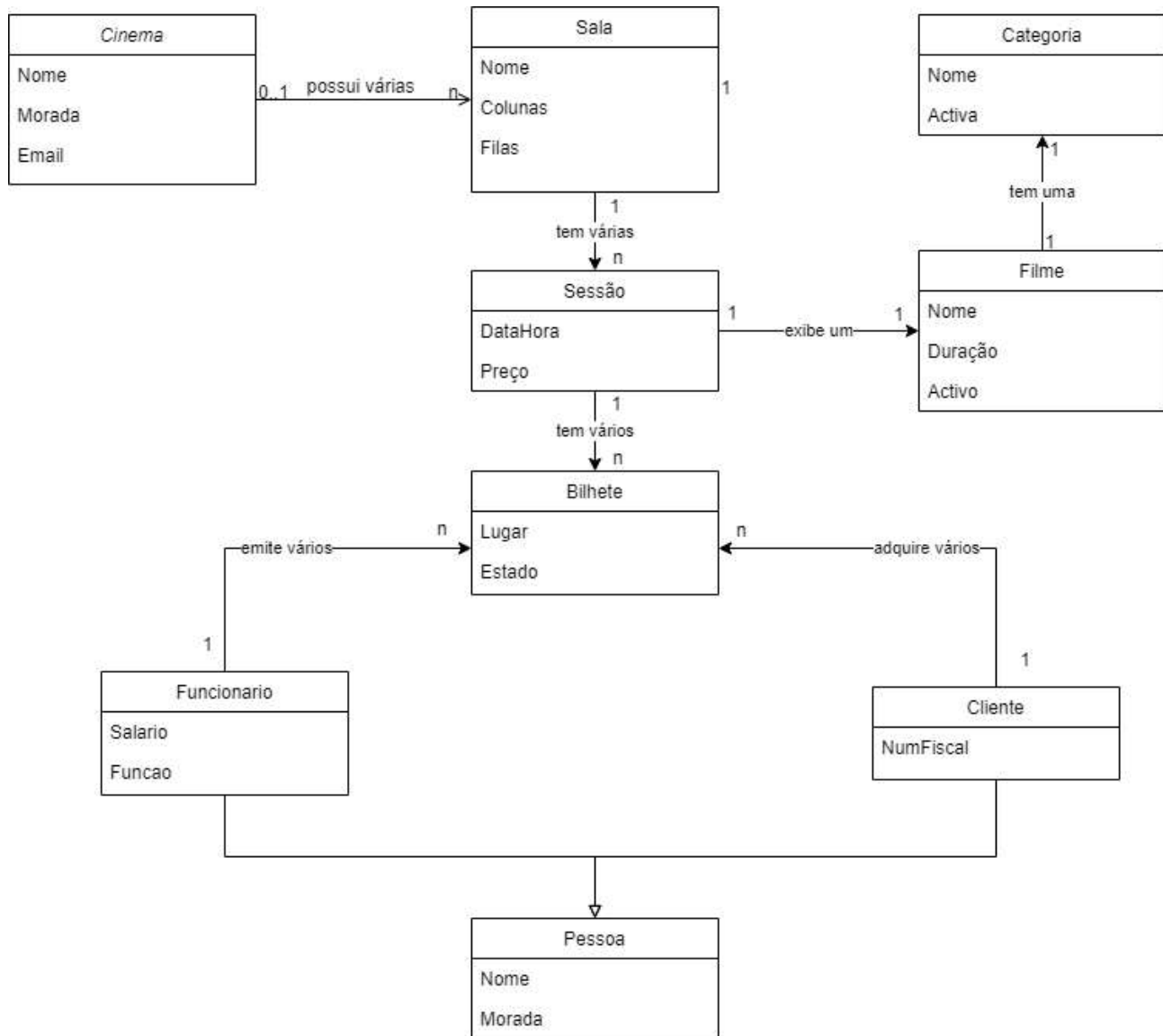
Assim, é importante saber que:

1. O programa será utilizado exclusivamente por trabalhadores do **Cinema**;
2. Por defeito, a aplicação não deverá conter nenhum **Cinema**, este deverá ser configurado na utilização inicial da aplicação;
3. As informações do **Cinema** (nome, morada, email) poderão ser editadas a qualquer momento, mas apenas deverá existir um **Cinema**;
4. O **Cinema** poderá ter um número configurável de **Salas**;
5. Cada **Sala** terá um número configurável de filas e colunas para lugares;
6. Uma **Sala** poderá, ou não, ter um **Filme** associado em exibição, mas um **Filme** poderá estar em exibição em mais do que uma **Sala**;
7. Deverá ser possível criar **Sessões** para uma **Sala**, onde será exibido o **Filme** associado a uma certa data e hora;
8. Cada **Filme** deverá ser **categorizado** numa das seguintes categorias: Comédia, Sci-fi, Terror, Romance, Acção, Thriller, Drama, Mistério, Crime, Aventura, Fantasia ou Animação;
9. Para cada **Sessão** existente, deve ser possível criar **Bilhetes** para clientes;
10. Um **Bilhete** deverá ter um lugar associado, que deverá ser único para essa sessão;
11. Um **Bilhete** só poderá ser usado uma vez, devendo ter um **Estado** que comprove se foi usado ou não, e que poderá ser alterado pelo utilizador do software;
12. A compra de um **Bilhete** poderá ser feita por clientes anónimos ou por **Clientes** previamente gravados na base de dados via pesquisa;
13. Quando é feita a compra por um novo **Cliente**, deve ser oferecida a possibilidade de guardar o Cliente para agilizar compras futuras;

14. Deve ser possível criar novos **Funcionários** no programa;
15. Durante a utilização do programa, deverá ser possível alterar o **Funcionário** que está neste momento a utilizar o programa;
16. Deve ser possível visualizar uma lista de todos os **Clientes**, que mostre o nome, morada, número fiscal, número e valor total de bilhetes adquiridos;
17. Para garantir a integridade de dados, todos os dados criados durante a utilização do programa devem ser persistidos em base de dados relacional SQL Server, com recurso ao Entity Framework.

2. Modelo de dados

Considerando a descrição do problema, o modelo de dados a implementar deverá ter a seguinte estrutura:



3. Formulários a implementar

Tendo em conta a gestão da informação indicada, devem ser implementados seis formulários obrigatórios:

- O **Formulário Principal** - onde deve constar um Menu de acesso às diversas funcionalidades da aplicação – Cinema, Filmes, Sessões, Clientes e Funcionários. Nesta janela deve também ser possível alterar o funcionário que está actualmente a utilizar o programa.

Esta janela deve também mostrar todas as Sessões marcadas para o dia actual, de forma a permitir a rápida criação de novos Bilhetes para clientes.

Ao seleccionar uma Sessão neste ecrã, deverá ser mostrado o **Formulário de Atendimento**.

- b. O **Formulário de Atendimento** – deve mostrar a Sala onde a Sessão irá decorrer, mostrando todos os lugares da Sessão, e marcando a verde os que ainda estão disponíveis, e a cinzento o que já estão ocupados.

Deverá ser possível seleccionar vários lugares simultaneamente para criar múltiplos bilhetes de uma só vez.

Deve ainda ser possível preencher os dados do Cliente se este for anónimo, ou pesquisar um Cliente já existente na base de dados para preencher os campos de dados automaticamente.

Após terminar a emissão dos Bilhetes, estes deverão ser exportados para um ficheiro .TXT ou .PDF, contendo todas as informações relevantes aos bilhetes comprados, ou seja, informação do cliente, filme, data e hora, sala, lugares e o funcionário que atendeu o cliente na compra.

- c. O **Formulário do Cinema** – deve ser possível alterar as definições do Cinema, bem como adicionar ou alterar as salas existentes e permitir o CRUD dos funcionários do cinema.
- d. O **Formulário de Filmes** – deve ser utilizado para configurar os filmes que estarão disponíveis para exibição no Cinema.

Um filme terá que ter uma categoria associada, e poderá ser activado ou desactivado para permitir (ou não) a criação de novas sessões para esse filme. Por defeito, ao criar um filme este deverá começar “desactivado”, para evitar fazer estreias antes do tempo.

- e. O **Formulário de Sessões** – onde será possível criar Sessões para os filmes actualmente em exibição. Ao criar uma nova sessão, deverá ter que escolher uma sala e um filme a exhibir, bem como a data e hora.

Uma Sessão não poderá começar numa Sala se esta já estiver previamente ocupada com outra Sessão a essa hora, pelo que deverão verificar se a Sala está ocupada antes de marcar uma nova Sessão.

- f. O **Formulário Gestão de Clientes** – onde deve ser possível registar, apagar, alterar e consultar dados de clientes, bem como visualizar os dados adicionais pedidos na alínea 16 da **Descrição** da aplicação;

O número de formulários poderá ser alterado em cada caso se o acharem necessário, desde que a funcionalidade pedida seja garantida.

4. Regras

- a. O trabalho deve ser realizado em grupo de três estudantes. Grupos com número diferente carecem de aprovação do docente da UC;
- b. A entrega do trabalho deve ser efetuada através de um ficheiro compactado (ZIP) que deve conter:
 - i. Um relatório simplificado, onde deverá constar o manual de utilização e, de forma sucinta, a justificação das opções tomadas;
 - ii. O projeto com todos os ficheiros de código necessários à sua compilação e execução;
 - iii. Um ficheiro readme.txt que apresente os processos e dados necessários à instalação, configuração e execução da aplicação;
- c. O nome do ficheiro compactado, definido na alínea anterior, deve conter os nomes (primeiro e último) e números dos estudantes que compõem o grupo. Exemplo:
`AnaFelix8216658_ClaudioMiguel56476784_DianaCosta8675434.zip`.
- d. O plágio de trabalhos conduz à classificação de zero valores;
- e. O aluno, em defesa, deverá demonstrar que é capaz de implementar as mesmas funcionalidades que produziu no trabalho prático;
- f. A classificação depende sempre da defesa e contará com um valor de ponderação da qualidade da mesma definido em percentagem de 0% a 100%;
- g. As funcionalidades extra serão contabilizadas para melhoria da classificação, desde que devidamente descritas no relatório;
- h. O código do programa deve conter proteções, de forma que não apresente erros inesperados ou feche abruptamente.
- i. Serão levadas em conta, na avaliação, as proteções contra falhas e outros melhoramentos;



5. Critérios de Avaliação

Avaliação	Peso (percentagem)
Relatório e Manual	10%
Aspeto Geral	5%
Utilização de arquitetura e qualidade do código	15%
Configuração e utilização do EntityFramework	5%
Formulário Principal	5%
Funcionalidades de Atendimento e Bilhetes	20%
Funcionalidades de Cinema e Salas	10%
Funcionalidades de Sessões	15%
Gestão e utilização de dados de Clientes	5%
Gestão de Funcionários	10%

6. Datas da avaliação

- 26/04/2023 – Publicação do enunciado do projeto;
- 19/06/2023 – Entrega do projeto, na plataforma Moodle, até às 23:59 (código fonte da aplicação, relatório, ficheiro readme.txt);
- 19/06/2023 – **[ÉPOCA NORMAL]** Entrega do projeto com a Adenda, na plataforma Moodle, até às 23:59 (código fonte da aplicação, relatório, ficheiro readme.txt);
- 04/07/2023 – Defesa individual de projeto (avaliação periódica e época normal);

7. Considerações relativamente às épocas de exame

O processo avaliativo tem por base três momentos estabelecidos (avaliação periódica, época normal e época de recurso). Para a época de exame normal, além do projeto definido neste documento, este deverá ser acrescido de novas funcionalidades a implementar em acréscimo às restantes já definidas neste documento. A época de recurso não estará contemplada neste enunciado.

Convém acrescentar que em épocas de exame o desenvolvimento deverá ser feito individualmente, e não em grupo. Isto inclui o projeto base. Importa realçar, também, que a avaliação na época normal incide sobre todo o projeto, dando destaque, como será lógico, às funcionalidades introduzidas pela adenda.

Bom trabalho!

8. ADENDA: Funcionalidades a Implementar para a Entrega em Exame da Época Normal

Segundo o estipulado na avaliação da unidade curricular “Desenvolvimento de Aplicações”, o Exame da Época Normal compreende:

- 1) a entrega do projeto definido no início do semestre acrescido da adenda descrita de seguida, a qual, individualmente, deverá ter uma cotação de pelo menos 50% para conferir aprovação;
- 2) uma defesa prática individual do projeto entregue, seguindo as mesmas condições que a época periódica.

O exame da Época Normal exige, para além da implementação de todas as funcionalidades atrás descritas para a Avaliação Periódica, a implementação e programação de funcionalidades extra, a saber:

1. A implementação de uma nova funcionalidade de agendamento automático de sessões para um espaço de tempo a definir pelo utilizador.

Exemplo: Para o Filme XYZ, pretende-se agendar sessões às 11h30, 15h00, 19h00 e 21h30, todos os dias, durante as próximas 2 semanas.

Para esta funcionalidade, devem ser considerados os seguintes critérios:

- a. O utilizador deverá selecionar um filme existente na base de dados;
- b. O utilizador deverá selecionar uma sala existente na base de dados;
- c. O período temporal a escolher poderá ser entre o dia de amanhã e 31/Dez/2100;
- d. O utilizador poderá agendar entre 1 a 5 sessões para cada dia do período temporal selecionado;
- e. Deverá considerar potenciais conflitos de datas/horas entre sessões, para todas as sessões que serão criadas no período temporal selecionado;
- f. Se existirem conflitos detetados, estes devem ser mostrados ao utilizador, e nenhuma sessão deve ser marcada;
- g. Poderá considerar que todas as sessões serão marcadas sem considerações por fins de semana ou feriados;
- h. Esta funcionalidade poderá ser implementada no formulário existente para Sessões, ou num novo formulário acedido a partir deste, ficando ao critério do estudante.

Tabela de Critérios de Avaliação e Cotações (Época Normal)

Avaliação (Requisitos – Avaliação periódica)	Peso (percentagem)	Peso
Relatório e Manual	10%	Peso de 70%
Aspeto Geral	5%	
Utilização de arquitetura e qualidade do código	15%	
Configuração e utilização do EntityFramework	5%	
Formulário Principal	5%	
Funcionalidades de Atendimento e Bilhetes	20%	
Funcionalidades de Cinema e Salas	10%	
Funcionalidades de Sessões	15%	
Gestão e utilização de dados de Clientes	5%	
Gestão de Funcionários	10%	
Subtotal	100%	14
Avaliação (Requisitos adicionais – Época Normal)		
Formulário para marcação massiva	15%	Peso de 30%
Seleção de sala e filme	5%	
Seleção de período temporal	5%	
Preenchimento de sessões diárias	15%	
Prevenção de conflitos na marcação	35%	
Persistência de dados	25%	
Subtotal	100%	6
Projeto base (avaliação época periódica)	70%	6
Adenda (avaliação época normal)	30%	14
Total (Projeto * 70% + Adenda * 30%)	100%	20